System Development & Validation Process for Emerging Growing Organizations

by

Jose Antonio Almazan Lopez

Submitted to the System Design and Management Program
in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Engineering and Management

at the

Massachusetts Institute of Technology

February 2009

Signature of Author____________________________
Jose Antonio Almazan Lopez
System Design and Management Program
February 2009

Certified by____________________________
Daniel Whitney
Senior Research Scientist – Center for Technology, Policy and Industrial Development
Senior Lecturer in Engineering Systems, Senior Lecturer in Mechanical Engineering
Thesis Supervisor

Accepted by____________________________
Patrick Hale
Director – System Design and Management Fellows Program
Senior Lecturer – Engineering Systems Division

This Page is Intentionally Left Blank

To those who were there during the rainy days…

To my family and the very best friends…

All that I needed… that has always been there…

This Page is Intentionally Left Blank

System Development & Validation Process for Emerging Growing Organizations

by

Jose Antonio Almazan Lopez

Submitted to the System Design and Management Program
in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Engineering and Management

Abstract

This thesis has the main purpose of presenting the Development and Validation phase of the product development system from the point of view of an emerging and growing product development organization, denoting the obstacles and challenges presented over the operation path or testing organizations as the evolve with solution alternatives from the industry and academia.

Through the study and analysis of a multinational company of the automotive industry, it was possible to build a framework of study among regions analyzing the test and development operation of engineering organizations at four levels of growth and responsibility. By combining experience and knowledge from leaders and heroes from these regions, with the knowledge and insights from academia and literature; this study is intended to serve as a guide for a testing organization facing the challenge of growing and acquiring greater responsibility.

Today's global scenario demand excellence in execution, reducing time and resources in the development of products, being test and development the most resource consuming activities within a generic product development process but at the same time critical and unavoidable.

Through a systems engineering approach, the main challenges and improvement opportunities for the development and validation entities at the Ford of Mexico Product Development Organization are documented, with the purpose of creating awareness and proposing the elements fixing a path that would ensure the success for the company as a whole. This path sets the steps for further work in the deployment of a well defined and lean testing system.

Thesis Supervisor: Daniel Whitney
Title: Senior Research Scientist – Center for Technology, Policy and Industrial Development; Senior Lecturer in Systems and in Mechanical Engineering

Acknowledgements

Attending MIT and being able to participate within the proud System Design and Management program, representing the company who placed its trust on me; has been a privilege, an honor, and a dream made true in every sense. The completion of this achievement represents a life milestone, from which I am in debt to all who supported me, trusted me, and encouraged me; my deepest gratitude and appreciation to all who have been there in my life, making me the person who I am now.

To my Mother and Brother, for all their support, courage, and unbeatable strength.

To my Father, for all the principles, values, and vision taught through a life; encouraging me in the quest for excellence and achieving greater goals.

To Pilar, Guillermo, and Alejandro; for providing the elements to overcome any obstacle, and create value in life.

Adrian, Takahiro, Antonio, and Raul; for all their support and wisdom across this journey, making possible the completion of it.

To all my Friends whose hands were always on my back pushing me forward during the most difficult times, and whose voices always provided warm support. It would be impossible to write all your names and my appreciation to all in a single sheet, but you all know that are here.

Carlos Alarcon, who trusted me, and always helped me to reach higher goals in my professional life.

Mark-Tami Hotta, whose principles and values are a true standard that shows people how leaders must be shaped.

Roberto Villegas, whose guidance motivated the research of this effort.

Marcos Perez, whose vision moves an organization forward and inspires people.

Dan Whitney, for giving light and direction for this work to come true.

John Grace, whose wise comments helped to discover walls before rushing into them.

MIT and the valuable lessons taught to me by all of you changed me, opening my mind into a larger world, and setting the quest for greater challenges and the new adventures to come.

Table of Contents

List of Figures

Acronyms

AWS	Analytical Warranty System
BIW	Body on White
CAE	Computer Aided Engineering
CDF	Cumulative Density Function
PDF	Probability Density Function
D&R	Design & Release
D&V	Development & Validation
DVP	Design Verification Plan
FoM	Ford of Mexico
LOPD	Local On–going Product Development
LPC	Local Product Control
LVT	Local Vehicle Team
NVH	Noise Vibration and Harshness
PD	Product Development
PDO	Product Development Organization
PVT	Plant Vehicle Team
TGW	Things Go Wrong
VD	Vehicle Dynamics

This Page is Intentionally Left Blank

Chapter 1

=== Introduction ===

Testing leads to failure, and failure leads to understanding.
– Burt Rutan

1 – Introduction

The Development and Validation phases are the most resource consuming phases within the product development process. These phases are characterized by the use of extensive testing with leads to the high consumption of time, effort, and money; yet these phases are critical and un-avoidable.

The extensive use of resources spent in testing phases are much questioned within product development teams, starting from being seen as the delay point, up to being considered as something that adds no real value to the product.

In world in of ever increasing customer expectations in which intense pressure from competition and continuous shortening of product and technologies lifespan, only the most efficient and effective companies will be able to keep on pace.

Companies can not only rely on having great products, they must have them fast, and they must have them cheap. Companies are able to perform quick and lean testing phases in their product development processes will survive, but those who in addition manage to add value to their products through their testing activities will achieve success.

This thesis and the work related to it focus on achieving a quick, lean, and value adding development and validation phases from the point of view of a growing organization of the automotive industry.

1.1 – Testing Needs for the Next Development Organization

As described by Wheelwright and Clark, product development today is driven toward maximizing Quality, Efficiency, and Speed.

In the case of the automotive industry, faster time to market products has become an essential need since companies face rapid changing markets where the life spans of products and technologies gets shorter generation after generation. In addiction the intense pressure that oil prices have put over the shoulder of the automotive industry, drives the engineering efforts to develop new vehicle faster, applying technologies and solutions for efficient and clean operation under a heavy and forced sense or urgency.

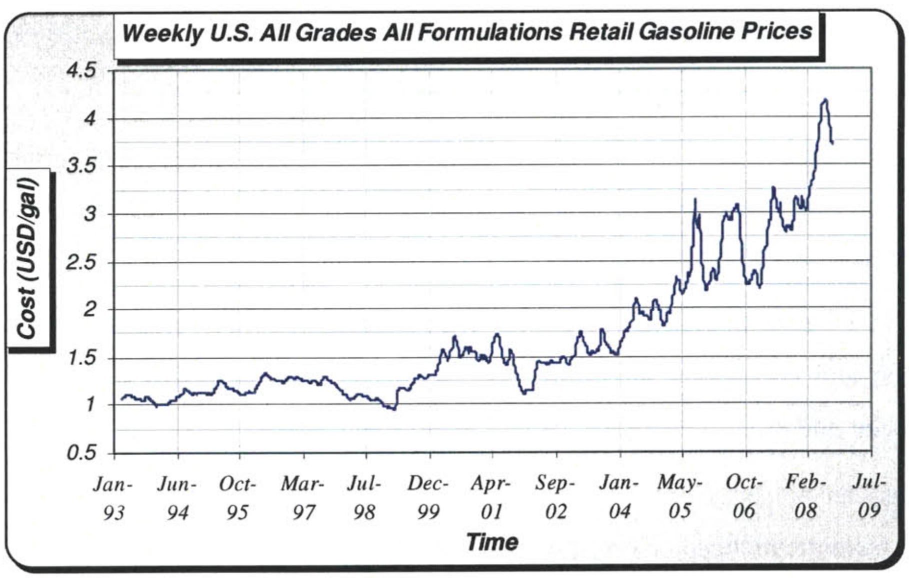

Figure 1-1: Weekly U.S. Retail Gasoline Prices (Energy Information Administration, DoE, 2008)

This environment is subject also to a heavy competition among manufacturers, and drives the industry toward maximizing their effectiveness and efficiency toward product development. Verification and validation of products has to be improved in order to deliver designs with increased robustness, but also has to be performed faster and with fewer resources.

The market demands better products to cover more specific needs, and the customer has evolved to be able to recognize the difference between product attributes. This puts increased pressure into product development organizations to develop more versatile products with increased reliability.

The automotive industry faces a path in which designs are produced in smaller volumes with decreasing life spans, demanding continuous improvements in effectiveness and efficiency in the use or their scarce resources. As the life of the design and volumes in which it is produced get reduced, the resources available for creating and developing products are reduced, this creates the need of a fast and efficient development and validation phase within the product development process.

The product development organizations who manage their test and development operation outstandingly, will achieve a critical competitive advantage.

1.2 – Current Testing Focus for Emerging Organizations

A product development organization responsible for the development of complex system, makes use of testing for the development and validation of the products that will be offered within the market, and as they grow they evolve through a series of phases in which testing is used with special focus for each. In most cases the purpose from which testing is performed could be classified as follows, but not necessary in the same order:

- Build-Up[1] Units Validation
- Local Product Control
- Local On–going Product Development
- Product Design & Development

1.2.1 – Build-Up Units Validation

This is the simplest testing scope a product development organization uses, and its most commonly found in large organizations with more than one branch with design authority and with ownership of a platform or product family. It is the testing scope generated at a local organization that will be introducing a foreign vehicle into their market.

[1] In the automotive industry the term Build-Up refers to a vehicle of foreign design to be sold at the local market, where there was no involvement at all during the development of the product.

The purpose of the Build-Up unit testing is to verify that the product is able to comply with the special characteristics of the local region, as it was designed by an entity for a different region. This kind of testing is also performed for regulatory purposes, as the local organization needs to provide compliance data of the design against local regulations. Marketing and benchmarking are purposes from which this kind of testing is used, but are not exclusive, and are performed as general testing of any type like overall performance measurements.

Examples of special characteristics to validate on a Build-Up unit are the temperature and altitude profiles of the local market, which could be more demanding to the systems (e.g. power train cooling, brakes systems, engine calibration, etc) of the vehicle and may require a fix from the designer.

Regulatory compliance data that a manufacturer has to provide to a governmental authority in order to be allowed to sell a vehicle, we can find examples as tailpipe & evaporative emissions, brake stopping distance, external noise, and crash safety testing among others. In cases in which the requesting authority accepts test results certificates performed by the organization that designed and developed the vehicle, this testing is avoided.

Due the nature of the product that is being tested, the build-up validation requires significant less equipment and facilities since there is no attribute development testing, and many tests can be waived by the previous work performed by another branch in an organization. It performance is not limited by the size or resources of an organization, and could be present at any moment a product or vehicle is imported into a market.

A common characteristics of the evaluations performed, is the amount of subjective data and the importance of it in comparison to the objective data that can be measured. One of the key elements of this type of testing is the evaluation of the product in the market environment, against products from the competition.

1.2.2 – Local Product Control

Quality control could be the best way to describe the testing purpose of a Local Product Control team. These teams can be found in large and well established organizations, but are more a characteristic phase of a growing organization, and placed within the boundaries of manufacturing plants.

The LPC is often the start up path for a new product development organization in an emerging market, and these opportunities appear within manufacturing locations of well established organization outside of their countries. Typically when a product is launched in a foreign country a local team is assembled in order to take care of the running product, making quality monitoring for continuous improvements as well as providing data to the main organization when an issue is detected.

The testing activities within a LPC environment are performed as a quality check for the production unit being built at the facility, and as well to track and build quality statistics. Fixes or minor changes to the product are also tested by these entities, when they are developed as continuous actions improvements to the production line, and are supported by the engineering entity with ownership of the product design.

Product development organizations in emerging markets may start and grow from this position and require formal testing teams and facilities. In the automotive industry we can see some examples as how the Plant Vehicle Teams (PVT) expands to form product development organizations and evolve their testing operations such as the following countries:

- Venezuela – The automotive manufacturing operations and Ford of Venezuela required testing for the local product being assembled locally as a PVT as well as performing Build-Up validation from other vehicles, forming a Local Vehicle Team (LVT). LVT performs testing operations following corporate and local testing procedures within independent garage and instrumentation facilities, aligned to the size of their current engineering activities.

- Mexico – Following a similar path as Venezuela, Mexico required PVT and LVT testing activities grouped as the Test and Development department. As an emerging market, a Mexican team was assembled for the testing needs that their

LVTs required for the local production vehicles as well as for the evaluation, and homologation[1] of imported units.

The involvement in different programs and a growth in the engineering activities performed by the organization drove the Test & Development department to reaching capabilities for incremental/derivatives product development projects, following paths for programs with increased complexity and responsibility.

- Brazil – Being a low cost country with large markets demanding their products, Brazil had the advantage of a location far away from the design headquarters of the main company. This combination provided an accelerated growth demand for the testing capabilities of the South American organization, since the large distance to the nearest corporate facility made a favorable business equation of building local proving grounds. The corporate local teams developing the product fostered the expertise and facilities for a Brazilian semi – independent product development team.

1.2.3 – Local On–going Product Development

A product development organization who starts its path as guardian of a product being done in local manufacturing plant, develops the capability of doing incremental or on–going development, directing the engineering efforts toward enhancements, cost reductions, add-on features, and product continuous quality improvement.

The testing needs for this kind of organization become more complex that just the default needs from a Build-up validation or the ones in a LPC team, since the testing focus moves from verification or measurement to development work, requiring more specialized and capable equipment.

Testing complexity arises by the need of understanding the behavior of a system rather than only is response. The development testing requires to read and understand the data profiles along the system components and interactions, during the whole intercourse of a test, and there the need of high capacity (number and types of sensors available) and

[1] Homologation is the process in which a vehicle is tested for validation against local regulations as well as for internal product information. Tests included are acceleration performance testing, brake stopping distance, exterior noise levels, and tailpipe emissions testing.

resolution (quality of data and frequency in which is taken) equipment is needed in order to record the data generated during each test.

As well as the equipment needed by a product development entity needs to have higher capabilities, such the facilities in which test are going to be conducted. In order to develop a product, even if it is an enhancement, a derivative, or a cost saving action; there is a need to test under a wide range of circumstances that require special facilities such as locations or testing laboratories.

Equipment capable of keeping pace with the testing needs of a product development organization is expensive, and must evolve as the capability of the organization grows. Testing laboratories represents a much larger investment toward the test and development needs within a product development process, and often a growing team is unable to handle such expenses. Instead a growing product development organization makes use of creativity finds alternatives for the use of expensive laboratories and facilities such as test tracks, by finding locations with similar conditions to the ones represented in a laboratory environment. Locations that substitute the use of complex test track systems and laboratories often are public road and sites, where actual customers may use the product and find an issue.

Growing product develop organizations that are branches of well developed transnational company, can overcome the difficulties of performing very specialized testing that requires very focused equipment and test facilities by sending systems and subsystems to test to the corporate locations, avoiding the costs except for the prototype costs and the shipping expenses.

In the same way, a growing product development organization can use the resource of outsourcing their testing needs in order to cover any lack of capability due to lack of the expertise, equipment, or facility needed to develop and validate a system or subsystem.

Although a starting product development organization will face many challenges in order to perform development and validation testing for systems and subsystems, the LOPD cases have the clear advantage of having focus on the local product being manufactured. This advantage reduces the scope of the testing equipment and facilities required, since testing would be oriented toward specific a specific system or product, rather than being ready for a whole spectrum of product families. Incremental or derivatives projects also have the advantage for the growing organizations, of having products with carryover[1] features that are not required to be tested or that minimal effort has to be invested.

An important milestone that signals this stage within a product development organization, is the beginning of CAE implementation, experimenting, and testing models against the work done at the test track as a mean of going deeper on the analysis of a failure mode.

1.2.4 – Product Design & Development

A product development organization that has reaches beyond the level of incremental and derivative projects could be considered to have reached maturity. At this level the product development organization is capable of delivering projects of great complexity by its own, in which every system and component is developed and tailored in order to create product platforms, families, as well as derivatives.

Testing in a fully developed product organization is extensive in the different kinds of test needed to apply, and complex in the way that systems and components must be tested to its deeper roots. From the materials used in the final product o a fully integrated system, everything will require to be tested at some point.

Recent product development methodologies makes emphasis on the reuse of platforms, systems, and component; but still each one of those parts will face different interactions and conditions that will require being verified and validated.

At this point the usage of vehicle level testing is carefully balanced with laboratory testing and CAE models analysis in order to achieve an optimal development costs and time to market; without compromising the reliability of the products being delivered.

When a product development organization has grown to reach this point, it has covered the testing needs of all the previous phases that have been previously mentioned. But as they grow and change their testing needs don't shift and only grow, since they continue performing the entire tasks. The system must not only be maintained, it must be

[1] A carryover feature refers to a system, subsystem, or component that has been used in previous models of the product and that are currently within in use by either the main product or a derivative. This condition avoids extensive testing to be performed on this system but rather just a light verification focused toward it interaction within a larger system.

understood, it must be created; and through testing knowledge is created and understanding is attained.

The creation and development of new products add a new point of consideration, which is the confidentiality of the product itself. This need for confidentiality drives testing to confined and controlled spaces, representing considerable investments for the creation of proving grounds and laboratory facilities capable of delivering all the testing needs required.

The scale of facilities needed is also reflected on the instrumentation and measuring equipment required in order to be able to measure with even greater detail and accuracy.

Since fully covering the capability to test every since system and component, as well as their interactions would represent an unfeasible scenario, product development organizations rely on suppliers and external facilities which develop the specialization required to cover the needs of several customers by providing testing and development services which often are managed jointly with the production of the system or part itself.

A growing product development organization is able to design and develop their firm products with total control, but also they could be a growing branch of a larger company developing projects where they don't have complete control over the design. An example of those cases could be when a product development organization works on a platform from which they don't have design authority, or when the final product is assembled in a different location where another organization has control of the product.

This limitation, in which the development is constrained within a platform that can't be changed, or that changes are unacceptable due the local assembly process, is faced by organizations which start reaching this point mostly when the design work is on derivatives of a base or main program being developed.

As it is depicted in Figure 1-2, growing product development organizations in emerging markets evolves through a series of phases where complexity grows exponentially. The ability to manage and cover their testing needs, often is the bar that limits the level of design responsibility that an organization will be able to manage in a reliable path.

1.3 – Testing Capability as a Product Development Growth Driver

A product development organization (PDO) naturally seeks a path for increasing their capability and attains greater responsibility in the design and development of products and systems. In their growth path, the PDO will be on quest of knowledge and expertise for their designer and developers, in order to be able to take responsibility of continuously increasing complexity projects.

Design and release engineering is a capability that can be reached through training and experience, a capability that can be deployed and used through the distance, offering the design teams to operate remotely and be able to maximize the efficiency and effectiveness of the groups through concurrent engineering efforts, selecting the best and optimal locations to allocate resources.

Although engineering capability can be reached through time in position, exposure to projects, training, as well as sourcing and hiring; the organization must be focused with a system mindset and keep track of the global needs it will face, as the complexity of every new project demands. Therefore for any product development organization, the development of attributes as systems requires careful study and analysis of interactions among systems and components, and how each system will contribute to the feeling of quality and satisfaction to the final users.

Development is a different capability since it not only requires training and exposure through experience, but its deployment is dependent on the actual facilities and equipment available to use by the product development organization.

Development is an activity that relies on experimentation through physical or virtual environments, with the purpose of refining the interactions among designed components and systems, to deliver an expected level of performance and quality to the user.

Often, the need to focus also on the growth and training of development activities may not be correctly identified as a product development organization evolves due the high content of carryover systems that are involved, and the availability of supplier that perform testing of their components products by themselves.

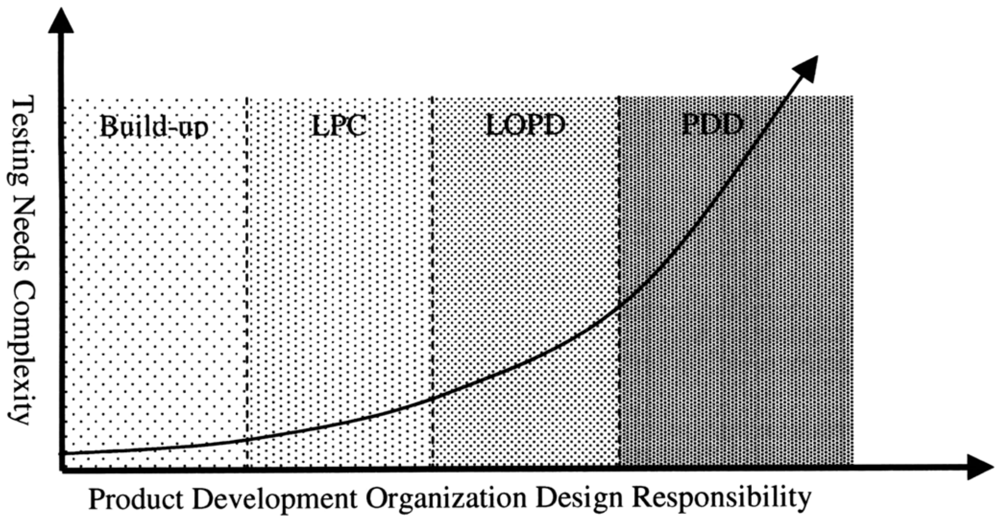

Figure 1-2: Testing needs complexity in a product development organization

- The increased usage of carryover systems, or use of components or subsystems already in use by similar products, vehicles, or derivatives; reduce the need of complex development testing for the small product development organization since assumptions of reduced performance impact are made due the lack of significant changes within the system or its components as a whole.
- Design & Release engineering may rely on full service supplier, or suppliers with enough capability to develop and validate components and subsystems, eliminating the need for a young product development office of creating the testing infrastructure to test their designs, or the new configurations being created.

Figure 1-2 briefly shows how the complexity of the testing needs for a PDO increase as it evolves and acquires greater design responsibility. While performing the validation of Build-up units, testing is reduced to homologation and for the LPC section it relies on the assurance that the quality issues of on production items are solved as well as the right implementation of continuous improvement and cost reduction actions.

The LOPD phase for a product development organization starts a ramp up of the testing needs that apply to the project carried, since the path of doing incremental design to new derivatives from an already designed product involve the verification and development of product or vehicle attributes in order to deliver the expected performance.

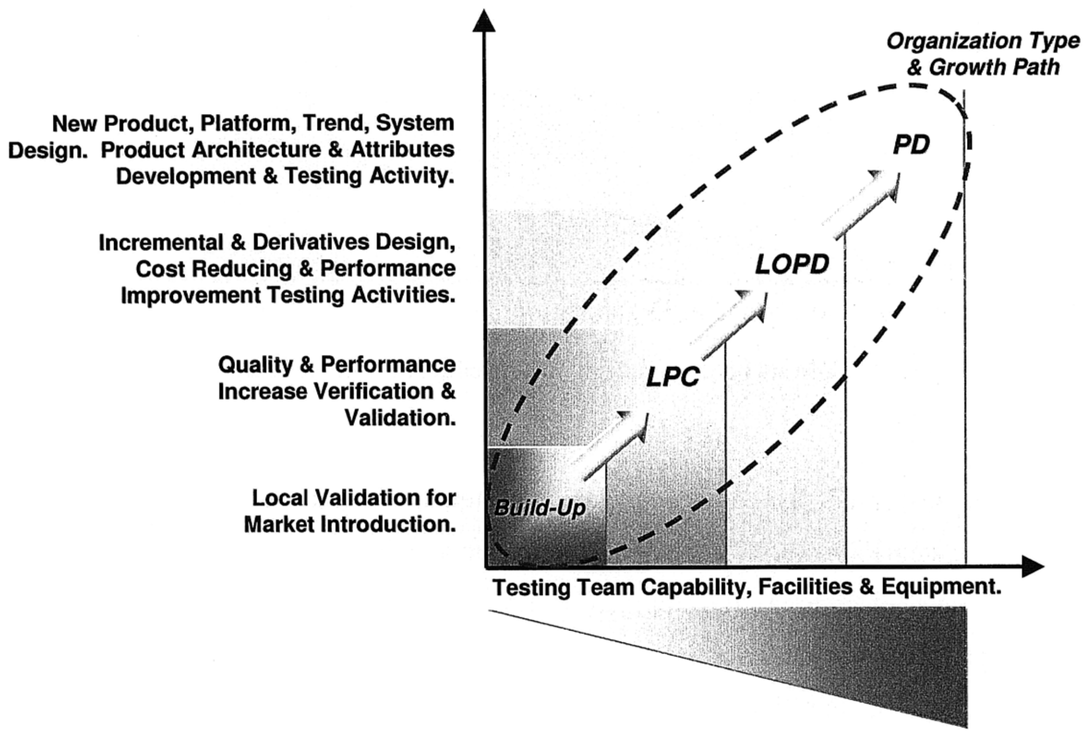

Figure 1-3: General responsibilities for PDO growth phases with increased testing complexity

From Figure 1-2 and Figure 1-3 it is possible to infer that the ability for a product development organization to make it through and LOPD phase and reach a PDD state of growth, is strictly dependent on the availability and capacity of complex testing, that support the need of advanced engineering design not based just on incremental changes or derivatives of existing platforms.

Ford of Mexico Test & Development has the mission to be the entity that supports the growth of the Product Development office, as an engineering problem solving entity that delivers real time design solutions with effective and efficient development and sign–off testing.

The Ford of Mexico Product Development growth requires a solid platform for performing design testing & development. Being able to perform development testing with consistency, quality, and reduced timing and cost; represents a critical competitive advantage for obtaining new projects with increased engineering responsibility.

1.4 – Thesis Motivation

The motivation of this thesis work is mainly based on a collection of feelings and ideas, experience gathered from a product development adventure that goes through times of change, when new needs require unprecedented effort to cover them, and a team may choose to follow the known path or explore the possibilities that a new generation brings.

The product design and development process is a the quest that a team undertakes in order to satisfy a need of the market, and within this quest there is a part that integrates its components into a one very first whole system, a system to test and develop until it is ready to fulfill its purpose. This quest is a path that rarely is as could be read on books, and rather it presents unexpected issues that require teams with unprecedented abilities and enthusiasm to solve them.

I have had the privilege to participate within several project programs, working with the teams that faced the issues, and created solutions that saved several projects from failure. And most surprisingly, I have noticed that issues arise although the team follows an established process in the best way they can, but the solutions that save the day often comes from paths that are out from the established set of rules and documented processes.

Experience has brought to me several questions about why following the procedure still fails to avoid the generation of drastic technical issues, and why the solution may come from going beyond the standard rules and practices. Often the out of process solutions are disguised in order not to be noticed, or not being scolded by acting out procedure, but it is well known that the development team needs those heroes that take the risk of going beyond to bring back the project into track.

This is the motivation for this thesis work, to understand the gaps or wholes during the development and testing work in the product development process, that brings the need of outlaws, firefighters, or just engineers that find solutions outside from the original set.

My motivation is to seek and find a better way to develop and test products, a way to reduce its costs, a path to have a robust delivery of a product in accordance to the customer expectations.

Which is the real gap that separates a growing organization from a well established one, rather than experience and facilities? Which is the gap that separates our current organizations from a global world class one? Are we not following all the knowledge and experience that formed or product development systems, or time has demanded an update over our existing architectures?

We are able to see the situations described above in the development of every system, from planes and automobiles, to the simplest ones as toys or the packaging that contains it. Every system has to be tested, and every system has a degree of complexity that can create numerous interactions which also require to be validated.

As Flannery O'Connor once said "Everything that rises must converge" and all this situations found in life converge to a very unique situation that gives existence to this opportunity, to the opportunity of conducting this work from the start.

Taking as an example the product development office of Ford of Mexico, a young engineering office, has received the appointment/opportunity to increase their capabilities and support the global organization. This is a problem that requires an evolution step from this young and small organization, how to achieve it and at the same time contributing to the general improvement of the process as a whole, is a task that gladly motivates this research, with enthusiasm and vision.

I am happy to be able to live these times of change, and to have an opportunity to promote a change, so much needed by the existing structures to reach the level required by the current levels of competitiveness. But beyond happy, I feel complete because I can look around and see a team pursuing the same goal, since this is a combined effort in which several students research over different PDD phases over their thesis work, to deliver a joint effort toward the new product development system that modern times demand.

In modern times we no longer hear about the great auto industry, the one that raises cities and countries into well being is a long forgotten memory and now they strive in the border to extinction. These are times of sacrifice but also times of gathering, interesting times to live on, times of facing the challenge to join within a great team at industry, at school, across life. Gladly I accept and take my part with other friends, motivated to make thing happen, and to bring change and hope.

From those who trusted me, from the ones that go front and behind me in this great effort, I found the motivation to search a path to have a greater, stronger organization that not only contributes to the survival of the whole company but also shows that things can always be done better.

1.5 – Thesis Reason & Need

The product development process is an universe so vast and deep that is difficult to concentrate it under the development work of one person alone, further more covering just a generic point of view of the product development process is not enough since our work requires the focus my company, industry, and country needs to survive these time and jump into success.

Through a series of thesis, a group of student make the team effort to develop the way and strategy in which Mexico as a small organization should pursue in order grow with excellence taking the best of an small organization and developing with excellence as a large one.

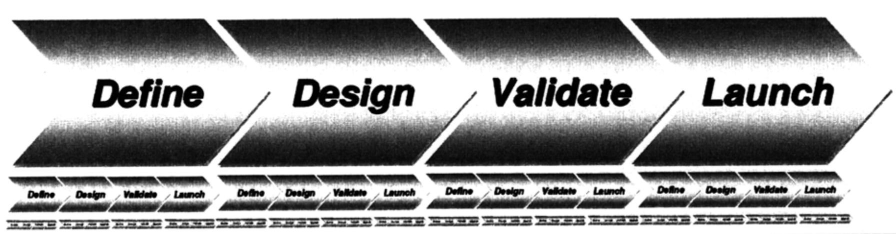

Figure 1-4: Iterative Four Phases within Phases of One Model (Endo, 2008)

Even by dividing the product development process into it most comprehensive phases (Define, Design, Validate, Launch) and each one of this phases is a whole process by its own, which can be divided again in the same four phases such as it is described in Figure 1-4 by (Endo, 2008).

The verification and validation phase is a block within the Product Development Process, with interactions and dependencies among all the other phases. Therefore this work starts within the define phase, building a strong system culture and education, in order to derive the requirements and specifications that a system must comply and that represent the core in which the development and validation will focus. The methodology will derive the requirements and the way to test them, joining lessons learned as well as engineering system specifications.

Integration is the key to effective product development systems and as part of an integrated team, each one of these students has experience and expertise in each of the four groups in which the Product Development Process is separated as phases, gathering into teams that could cover at least in the most essential form, each one of these phases for the construction of one big project as a whole. Each one of us fulfils an unique chapter, and each one of us provides a point of view from a different angle.

Product Development excellence relies not only in gaining engineering expertise, but also to gain the capability to support the needs of the engineers in order to develop a reliable system able to work together and perform as intended. The whole process demands excellence in every phase and therefore I feel fortunate to be able to apply my experience into the development of the validation phase of the PDP, helping to build a systemic project along with other alumni thesis and integration tools.

The focus of this work is for implementation on the automotive industry at secondary markets, with the purpose of achieving a faster and more robust, and reducing the resources required for vehicle validation.

Ford of Mexico testing and development operations must evolve in order to achieve world class performance, gaining trust and confidence, and become an alternative of choice for programs development and sign–off.

1.6 – Thesis Objective

The motivation for this thesis work describes the origins to create a thesis objective, and it is based on many questions and discussions gathered through experience and research, all merging into the single statement: "How could we make this better"

Based on a collection of feelings and ideas that was built through experience generated across years and project within a product development organization, I try to set and objective answering questions found in a young organization that not only tries to excel in its duties, but also to contribute to the success of a company as a whole. These questions are as follows:

- Why exhaustive development work against an established library of requirements, fail to deliver a winner product?
- Why well established experience learning processes fail the avoidance of failures and rework generation during the development and validation process?
- Why after successfully validating a vehicle or system against requirements, we still have failures and claims on the field?
- Which is the path that a growing organization needs to follow through its evolution steps, and how to identify it real state?
- Which are the elements in the development and validation testing phases within the product development process, which inhibits or slows down the journey to a world class Product Development Organization?

The answer to this questions lead to the following thesis objective: To define and design the process elements for developing the testing and validation phase within a product development system, for the successful grow of an evolving organization.

To fulfill this objective, tools with a systemic approach are needed but in the same way identifying all the well known good practices applicable to implement them will be essential.

As summary, the key point for creating this work is being able to design Fast & Responsive Development & Validation cycles for shortening time to market with optimal use of resources.

1.7 – Thesis Hypotheses

Automotive product development literature or product development studies in general, make just a brief approach to the validation phase within the product development process, stating it as a section in which concepts are tested in order to become refined designs. Although testing is usually a path of repetitive iteration until a final design is released, perfect planning and execution helps to reduce those iterations and provide a faster and cheaper product development process.

Although there is not an assertive and detailed path for improving the quality and effectiveness of the testing process and at the same time reducing it cost and time required to perform it, it is clear that a key element for achieving so relies by improving the up-front process for the development and validation areas, which is the analysis of the requirements to verify and validate, as well as the correct definition of test plans and prototypes.

From a systems engineering perspective, testing phases can be grouped in three activities that are linked together:

- Requirements Analysis – What do we want to test and what do we want to find?
- Development and Optimization – What will be changed, tuned, or adjusted in order to get closer or exceed what our requirements and targets depict?
- Verification and Validation Process – How are we testing toward requirements and targets? How much are we spending on prototypes and testing?

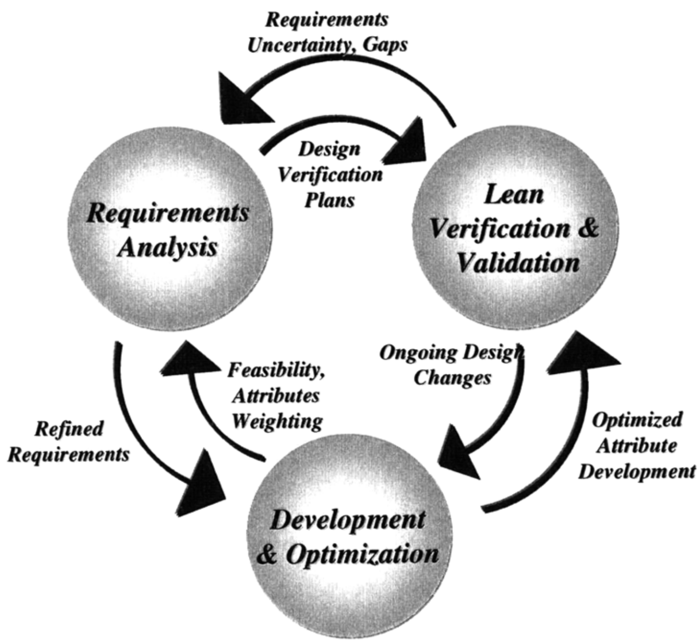

Figure 1-5: Elements Interaction of the Development & Validation System

These activities are closely related and together they define the full development and validation process as it is shown in Figure 1-5. From Figure 1-4 we can distinguish that at the top level, the validation phase from the product development process receives input from both the define phase as well as from the design phase. These previous phases make the requirements analysis the initial step within the validation phase.

The requirements analysis phase is the step where the current level of design and program definition is analyzed in order to determine against which targets and requirements the system will be developed and tested toward. In the same way, a careful study of previous lessons learned should be applied toward the interaction of components into systems and the delivery of system attributes.

The analysis of requirements as a essential part of the system integration process gives as result the target from which each system attribute and functional area will develop their section in order to fulfill the over all expectations of the product, as well as the design verification plan (DVP) defining the testing work to verify the product reliability and performance.

These results are the input for the Development & Optimization activities which performs a systemic multidisciplinary design optimization to deliver the attributes that the final customer will feel with the product, in this case the vehicle, attributes such as performance, NVH, vehicle dynamics, among others.

The results from these two previous mentioned activities feed the verification and validation portion, which acts as a final sign-off entity, documenting and verifying that every aspect written under the vehicle/program/system definition and assumption has been correctly addressed.

As shown in Figure 1-5, the current validation phase within the PDD of a established and growing organization is affected by iterative loops created by incorrect selection of requirements and targets that create gaps or demand unfeasible system expectations or interactions. At the same time, a never ending development process is carried away in order to pursue exhaustively perfection, and iterating through infinite design changes and modifications produced by incorrect initial assumptions or new/changed requirements that are found on the fly.

The following three hypothesis attempt to describe the actions that could minimize the presence of the previously described loops, shifting from a D&V phase shown in Figure 1-5 to the one envisioned in Figure 1-6.

- An efficient and effective requirements analysis process that delivers refined/smart customer/appeal driven requirements and a target that incorporates correctly all previous knowledge and integrates every system interaction, delivers the elements to avoid unnecessary definition iterations.
- A lean testing process is not only the one that performs verification with an optimized use of resources, but the one that performs the right exact testing both in procedure and platform (virtual, physical) to analyze the right elements to deliver a requirement or target. Lean design verification plans involve performing the right tests for a system, searching the best platform to achieve them.
- Development & Validation work of system level design can be optimized by pursuing targets and requirements until a certain level of confidence is

reached rather that exhausting time and resources, under specific acceptance criteria.

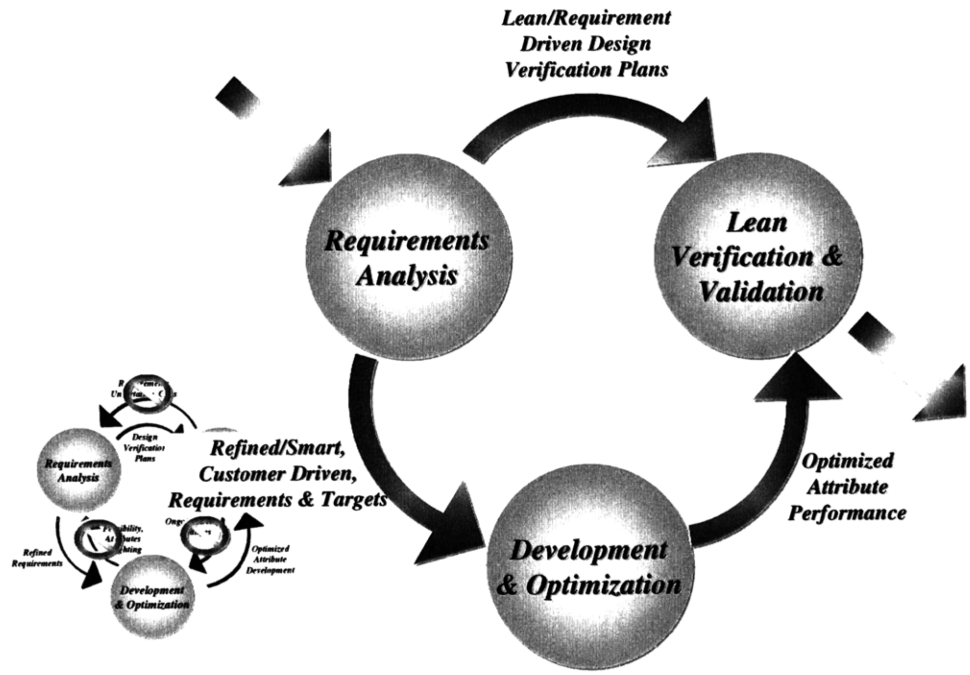

Figure 1-6: Vision of an improved FoM PD Development & Validation phase

Considering a general product development system such as the one described in Figure 1-4, an improved system development and validation phase would present a flow such as shown in Figure 1-6 where unnecessary process iterations are removed aiming for the character elements of an ideal Product Development System identified in (Aguirre, 2008).

Although the development process of a system is an iterative design, a perfect execution mindset can be applied in the analysis of the requirements to apply as well as in the performances of testing procedures.

A nimble decision process is key for the selection and analysis of requirements and targets, in order to be able to optimize and develop correctly reaching point that are probabilistically acceptable, or when risk has been reduced to a minimum.

The remaining two character elements described by the author depict the grow path for emerging organization that desire a world class validation phase in their process, since

Excellence in Communication is the character element that will correctly define the roles and responsibilities of each development team and that will enhance the results of their interaction.

Finally the flexibility in capabilities of a growing organization defines it path for success, being able to address correctly it resources to where most value is added, in a shifting process to fit with the ever changing characteristics and environment controlled by well developed and elderly organizations.

1.8 – Information Collection & Analysis Methodology

In order to be able to correctly support the hypothesis for improving validation phase of the product development system in a general perspective for the automotive industry with focus to product development organization in emerging markets, several sources of information and insights needed to be analyzed and pursued into it roots.

Although the academic data and book about product design and development is vast, even for the automotive industry, the validation phase of any product development system is still a topic seen from a general point of view. This knowledge gathered with best practices followed not only in the automotive industry but in general will serve as a platform where to build and focus in the environment of a growing organization.

Over the platform built through covering the existing literature, the characteristics and backgrounds of several regions in conjunction with tools, lectures, and guidance received from the System Design and Management program; will build the architecture over which an engineering organization would be recommended to be directed in order to increase its rate of success as it grows both in size and responsibility complexity.

In order to gather data from each region interviews were conducted with the teams in charge of conducting development and validation testing, as well as with the teams that performs planning and integration for the testing to be performed for each project. Each region was analyzed based on their testing capability independently from the engineering level, taking into account their location (market characteristics), size and development responsibilities; revealing best practices and patterns, which help to identify a growth path enabling an organization to strategically plan its growth.

Best practices were studied not only from the automotive industry to give a wider scope to the lessons learned that can be implemented, wide as the differences between an automotive industry such as Ford/Toyota and a software and electronics corporation such as Microsoft/Hewlett Packard.

Analysis of the requirements definition and validation processes in lead user companies, through the review of papers, literature, and thesis; regarding those themes in other companies.

It is clear that Brazil, Mexico, Venezuela, and the United States; have different levels of testing equipment and facilities although they all have a local product development office.

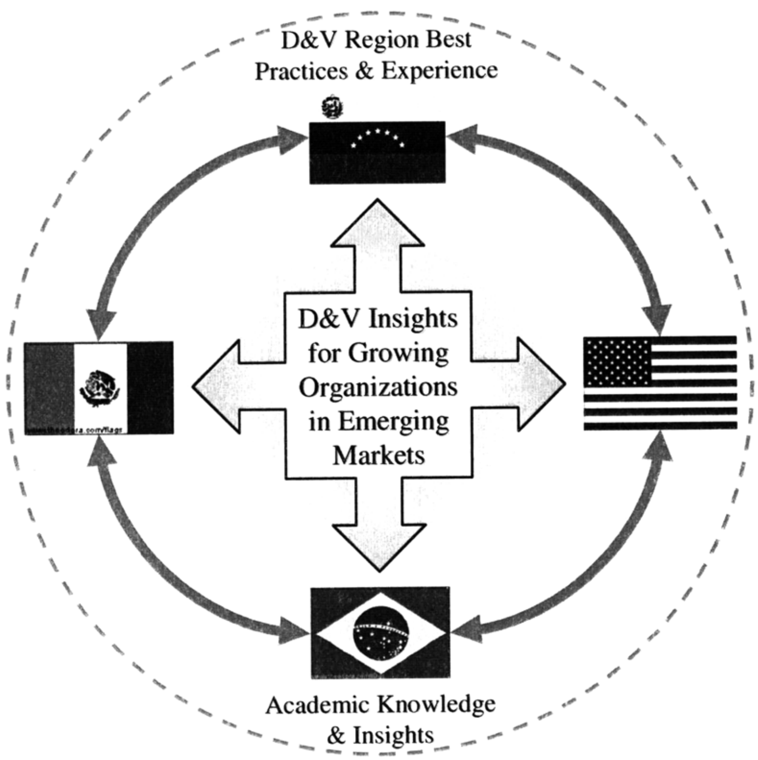

Figure 1-7: High Level Thesis Architecture and Research Sources

Trips to locations and working experience across the development and validation test work from the countries shown in Figure 1-7, gave a detailed vision of several product development organizations in different growth stages, and their insights serve as a path to foresee the challenges to be found during a growth path.

Application of System Design and Management Methodologies such as Design Structure Matrix as well as Value Stream Mapping through the development and validation process, with the objective of have a clear understanding and identify the improvement opportunities as well as to model new process and information flow alternatives.

Close work has been done with the teams from Brazil and Venezuela, as years of working experience had helped to build relationships, and the tools from the SDM curriculum and opportunities served to bind them.

In the case of Mexico the sponsorship, championship, and guidance from the Product Development office gave richness to the analysis as well as a space for tryouts and experimentation as this organization evolves.

Although the United States is not considered a growing organization and neither an emerging market, it is vital since it sets the boundaries for a fully developed development and validations organization, and sets the example of what practices and structures need to be replicated and which ones to be avoided. Being able to learn through the last two years from this organization, gave further insight of a whole development and validation structure by shifting among several assignments through functional testing, attribute development, and test strategy and planning teams.

Chapter 2

=== Development and Validation Concepts ===

"The purpose of war is not battle but victory."

or:

The purpose of analysis is not modeling but understanding.

– Sun Tsu, The Art of War, ca 500 BC

2. – Development and Validation Concepts

In the automotive industry the development and verification are within the phases are the most resource consuming in the Product Development Process being critical and unavoidable, both phases involve intensive testing activities but each one sees the product from a very different point of view and ever since they have to be managed differently.

This chapter has the intention to define and describe in a general sense the development and validation process, making a systemic approach from the several points of view available in current literature about these themes in conjunction with insights and experience of these processes from experience gathered within the automotive industry. Although the term development implies a wider and deeper process over which an design evolves, within this work we will be referring to the development testing work.

From a general point of view development and validation testing is the process in a PDO generates information that guaranties that the designed system meets the requirements toward it was conceived.

As described by Endo 2008, each phase within the product development process has its own validation step, following the same purpose of ensuring that all the current progress is performing as it is intended to do so or ensuring that it will with a certain degree of confidence. As an example of validation sub phases across a PDD system, are the milestones across the timing of a program which act as checkpoints validating the current progress done by the team and their feasibility to complete the program against corporate and financial targets and requirements. Some activities where validation sub phase are in charge of can be:

- Validation of timing and project alternatives
- Validation of concepts
- Validation of feasibility of program targets and methods
- Validation of feasibility of project delivery under financial considerations
- Validation of design process
- Validation of systems and component functionality and attributes performance
- Validation of manufacturing process, tools, and equipment.

2.1. – D&V Testing Form and Function

Development and validation are themes that involve intensive testing, the embodiment and discovery which become the source of knowledge over which the differentiation of engineering organizations among others is founded.

In order to correctly differentiate the essence between development and validation testing, it is convenient to first develop the concept of the root activity which is also the root of the common misunderstanding that avoids the proper handling of teams that performs product development testing within a PDO.

The concept of a system is an attribute that links the internal function of the whole system [to its form, upon which the architecture of a testing system has to be developed. As described by Crawley (2008), every system needs to be designed correctly from it foundations, developing an architecture where to build it that needs to be able to support it as it evolves and grows in complexity.

2.1.1. – D&V Testing Function & Goals

According to Rechtin & Maier, the structure of a system must resemble its functional structure. The function is an element of the concept and attribute of the system that set the goals that need to be completed by the system, and describes the end product or deliverable in its most basic form.

$$Function = Process + Operand$$ (Crawley, 2008)

Equation 2-1

Expressing the function as described by Crawley in Equation 2-1 of a test system as a Solution Neutral Statement[1] requires defining the operands of the system which in this case are the design solutions for creating a product which could be from a component of a full system of interacting components. In the same way process must be defined which in conjunction with the operands give as a result a function of the system. For a testing system, solution neutral statements within a PDP for the described operands are defined as follows:

- Verifying Design Solutions (and not: Testing)
- Comparing / Understanding Design Solutions (and not: Benchmarking)
- Documenting Design Solutions (and not: Reporting)

Examples of operands that a testing system use within a product development process could be from a HVAC for climate control system to a full vehicle.

Having identified the processes (Verifying, Comparing, Understanding, and Documenting) and the operand (Design Solutions) it is possible to state the goals of the testing phase and sub phases of the product development process as:

- Verify design solutions against requirements for needs compliance
- Compare design solutions performance for options reduction
- Understand competition design solutions for implementation or improvement
- Document design solutions for lessons learned collection and product documentation

2.1.2. – D&V Testing Form

The form is a system attribute as well, that enables the delivery of a function being the element over which the system is recognizable by either physical or virtual means (Crawley, 2008). In simpler terms the form of a D&V testing system refers mainly to

[1] As stated by Crawley 2008, the Solution Neutral Statement is the most basic form of expressing function, which must not involve a solution of specific function or form, which focuses on the operand and the attributes of it that need to be changed.

equipment and facilities, elements that make any individual identify the activity that is being performed since there is a explicit function that is carried away.

But beyond what can be seen as elements of form, the form of the system includes no only the physical or virtual elements but the way that they are interconnected. What makes the elements of a system build the form of it is not just the grouping but the interactions that arise among them as it is explained in Equation 2-2.

$$Form = Elements + Estructure \text{ (Crawley, 2008)}$$

Equation 2-2

The structure of the system is a is what completes the form, and becomes the first differentiation among systems that may look alike at first glance, for example development testing is very different from validation testing, and the structure of their form helps to clarify the difference in a very brief way but we will talk later about that difference.

The form, shaped for a specific function, is the part of a system that is operated in order to get a results as well as the portion that is referred as implementation, again in either physical like a laboratory or virtual as a process.

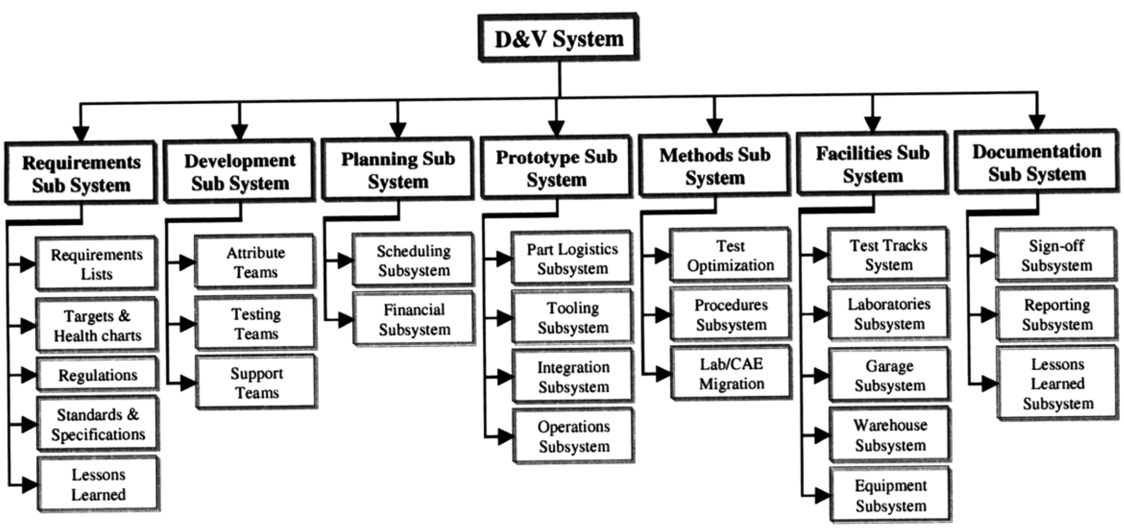

Figure 2-1: D&V System Decomposition

The form of a system is able to be decomposed into finer elements, an exercise done in layers as shown in Figure 2-1 with is called a decomposition view of the system. The system decomposition reveals subsystems and component that conform the main system, but this decomposition may involve a mix of sections that are physical or virtual, but that are interconnected in order to provide the function.

The way that the elements of form could be decomposed and grouped into smaller systems, defines it structure being the driver to derive different functions among systems that may look alike or that use the same elements in order to exist. Each level of decomposition may show new systems with independent functions and form, gathered in a structure for achieving a global function that fulfills the validation phase of a product development system.

Assigning a person responsible to each element in a form decomposition created an organizational chart, and as form defines, it is a diagram that shows the elements in an organization and the structure of it thought the way each person interacts with the others.

As a decomposition view of the form of a D&V system the organizational chart also is described by level, the interaction between an upper level and the one below describes a dependency. Elements within a same level may lack enough information in order to describe the interaction among the, which often can lead to wrong assumptions of independency among elements with high interaction for the correct delivery of system function.

There is a common problem within organizations, which relies in the usage of "form" to describe themselves and the way the PDO operates. Although we have not explained the difference between development and validation, this confusion of analyzing a system by its form leads to management and resource allocation confusion especially within young and growing PDO that present overlapping phases within their Product Development systems.

Although the form can be optimized for being the best for a system, it may be only the best for that system in particular and not for others. Especially among well developed systems, it form may not suit systems under development or growth ant therefore mirror images of a larger system may not apply successfully.

2.2. – D&V Testing Concepts

We have discussed and reviewed the generic form and function of a Development and Validation phase of a product development process, which are attributes of the system. In order to define a system in its simplest way, there is still the need to define how form and function are connected.

Contrary to the definition of form and function the system concept is a mapping rather than an attribute which embodies the principle of operation (function) including an abstraction of form (Crawley, 2008). At this moment we will rationalize the concept of the development and validation phase of the product development system as:

"Evaluation of Functional and Attribute Delivery"

This statement parts from the solution neutral statement of "Evaluating (summarizing) Design Solutions" and maps the specific operating processes with its related form representation which can be decomposed as much as desired until reaching the basic elements.

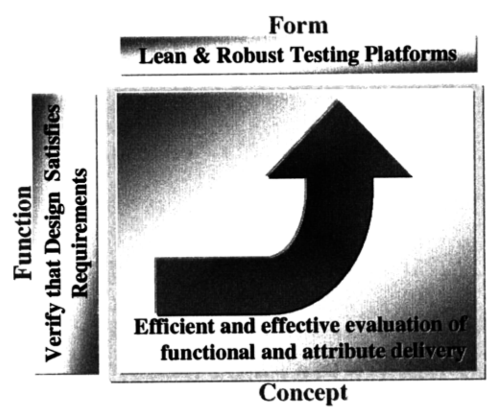

Figure 2-2: D&V Highest Level Concept

The concept of the validation phase includes the vision of how the function "Verifying that the design solutions satisfies the requirements" is going to be accomplished since the process to engineer a form is oriented by describing that the function should be reached in an "Efficient and Effective" way. This concept created by the architect of the organization demand specialization or characterization of the form which in this case is referred as "Lean and Robust".

The concept of the system is a vector that defines not only the magnitude over which a process is accomplished but also the direction. Figure 2-2 show the typical direction that is used during the design of a system being the general convention, but an alternative concept may lead to a different direction as show in Figure 2-3, giving a different connotation of the process being performed.

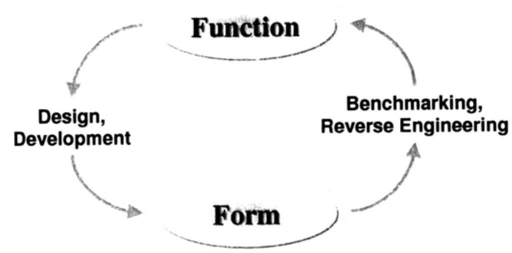

Figure 2-3: Concept defining magnitude and direction of processes

Until this point there is little differentiation yet between Development and Validation testing, since both root from the same function sharing a similar form (from its decomposition view), being governed by a simple concept that applies to both of them.

As analyzed previously, the D&V system has several solution neutral statements as well as goals that derive from them; and the system concept can be divided or have several concepts as its complexity raises in order to allow the system to execute its functions.

Expanding or scoping on the available concepts that are managed within the validation phase of the product development system, making a clearer focus on the main validation phase and not the present within each other phase as described by Endo 2008; the solution neutral statement delivers several main concepts, which make the link between the elements of form conforming the system and the various functions that need to be accomplished into goals.

Figure 2-4 shows clearly the main specific concepts that map the functions demanded of the system to the main elements of form that deliver them. It is important to note that all the elements of form as described before are involved in a successful delivery of function and therefore they are crucial.

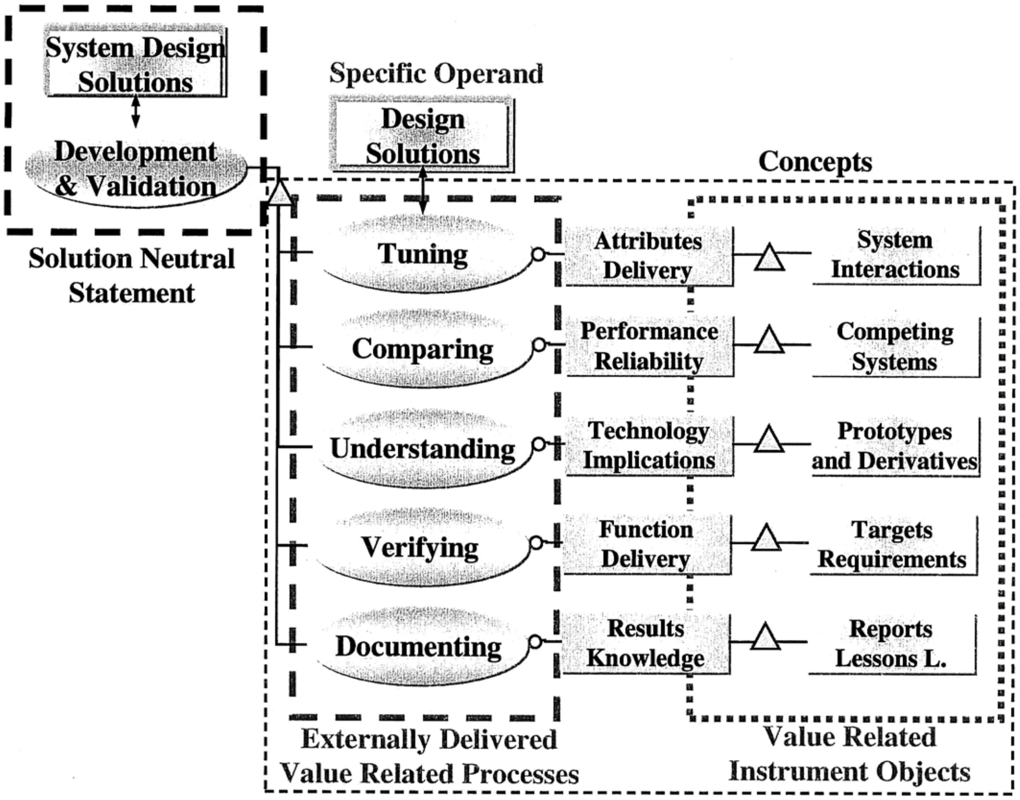

Figure 2-4: D&V Specific Concepts

The proper interaction within the parts is the key to a successful system, rather than focusing on the excellence of each element of form by itself. Based on this statement we join these concepts into a single one as shown in Figure 2-4 into a single element for easiness of consideration (Figure 2-2) remembering that focus need to be made on the whole system and the right interaction of components rather than the excellence of components as stand alone elements.

2.2.1. – D&V Testing Value Relationship

The importance of designing the concept for a D&V system and correctly defining the elements of form and the functions to be achieved relies within the link between these elements and the real world.

The functions that the systems must perform are intended to create a benefit, considered as the original purpose for creating the system which in this case is to prevent the design to fail in the field or in the hands of its users.

As described before, form is the attribute that delivers the function and therefore is the part of the system that carries with the cost, which needs a clearly defined concept in order to design elements of form that reduce the expenditure at the maximum without affecting the delivery of function.

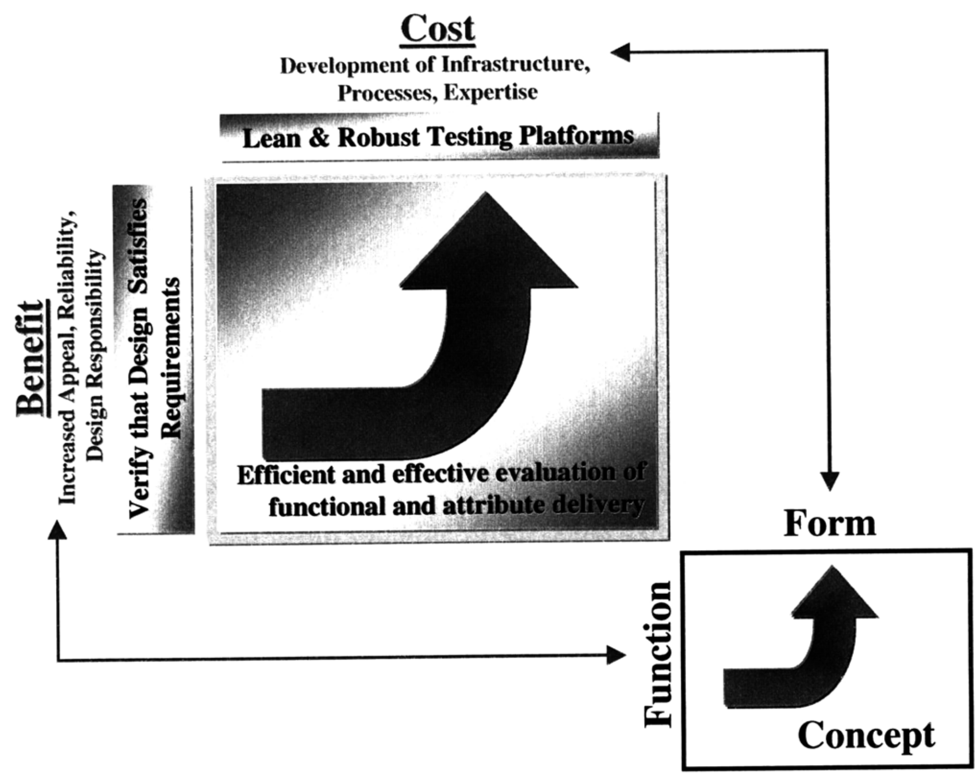

Figure 2-5: Testing System Architecture and Value Relationship

As mentioned by Murman et al, the definition of value is more complex as the function is farther to the end user, which is the case of validation systems. And the value is defined as the worth of the systems, the ratio between the benefit (function) and the cost (form).

2.3. – Development and Validation Testing Differentiation

So far we have seen that development and validation testing activities present the same attributes of function and form, with shared concepts with create in most of the growing organization a differentiation problem. Differentiation problem between testing activities leads organization to a wrong allocation of resources as well as a mix of roles and responsibilities that may prove useful for emerging organization but generates waste and conflict in mature and developed PDOs.

The element that serves to differentiate the development from the validation testing is the context under which the concepts are applied, which later serves to build different architectures in order to attend the specific needs for the stakeholders involved in each phase.

As defined by Crawley, the context is a boundary region that includes the system under design/analysis as well to the other systems and components that are around while the main system is working but not dependent for the system to perform its functions and deliver value (Crawley, 2008). Summarizing, the context is a layer of form above the complete system that helps to define the purpose or reasons for the system.

In the decomposition of form it is possible to see all the elements that interact in order to deliver a function, but within a context the elements that surround the complete system may not necessary interact with it but serve to orient or direct the function of the system to a strategic purpose.

Figure 2-6 describes two different contexts that differentiate the two kinds of testing activities in which we categorize the work done in the validation phase of the product development system.

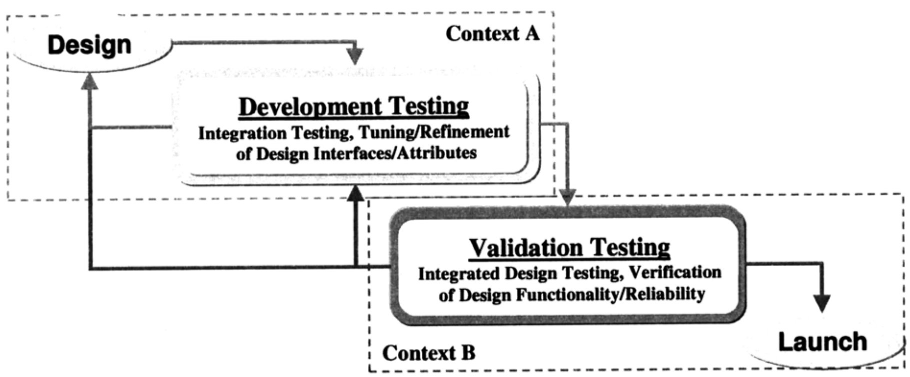

Figure 2-6: Differentiation through context between D&V testing

The context that covers the Development testing system is bounded to design activities which lead its functions toward the integration of the efforts of the different engineering activities, performing tuning and refinement of the interactions of those activities in order to deliver a desired attribute performance.

Validation testing has a similar concept, form and functions; although its functions are directed toward the launch phase of the product development system by focusing its efforts to testing already integrated designs, performing verification of the interactions and functionality of the system in order to deliver a desired reliability performance.

Although testing activities could be confused, the context over which the system is build is key for the correct allocation of resources and the timing at which each system performs across the development of a product.

While development testing is a supporting activity for the design phase of development, validation testing is an enabler for the launch phase of the product development system.

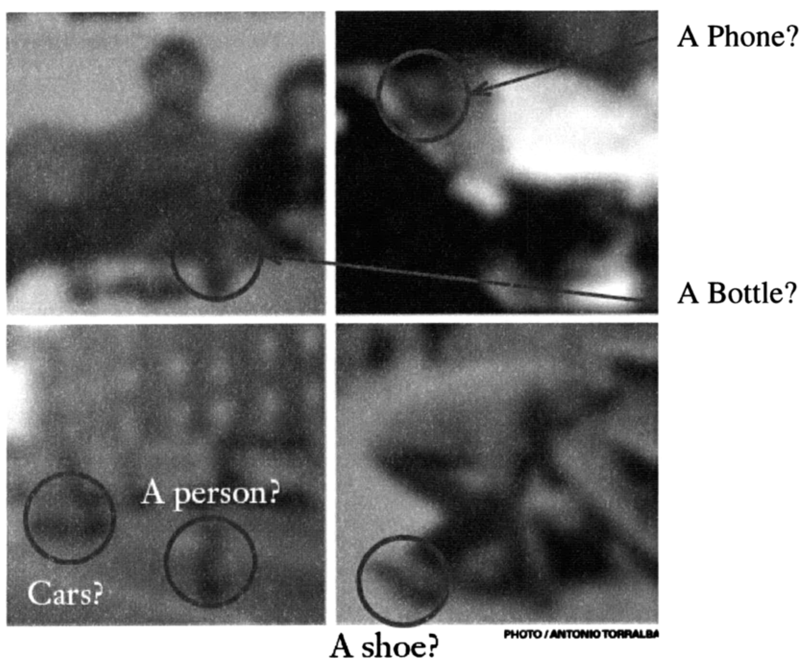

Figure 2-7: Context as differentiation factor analogy. (Torralba, 2008)

Although testing activities can be described with similar concepts using the same elements of function and form, context is the differentiator that creates differences and that drives management to treating each case separately, but at the same time is the driver for small organizations to be able to group them under the same umbrella.

Making an analogy to image recognition systems, a very same system can be identified and seen as many differences depending on the context that is being used as seen in Figure 2-7, were Antonio Torralba insert a identical images over low resolution images that he uses as context in order to demonstrate that context affects the recognition of objects. (Torralba, 2008)

By the context over which each activity performs, each one of them may be represented as a different type of control system.

Development testing is an iterative process at which the design is tested toward the system requirement from which its concept was defined. The dynamics of the development process is reinforcing until the system development delivers a prototype that is capable of meeting the system requirements under a certain level of tolerance. This characteristic makes the development process to be represented as a feedback control system, as shown in Figure 2-8.

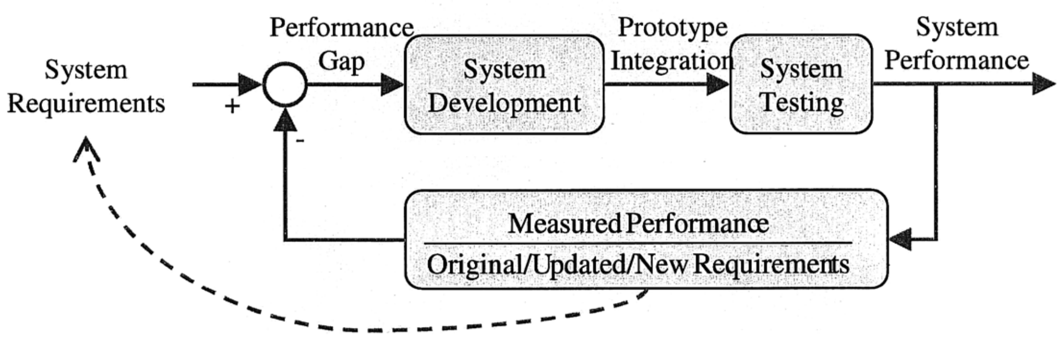

Figure 2-8: Iterative Closed Loop System Development

On the other hand a validation system makes in theory an assessment of the performance and reliability of the system with the purpose to verify that all requirements are met and be able to give sign-off and proceed with the next phase of the product design and development process.

This behavior is closely represented by an open loop control system as shown in Figure 2-9, although it is important to remark that there is feedback on the process when issues are detected or the system being tested fails, but this feedback is sent to the engineering community for correction, being correction, redesign, or development; not an activity of the validation process.

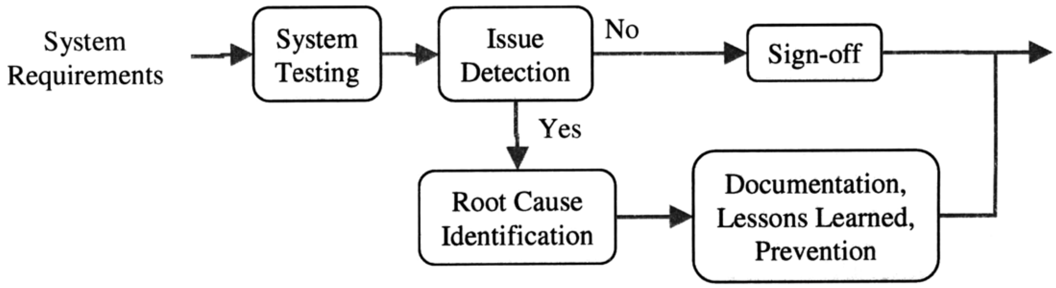

Figure 2-9: Open loop System Validation

2.3.1. – Testing Phases Overlapping within Emerging PDOs

A characteristic of product development organizations is the merging of roles and responsibilities when their size or degree design responsibility doesn't justify the specialization or division of area. This phenomenon is perfectly normal and completely makes sense in enterprise contexts where the existence of several independent testing organizations would not make an optimal use or resources, such as growing organizations branches of a well developed one.

The problem identified in growing organizations, especially in emerging markets due to the focus of the analysis made, is the overlapping of testing activities and therefore the merging of contexts within a same activity or planning. The nature of this behavior relies in the complexity of the projects handled by an organization under development in conjunction to the relative size of it in terms of headcount and expertise.

Development and validation are sequential activities which degree of separation is based upon the complexity of the project that is being held. Complex projects require intensive integration work in order to combine all the systems and more development work is required in order to deliver the targeted performance and refine the interfaces that new technology or applications require. After this development work is completed, functional prototypes are ready to start validation steps and confirm that the system as a whole is meeting its functional and quality requirements, presenting a full separation between testing phases at least on the affected systems. (Hotta, 2008)

On the other hand projects with low level of complexity such as derivatives of programs already in production require a less significant amount of integration and development work that enables some systems to be validated in parallel with the integration of the system. This case is supported by the assumption that the derivative being built would have a major effect on the base program attributes, and therefore doesn't require a special team for development and refinement. (Hotta, 2008)

Observations within a Mexican Product Development organization match with lessons learned from a Brazilian one as well with the recommendations made by the vehicle evaluation team from the North America Product Development at Ford Motor Company. Emerging product development organizations present undifferentiated testing activates due the complexity of their projects, and as complexity grows development work is not considered or correctly planned in terms or time and resources required while performing the original test plans. Some of the most common problems reflecting on a project are:

- Overrun or planned timing and resources due to unexpected development iterations.
- Validation testing repetition due to modifications on a system under developed that was already tested for quality and reliability.
- Late firefighting due undesired systems interactions that were planned for direct validation instead of early development.

As seen in well developed engineering originations such as in electronics, software of automotive; the development and validation phases are clearly defined mostly by the teams at the phase of design at which each one of them acts. But in the same way for small organizations, both activities can be performed by the same group/test team but there is the need to correctly define the context of each activity within a test plan or overall program plan and do not process to validation when systems affected are still under development.

The key for avoiding firefighting and unnecessary expenditure is correctly understanding context and clearly defining the boundaries of the systems and their testing needs (requirements and targets), this in order to correctly set the responsibilities of the testing group at each phase and be able to correctly set goals.

2.4. – CASE – Post Validation Brakes Development

Derivatives of existing platforms is the most common project that growing organizations handle, and in some cases the project handles enough complexity when the product derivative is not done only for the house market of the engineering entity but when it going to be sent also to other emerging markets.

This is a very special case since emerging markets present simple or more relaxed sets of regulatory requirements, and the customer requirements can be managed enough due to the socio-economical boundaries or contexts of those markets.

For this case, a new derivative from a top selling product is being made and close to its product development completion, some prototypes are sent to Mexico in order to be evaluated against local requirements and expectations. Since the South American office is far larger and more developed than the Mexican one, it is leader on the design and development of the program.

Almost at the end of the validation process, finishing the durability of the full vehicle system in South America; the prototypes sent to Mexico for evaluation present a brake noise which is totally unacceptable and will stop production of units. This noise is rarely presented in the South American market and yet when it is presented it not considered something unacceptable due it frequency of occurrence, but in Mexico City heavy traffic condition makes it appear frequently and in conjunction with the higher expectations from the Mexican customer created an issue difficult to solve.

This scenario didn't stopped the validation since at the beginning it was considered something very easy to solve and the development work was performed in parallel if not after the validation was conducted. Since this NVH issue was harder to solve than expected, great pressure leaded to heavy firefighting activities in order to be able to be able to proceed with production since the base program in South America was reaching completion.

The result was the creation of a generic brakes design verification plan, that has to be performed during the development stages of any vehicle that is planned to be sold in the Mexican market, in order to assure that product attributes and local needs are met before whole systems are validated and signed-off for production.

This case not only shows the importance of proper segmentation of development and validation phases, but also reveals the great need of a proper requirements and targets setting process which is a challenge face by any growing engineering organization when their work is not only intended for a local market.

2.5. – Summary and Conclusion

- Develop and Validation systems have architecture with common elements of function and form which may lead to confusion, since both require different approaches of usage and management.
- Form or decomposition view of form must never be taken as diagrams that explain how the system works, what does it delivers, or how it is managed.
- Elements within a same level may lack enough information in order to describe the interaction among the, which often can lead to wrong assumptions of independency among elements with high interaction for the correct delivery of system function.
- Although the form can be optimized for being the best for a system, it may be only the best for that system in particular and not for others. Specially among well developed systems, it form may not suit systems under development or growth ant therefore mirror images of a larger system may not apply successfully.
- Benefit and cost has a direct relationship to the function and form of the system respectively, being the value of the system the ratio of benefit over cost which relies on the importance of achieving greater functions with the less complicated form.
- Development and Validation testing present similar form and function, with a similar concept; which enables the activities to be performed by the same team for

small growing organization but as they grow the key differentiator need to be a well defined context and boundaries, which means clear roles and responsibilities among testing teams.

Chapter 3

=== D&V Background for Growing Organizations ===

The customer doesn't expect everything will go right all the time;
the big test is what you do when things go wrong.
– *Sir Colin Marshall*

3. – D&V Background for Growing Organizations

This chapter is oriented to describe the grow path that a testing and development group follows as a product development office evolves. Although the focus is around an specific organization of the automotive industry, insights and lessons learned from software and other industries are mentioned and applied in order to promote a wider scope.

By describing how Development and Validation teams are created and grow it is intended to show how context evolves and shapes the structure of testing organizations, and describe the path or pattern in which testing organization are shaped in order to be able to contain the needs of their product development organizations.

As described in the previous chapter, the context over which a D&V organization works define the difference among the responsibilities and the time over which those are developed across the development of a product or project. As well as the environment over which the D&V organizations operates, the boundary and supporting elements that are present contribute to the creation of value by the system.

The importance of a well designed context within each phase of growth for an organization resides in the enabled capability to correctly define the requirements and needs to fulfill by the D&V system.

3.1. – High Level Operating Strategies

Regardless of the engineering location size and expertise, or in the complexity of the program being developed, every project is bounded to effectively combine two of the three opposing strategies described in Figure 3-1 defined as the iron triangle by Oliver de Weck for general project management.

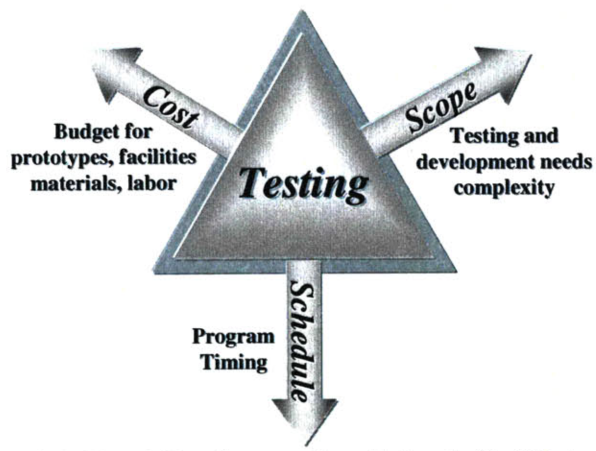

Figure 3-1: Test & Development Iron Triangle (De Weck, 2007)

On the test and development context, it is easy to understand how the iron triangle applies, just by defining the elements of function and form for the specific D&V system.

It is clear that a testing organization will search the most effective use of resource in order to minimize costs and be competitive and efficient, this strategy applies in the defined budget for facilities, materials, and personnel used to run all the operation required.

At the same time an objective of meeting all program deliverables on time, will the increasing pressure of reducing the time to market for all vehicle programs will move organizations for minimizing the time spent on the development and validation of a system.

End but not last, an engineering organization may look at least at the beginning to be able to have capability enough for developing the full attributes and functions of the system, regardless of the testing complexity of any given program.

3.1.1. – Most Likely Strategy Combinations for Testing Operations

It is evident that searching a wide scope at a low cost will result in increased lead times and long scheduling as the easiest alternative is to have equipment capable of developing many functions but not all at the same time having a low capacity and therefore long waiting lines among functions. Just as a computer with high processing capability but only able to do one operation at a time.

In the same way an organization trying to cover all the testing needs that could be used for any program with guarantee that capacity will be always enough to meet the ever compressing schedules that the industry needs, will result in a considerable investment with high operation costs, without even including the wasted resources for wasted capacity since not every program will need to test everything.

Parting from what was said by Wheelwright & Clark, in a rapid changing market, developing and validating products extraordinarily has become a competitive advantage. This competitive advantage can only be achieved by the use of lean testing structures and shortening time to market schedules.

This is the most common combination see within automotive, pharmaceutical, and software environments; but in order to make it successful there is a third element to be managed and is the clear definition of the scope that corresponds to the designed functions to be achieved by the system.

The scope must be set in alignment with the goal of the engineering organization from the PDO, being careful by being conservative enough for not being highly above it creating wasted capacity but at the same time not being conservative enough in order to be a boundary or restrictor for the full PDO growth.

As examples the scope within the testing departments from product development organizations must rely on the capability that the full PDO needs to achieve, which can be the development of a full vehicle, just the platform[1], just the top hat[2], just the powertrain; or a combination of commodities. The testing organization will have

[1] With focus on the automotive industry, we will be referring by platform to the group mainly composed by body and chassis structural components building the floor of the vehicle, considered the basic architecture over which derivatives are build and different powertrain systems can be mounted.

[2] By top hat we will be referring mainly as the group of components in a vehicle that forms the body sheet metal and exterior, as well as interior trim and the component and systems contained within them.

capability for performing testing on each one of the items mentioned above, but the degree of complexity will be limited by the availability of equipment and facilities, being defined by the specified scope.

For example the PDO will have capability for full vehicle testing but may no decide to focus investment on large and expensive facilities such as a electromagnetic testing chamber, or full vehicle environmental chambers requiring. It may have capability to develop a top hat, but the organization may decide not to focus on key life testing machinery for IP or seats relying on suppliers. This means that scope is closely related to an initial cost or investment decided within the company to set the capability testing scope.

In the specific case of an automotive engineering organization moving from an ongoing product development stage to more complex assignments, it is recommended to perform a vehicle overlay and focus the testing capability to be attained to the commodities with larger expertise and involvement (without recurring in unnecessary duplication with the mayor organization) with minor focus to the full program validation in general since it can be reinforced by the base engineering organization, suppliers, and external facilities.

3.1.2. – Common Issues in Growth Operating Strategy Implementation

It is not to blame for a testing organization to be able to address the three sides of the iron triangle, but unless an extraordinary management is implemented across an environment that facilitates the implementation (in other words acting at the right place in the right time when all conditions are given) projects may become the death march of the organization which ultimately finishes with the elements that conforms it.

Death March projects as defined by (Levenson, 2008) are a driver that generates collective irrationality within teams over time, and as shown in Figure 8-7 & Figure 8-8 it can damage the internal structures of team yielding them to un-functionality and separation.

Features that serve to characterize a death march project in order to stop it or restructure immediately include undefined scope, integration problems among project team members and system elements, constant re-estimation of the resources and schedule, design changes during validation testing, and lack of proper documentation (Levenson, 2008).

It is not possible to ask management not to press on the system to get results, but it is important to realize that systems can not work in emergency mode always due to the stress generated and the consequences it may derive.

Oliver De Weck describes that emergency mode are generated due wrong strategies for adopting the edges of the iron triangle, which detriment the success of the project in general. This strategies can be defined as:

- Poorly defined project objectives or shifting system requirements/specifications: Which lead to constant rework and higher resource expenditure in order to keep on schedule
- Mismatch between project objectives and appropriated funding: Which overstresses the existent resources in order to meet objective which will eventually fail in case of facilities and equipment, or leave in case of human resources.
- Schedule is too aggressive: Which overruns the project expenditure and detriments the functionality of the results given by the team since there would be always elements of the project left behind to correct later.

As shown in Figure 8-6 in conjunction with Figure 3-2, management may decide to operate their team under emergency mode most of the time or not, defining the type of project over which the team may be described.

Overstressed System Failed Expectations	Motivated System Accomplished Expectations
Project likely to succeed, unhappy workers	Project likely to succeed, happy workers
Project unlikely to succeed, unhappy workers.	Project unlikely to succeed, happy workers

Figure 3-2: Types of Problem Projects (Yourdan & Levenson)

As defined by Levenson and Yourdan, the projects on which the system is overstressed may be named Ugly or Suicide depending on the likelihood of success, but the common characteristic is that the team is not happy on the place they are leading to collective irrationality. On the other hand, motivated teams with accomplished work expectations remain motivated even if the project reaches success (Mission Impossible), or not (Kamikaze).

It is not implied that the motivated system can not be stressed for results, but the way stress is managed and the overall management of the team is what defines the point of view from inside of it.

3.2. – Background Review of a Growing Organization

Although literature covering the Product Development theme is extensive, the validation phases of it are not deeply covered and the elements that describe the context for the testing operations refer to a context of a well developed engineering organization rather of an emerging one.

The context of operation may vary as the organization grows both in size and responsibility, and in order to correctly exemplify the context evolution for a testing organization we will review the case of one of the engineering branches from Ford Motor Company. The engineering branch in Mexico is good example since it involves all the aspects seen in a testing organization that suddenly starts again from zero.

3.2.1. – Ford 2000, Bringing the House Down to One

Ford Motor Company after an extraordinary success of consolidating the development of powertrains in 1993, decided to expand the initiative to the development of full vehicles centralizing the engineering center in the quest of a global system that used less resources and could be able to develop vehicles faster. This appeared to be a great idea at Dearborn but for the Ford of Mexico product development office it was a calling to almost disappear since the regional engineering structure was shifting to a central one in quest of efficiency by having one global model that fits all markets including the so desired emerging ones.

The company restructuration had many aspects regarding engineering, manufacturing, marketing and sales; while led the division of the entire vehicles to be developed between Dearborn and Europe, entities with the capability to tailor models to the specific needs of every market (Mexico, China, Russia, Brazil for example) in accordance to local regulation and needs but under the same powertrain and platform.

Engineering operations at Ford of Mexico reached at point of near extinction, keeping just the necessary for ongoing product development for the vehicle being locally assembled as described in chapter one as LPC – Local Product Control.

The test and development center saw its operations diminished and its central laboratories torn apart leaving in disuse the little remaining equipment, which overtime became discarded. Testing work was reduced to local homologation and Build-Up validations.

At the time it was say by Ed Hoegenlocker, an executive in charge of automotive operations that this centralization would take the broadest possible view of market opportunities being able to develop products to serve multiple markets. Unluckily the second attempt from Ford to develop global vehicles (this time from a centralized perspective) was wrong especially in the regional emerging markets where the lack of a consistent definition of local market requirements led to vehicle failures that changed the way the customer saw the Ford brand for many years to come.

Now the regional offices have been restarted, each one covering again the path of growth and challenges, with the responsibility of being the hope of a new company restructure over the recent automotive crisis, on the third attempt of a company to develop a global car but from a global team perspective.

3.2.2. – Fresh Start for a Generic Testing Entity

With responsibility of only homologating incoming vehicles and verifying against the most basic needs from the Mexican market, a single General Testing Operations entity fully leaded all the testing and development activities performed for local programs, receiving information and test request from the existing engineering teams, and performing general vehicle testing in attendance to well know areas that could present issue.

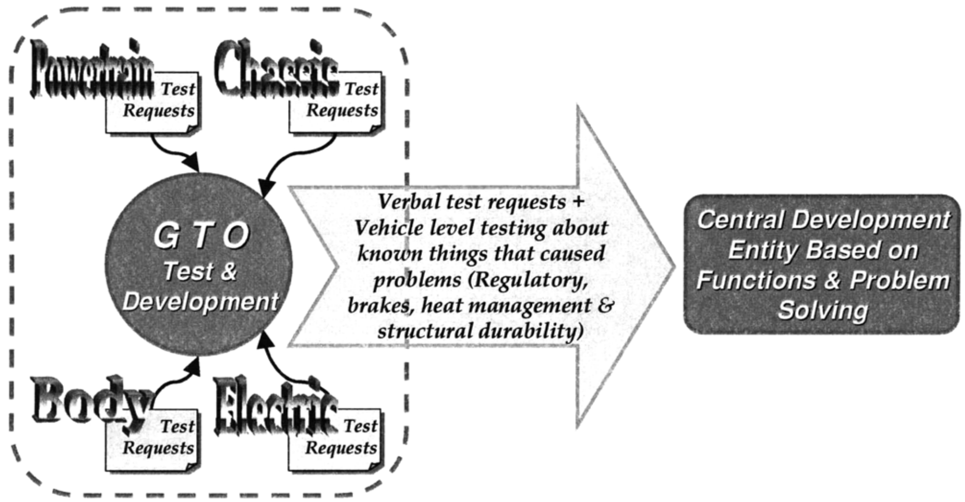

Figure 3-3: Build-Up and LPC Generic Testing Structure

Figure 3-3 shows the context over which the test operations performs in the early stages of growth of a PDO marked in the dashed rectangle, where a central testing team works directly with each of the engineering groups in order to validate the work regarding running changes to local products as well as problem solving alternatives for quality issues.

3.2.3. – Ramping Up the Needs for an Organized Development and Validation

As responsibility grows we know see to separate context, one that contains the engineering organization by functional area and another that contains the integration, development and validation activities. These concepts have a certain degree of overlapping due the need of component testing or special requirements for functional performance; as well as for the heritage of a previous phase. This stage is demanded by the complexity reached by the development of derivatives from other existing programs as well as the continuous development of local products

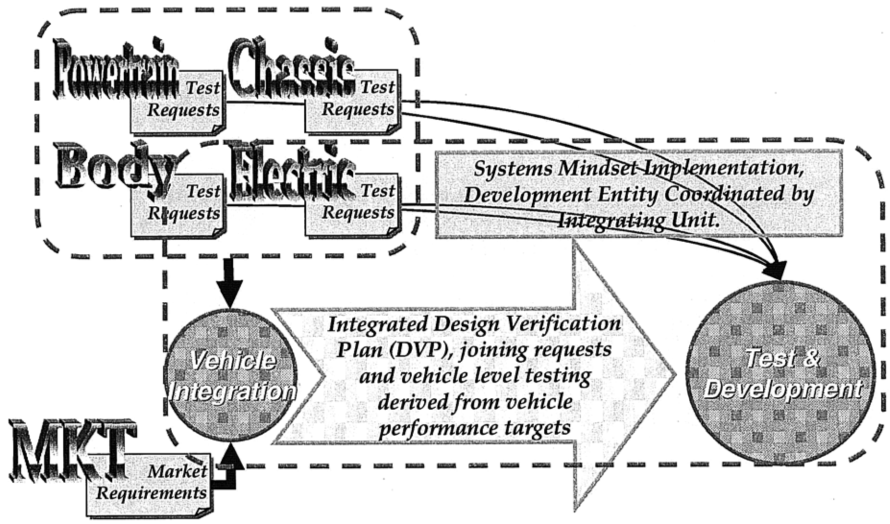

Figure 3-4: LOPD Generic Testing Architecture

Within the context a vehicle integration entity created to integrate the needs from the engineering teams and combine them with the attribute performance targets derived from the local market needs and benchmarking. Establishing communication links and organizing the process. Test & Development continues to receive additional engineering test request for special testing, and functioning as a development problem solving entity.

A wrong concept within product development, which is widely seen in a growing organization, is the confusion that testing is no more than "Verifying" that we are good or ok in a component or system. This error places testing as the individual in the soft-drink industry, who is watching the empty bottles pass through a light, verifying that the bottles are clean and empty in order to be filled.

This error leads testing to be seen as process that instead of adding value, just creates waste, and waste has to be eliminated. This behavior drives testing to be performed late, when the engineering team has a prototype ready just expecting to be stamped with a big OK, in order to continue with the next milestone. As hard as it sounds, it is true, and it is wrong since problems are revealed in late stages of the process and we loose all the learning opportunities by firefighting.

3.2.4. – Increased Complexity for a Complex Emerging Organization

Testing organizations have a huge challenge while trying to catch up and evolve with the same pace as their engineering office. It is mucho more difficult for testing to keep pace since their engineer not only need gaining expertise in attribute development, methods creation, and strategic planning; as the engineering community needs, but they also require a growth in facilities and equipment which requires investments that in many cases are not possible to afford.

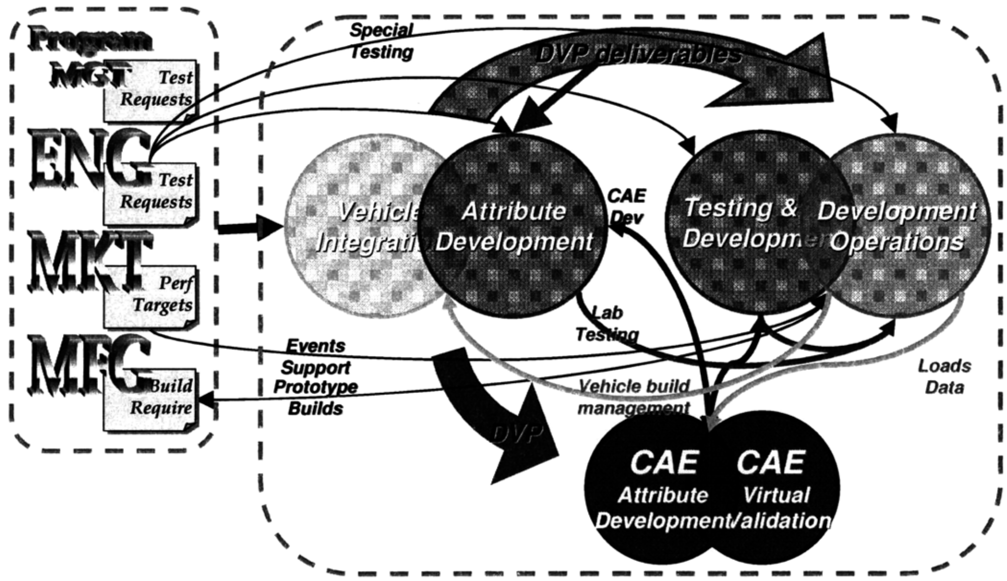

Figure 3-5: D&V Architecture for a Young Restructuring PD Community

Engineering capabilities growth require only knowledge and expertise, but the growth in attributes development and systems validation require in addition a big investment in facilities being the constrain for many organization to reach the next step (Arbitter, 2008).

Figure 3-5 shows a Test & Development structure is much more complex. The integration entity starts a growth path on two sections, the vehicle integration and a starts new vehicle attribute development branch. Testing operations could be identified in two sections, the one in charge of development and sign–off testing, and another in charge of the testing operations (instrumentation, facilities, and vehicle services). CAE is a starting tool with involvement in development and testing operations, in search of better, faster and cheaper development.

At this point, the integrated Development and Validation Team presents duplication in roles and responsibilities due communication and knowledge capturing issues. Existent expertise is not captured but replaced, and restructure is needed in order to maximize the use of existing resources now oriented into the new needs of the PDO.

3.3. – Development Constrains for a Growing Organization

As described before the needs that a product development organization has to fulfill in order to have a significant growth in terms of system development and validation may become an obstacle for assuming responsibility of larger and more complex problems.

So far and organization that has successfully managed local ongoing product development for local products has the advantage of the flexibility of regulation and sign-off systems of the region, since the product is not going to be released on more stringent markets that demand complex facilities. This context applies din the same way for validating existing designs from the global portfolio for it introduction to the local market, as well as for variants from the global portfolio to the local market

For an emerging PDO the challenges for achieving increased engineering responsibility for future programs are:

3.3.1. – Facilities

Resources for testing and validation are an investment that every organization must plan how to manage. On lower complexity project organizations have been successful in being able to replace the need for large and costly facilities that replicate the desired conditions for verifying a requirement but that lack some control attributes that can be waived due the nature of the project.

Current level of resources for development and validation activities has driven for example to replace climatic wind tunnels for vehicles running on chassis dynamometer or even in public road on hot weather ambient locations.

Although this has enabled organization to validate requirements alternatively, it still lack in the control, safety, and confidence of a closed environment. In the same way the level of complexity of the programs managed avoid the need of extreme secrecy or confidentiality over the prototypes for the derivative scoped.

In order to solve facilities issue, organizations should follow an organized strategy to achieve the capability. Briefly mentioned the steps for increasing the facilities and equipment toward testing capability enhancement are:

1. Define a clear scope and goals for the programs to develop
2. Identify the functional areas and attributes affected
3. Identify items (equipment/facilities) to fill the capability gap
4. Identify alternative test locations or prepare business case assessment
5. Prioritize based on most urgent needs

3.3.2. – Degree on Development Changes against Manufacturability

This is a critical item when the derivatives being developed by a PDO is being assemble at another region where the local PDO is owner of the design as well as for the manufacturing process.

Design for manufacturability meets design for their manufacturability when the home region PDO is not able or willing to modify the current manufacturing structure in order to accommodate the changes demanded by a derivative and instead it is asked for the designers to adjust to the current way the vehicle is done.

This constrain is a requirement that in order to be avoided or minimized has to be clearly negotiated at the beginning of the program phases in order to set it as a requirement that avoids later firefighting.

3.3.3. – Design Authority

Linked to the point mentioned before, there is a degree up to where a program derivative or a special version for a region can be manipulated when the PDO making the derivative is not the same that owns the design.

A correct definition of scope and requirements to test having in the loop the derivative designer as well as the design owner creates an easier path for the avoidance of rework and firefighting for new alternatives in later steps of development.

3.4. – CASE – Electromagnetic Testing for All!

Setting a correct scope for the testing capabilities that will fulfill the needs of a product development entity is a hard task since it requires a forecast of larger view than the one done by the PDO in the way that investment in new facilities or search for the media for verifying requirements.

Creativity is the element that will create the conditions for covering the testing needs of a PDO by rather investing, developing alternatives (such as CAE, suppliers, external labs), or correlating existing methods in order for them to be applied effectively.

On the other hand unused testing capacity is the worst waste that can be found in any PDO, and should be avoided at all costs. Alternatives for selling testing services or collocation of equipment among regions are the recommended paths for this issue, since equipment and facilities sitting idle is a waste that few companies can afford.

In term of electrical and electronic systems development, Dearborn Development Center has a small EMC[1] chamber that is nearly enough for the needs from the electrical and electronics group, but lacks both the capability and capacity to be able to fully cover all the development and validation needs required for the programs managed.

In order to cover the needs from Ford Product Development office the test operations ask for the service of the GM facilities in order to cover the capability and capacity gap. On the other side GM has five times the capacity available at Ford with higher level of capability, which is not able to be fully used by its owners even by selling service to other OEMs.

The example shows clearly the importance to set correctly the scope for the testing capability and capacity, since a company must only have what they need or expect they will need avoiding having such an excess.

On the other hand the error from a firm is the opportunity for others, since it is not required for other companies to build such capacity since GM's is ready available at least for now. In the case of growing emerging organizations this is a vital point since these

[1] EMC stands for Electromagnetic Compatibility, a test required by any vehicle development that involves electrical systems in order to ensure that signals on the environment doesn't interfere with the correct functioning of the vehicle and in the same way ensure that the vehicle doesn't emit any signal harmful to other systems in the environment.

organization handle a very limited number of programs at the same time due its size constrains, and never would obtain a favorable business case for such large and expensive facility with the projected usage their needs demand.

The remaining question is if an organization that has equipment no longer being used should get rid of it?

3.5. – Summary and Conclusion

- Test and Development activities must be managed with leverage among cost (budget for prototypes, facilities, material, and labor), scope (testing and development needs complexity), and schedule (program timing); where two of those three strategies have to be chosen as main drivers.
- The recommended approach is to follow an strategy driven by being effective and competitive both in cost and schedule, and let the scope be set in alignment to the goals of the PDO focusing on the program characteristics to be designed or specific commodities within it.
- Although teams could give great result working under emergency mode, or high pressure against the fulfillment of scope, schedule, or cost targets; it is necessary not to stress all the time the system in order to avoid the generation of death march projects which terminate the motivation of teams over time and generates collective irrationality.
- As testing organizations grow the context over which they work changes little by little, in other words the elements that are present and have interaction with the system for it value delivery shift and evolve as well. Therefore the structure of the system must be rearranged in order to avoid duplication of tasks and take advantage of existing resources and expertise instead of creating new.
- Testing and Development activities may be constrained with the following factors for emerging organizations:
  - Extent at which redesign can be made on designs not owned by the PDO
  - Degree of development changes that can be achieved on vehicles not produced locally

- Current level of resources for development and validation activities

- In terms of facilities and equipment which is the most common obstacle for growth in test and development organizations it is recommended to have follow a structured method now briefly mentioned as:
    1. Define a clear scope and goals for the programs to develop
    2. Identify the functional areas and attributes affected
    3. Identify items (equipment/facilities) to fill the capability gap
    4. Identify alternative test locations or prepare business case assessment
    5. Prioritize based on most urgent needs

Chapter 4
=== System Requirements Analysis ===

If you don't know where you're going, you're unlikely to end up there.
– Forrest Gump

Perfection of means, and confusion of goals seem, in my opinion, to characterize our age.
– Albert Einstein

4. – System Requirements Analysis

The define process that is performed before the validation phase starts is decisive for the speed, effectiveness, and efficiency of the testing phases done within the development of a new product. (Harvard Business Review, 1995)

Many thoughts and discussions have been carried over in order to determine what is needed to do in order to significantly improve the development and validation phase within the product development process. The result agreed is that for every great problem the solution must attack the root cause, so searching the beginning of ever testing activity we find that the definition of requirements is the key element to have an effective and efficient development and validation phase. Poor planning at the outset is cause of testing delays.

The definition of system requirements is an ability that must be developed by any growing organization in order to avoid falling into a highly component focused specialization, being able to achieve systems integration with clear requirements and specifications for an efficient DVP.

The point of view for this analysis is from the experience gathered as a NVH vehicle engineer at Ford Motor Company in Dearborn during the creation of a new group as well as the targets and requirements for the system assigned; and in the same way insights from the vehicle evaluation and verification sign-off groups were applied..

Experience and insight were shaped through the systems engineering perspective and the tools learn, in conjunction with the experience achieved during the years performing as a test and development engineer in an evolving Ford of Mexico PD office.

The objective is to build a path with clear system integration structure to define and select system requirements and specifications, combining lessons learnt and the root causes of issues that generated those requirements; for achieving lean development and validation phases doing the right tests in the optimal platform and method (i.e. prototype – CAE model)

This chapter proposes an framework for the derivation of requirement for the elaboration of efficient and accurate design verification plans. The purpose is to achieve a well designed system for processing the system requirements and specifications in which a vehicle will be aligned. The system desired to create a driver for a developing organization with a methodology for the definition of the constrains and boundaries to design the vehicle based on the experience and insights from the industry (best practices) as well with future tendencies in the industry and academic perspectives.

4.1. – Definition and Classification of Requirements

As defined by Grady, a requirement is an essential attribute or characteristic of a system that is expressed by numbers or engineering terms. Requirements must be expressed by either quantitative forms (the noise level must not exceed 64 dB at the driver position) or qualitative forms (the bracket should not brake or present any crack after full life testing), and be expressed clearly without any points of misinterpretation or ambiguity. Valuation of requirements can be done thru one or more of the following approaches as long as they are consistent with the physical properties or ways the product is perceived by the customer:

- Material or components capability (structural, thermal, etc)
- Expertise and lessons learned on the system or component
- Customer need/appreciation valuation

Within the product development process, the system requirements analysis is a task that requires attention from the moment a concept is created. During the definition phase of a program, the team establishes the objectives of the program, a list of attributes that the product will do.

When the attributes that the product will have are defined, the team must determine the specifications and requirements that become the boundaries in which system design will be made, boundaries that will define the methodology to follow that the design will meet them.

Requirements can be presented in several forms depending on what do they asses as well as the importance they have for ensuring the success of the product and its continuity within the product development process. The main types of requirements could be described as follows:

4.1.1. – Attribute Requirements

This kind of requirements refer to the performance of the system as perceived by the customer, known as attribute, which act at a vehicle level in order to be in line with the customer usage mode of the system and perception transfer paths.

- Noise Vibration and Harshness
- Vehicle Dynamics
- Performance
- Thermal Aero
- Fuel Economy
- Packaging
- Ergonomics
- Safety
- Craftsmanship

This requirement define the way the customer will perceive the vehicle and are the most direct path toward the customer need in the creation of a product and therefore the importance of clear definition and commitment to fulfill them.

This requirements have the particularity that are shift from vehicle or product line to other depending on the market and the market segment they are designed for, representing that the derivatives of a same vehicle may have different attributes requirements depending on the market segment it is planned for.

The definition of this requirement is difficult at takes a lot of effort as we will describe the process later, but nevertheless these don't warranty success on the vehicle since customer perception vary across cultural, social, and geographic contexts. As an example, although the climate control system for a Mercury Milan and a Ford Fusion is exactly the same, the Milan gets higher scores in climate control NVH perception while the Ford system is highly criticized.

4.1.2. – Government Regulations

This is a very special type of requirement that refers to local law and regulation from each country. Safety, emissions, and braking are some kinds of governmental requirements which need to be clearly stated at the moment a vehicle is conceived in order to assure that it can be sold at the desired markets. In these cases engineering requirements that are constrained to meet the law and regulations that controls the form and or function of components and systems.

Although at some instances a region may accept validation certificates from another country when the testing capability is not ready available for the vehicles sold in the market, it is often demanded that testing to assure that the vehicles comply with local regulation requirements. In many cases regulation is not strict enough to produce important changes within the architecture of a system.

Homologation is the activity that ensures the correct gathering of regulation requirements that apply for a product in accordance to the markets where it is going to be sold, ensuring that the appropriate testing is done successfully ensuring the compliance with each entity required.

Regulation affects the architecture since it may override the boundaries defined by the strategy regarding the customer needs and company goals. Regulation sets definitive boundaries in which the product architecture has to be conceived, it has the benefit of the users at very high level but rarely directly addresses the need of the customer at low, short, or conscious term levels.

4.1.3. – Design Specifications

Design specifications is a requirement type that act as a boundary to the design of systems and components in terms of what is the minimum performance expected (e.g. brake pads to last at least 30,000 km) or point that the system must not pass (e.g. coolant temperature not to exceed 110ºC). The specifications and requirements that will apply for a product program need to be clearly defined and documented in the very early stages of design.

These requirements are elaborated for company usage based on expertise and knowledge of the system, as well as lessons learned from previous programs and customer needs. Although they are mandatory for the full production of a product but can be waived in very exceptional occasions, they dictate the rules and in many cases act like gatekeeper for the sign-off of the program. In cases where the fulfillment of a requirement is not critical for the performance or reliability of the product on the field, deviations could be signed by senior management in cases where the context where the vehicle will perform or the system capabilities doesn't require the protection of a strict requirement.

The main purpose of specifications is to ensure a certain level of quality within the product as well as to ensure a continuous improvement of the technology used.

4.1.4. – Engineering Standards

Standards are much related to the word branding, since they are not rules but internal agreements within a PDO for how does a system, subsystem, or full vehicle should perform or look like. It other words it defines a vehicle in it minimum terms that are needed in order to consider it within a brand or family of products which could be related both to performance and content.

Standards define how the components should be designed and how do the interfaces should be in order to have a reliable product that meets the brand standards.

Besides that proper function within brand characteristics, standards are based to customer satisfaction ratings in order to maintain a certain degree or value delivery to the customer.

Standards enable the vehicles and systems to be designed by different and distant (thru time and space) set of engineer using established patterns. They apply both to components but as well to the interfaces which make them very valuable in systems engineering and integration labor. (Whitney, 2004) Following standards simplify the design by making the engineers use established concepts which in may cases helps emerging organizations to develop their capability and attack complex projects when their engineering critical mass still doesn't have the full required expertise.

4.1.5. – Engineering Guidelines

These are not requirements in proper words, but knowledge and best practices gathered thru time and programs with lessons learned and recommendations about what is desirable for design characteristics and what to avoid on basis on problems generated in the past.

4.2. – Timely Requirements Definition Importance

As mentioned before, proper definition of requirements to be validated or pursued thru development in a program is essential for achieving a lean and effective validation phase; but this activity is clearly dependent from the define phase of the product development process.

Not achieving a timely definition of requirements generates waste of resources by adding firefighting and late changes to the product; and on the other hand the full performance and quality of the product is inhibited when design solutions are already established just to be tuned or downgraded until a certain requirement is met.

Three main drivers affect the timely definition of requirements for projects developed by growing organizations which could be described as follows:

4.2.1. – Solution Based Verification Plans

Solution based Testing: Especially during development work, it is common to find that management/engineers rush into solutions without fully understanding the requirements and constrains that affect the system in order to provide a nimble development of systems and problems resolution. After the solution is being worked and tested, when the results start to come out is the moment where requirements and standards are looked for in order to assess if the solution really solves the problem.

This process often lead to rework due when the applicable requirements are followed and intensive work is done to the solution in order to make it fit within them; being the event in some cases where the best requirement that applies for their solution is selected.

Requirements should be independent from the design and not predetermined by a specific design solution point (Grady, 2006), on the other hand design solution must be thought taking them as a basis.

4.2.2. – Test Based VS Requirement Based DVP Definition

As observed within the PDO under analysis, historically the design verification plans have been elaborated with a test based mindset, which results in validation plans where requirements are assumed to exist but not analyzed and neither reviewed. This generates rework and raises the probability of field failures since the products are not being tested on what they should be, and waste is produced since tests that don't need to be applied is done.

Preparation of test plans required a systems thinking based on requirements, parting from the very functions and goals from the program in order to define what is needed to be tested and how.

A test plan based on a test based system may indeed test all the systems and components that require to be tested but it doesn't assure that the proper testing is being done and that the right failure modes are being analyzed.

In previous steps of growth of an engineering organization the test based scheme has proven to work since testing work is not intended to develop in depth the systems and components but rather just validate them or homologate them, and often the changes involved are soft enough that issued can be detected by the most general testing rather than focusing on specific requirements for special functions.

On test based system it is possible to fall on over engineering, or testing without purpose. For example in the personalization of a vehicle for a local derivative, a change in the tire wheel assembly required testing, and a method was identified. For this method a specification to meet was set, although not even the request had or knew that specification, leaving a verification plan without a value to know if the test was successful or not or if the test assigned was the right one for the change to validate.

4.2.3. – Improper Definition of Program Functions

Lack of definition of program functions: Requirements are derived from the functions, an attribute of the system that has to be clearly defined in order to design its form. Therefore it is impossible for any engineering organization to deliver an architecture for a product or system when the functions of it are not clearly defined at the definition phase but rather when the first functional prototypes arise.

Over the years I have seen a lot how there is no clear definition of function, and when the first prototypes are ready to be evaluated it is assumed that the system will give successful results when at first time there were no results to pursue in the definition and design phase.

The most common result of this type of practice, seen at least on 80% of the project managed by a your emerging PD organization is the gap existing between the system level requirements which are undefined, and the component level requirements which are done blindly by the design engineer that need to get started.

4.3. – Development of Targets

Target setting is an essential part of development testing since it controls the interfaces of systems and components, tuning the behavior of systems as one entity which results affects the perception of a the customer as a whole for a certain attribute (e.g. NVH, vehicle dynamics, fuel economy, etc.)

The importance of a proper target and requirements generation system in critical within any product development organization, and the existence of this system is in fact what defines the existence of product development as an engineering activity.

If the development, design, and engineering teams don't work together in the development of targets, specifications, and metrics; the whole development department becomes just a test measuring center. Engineering, development, and design teams need knowledge and experience sharing, linking it to full vehicle requirements cascading them up to component level in early phases of the PDP.

In order to develop the proper target for setting up the development work for a vehicle or any product in general, appeal studies must be taken into account in order to have a direct feed of the current customer perception against surrogate and competing products. Appeal targets must be defined based on the specific regions and markets where the product is intended to be sold and in the same way be based on segments. Regional (Mexico, Brazil, Russia, India) and segment (compact, midsize, large, truck, premium) division is a driver for avoiding overdeveloping and reduce test expenses since the items to verify are directed to contexts where customer demands could focus on attributes that others regions don't, being the same case in segments.

As an example, when developing targets for climate control systems in specific the quietness of the Heather/AC fan, it is important to analyze the region and markets segment to understand the conditions over which it is used, in order to understand the usage (extreme min ←→ max speeds of operation) and the vehicle segment in order to identify feasibility constrains (subcompact vehicles may not have space for noise suppressing AC ducts)

In emerging countries it may be not easy to find a complete appeal study or even finding one at all, being a disadvantage in the target setting process between developed countries where customer feedback could be pursued even up to system level (A/C NVH performance for example). In these cases it is recommended to base the analysis on every bit of information regarding the customer satisfaction and feedback, but other systems such as warranty or customer claims (not always possible to correlate to appeal) may be combined with benchmarking testing.

Other sources for the appeal information come from the direct relationship with internal marketing entities in order to import and translate data from market research, surveys, and focus groups; part of the PDP process where the customer needs have to be defined into engineering terms. The definitions of these methods for acquiring information are essential to make sure that the engineering metrics correlate with the customer needs.

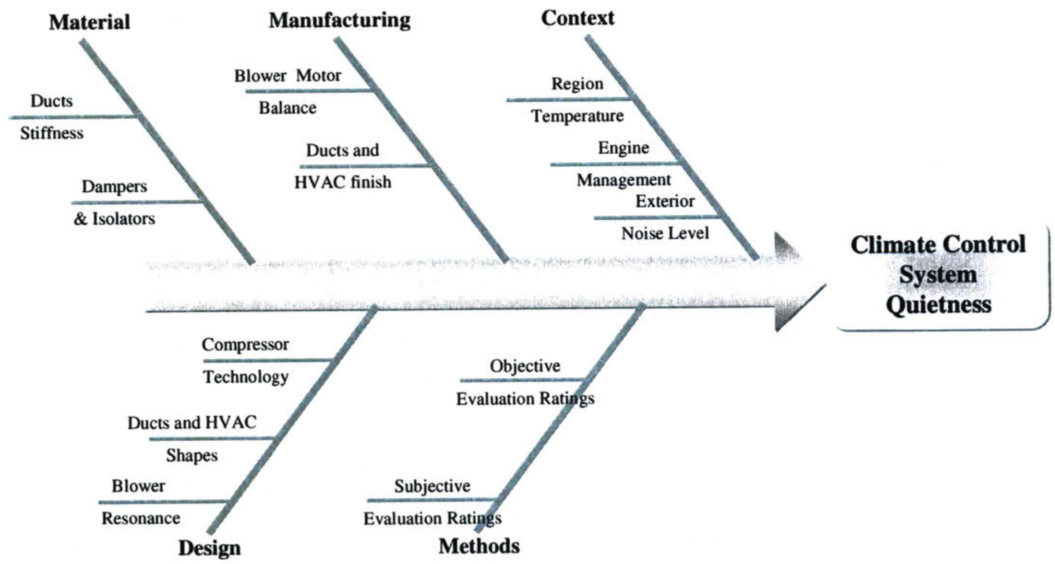

Figure 4-1: Ishikawa Diagram Example for Climate Control NVH System Targets

Making a form decomposition of the system responsible for the attribute being evaluated, system engineering tools and processes such as the use of Ishikawa of fish bone diagrams with QFD support serve to identify all the "Xs" that contribute to the delivery of the attribute receiving an appeal score.

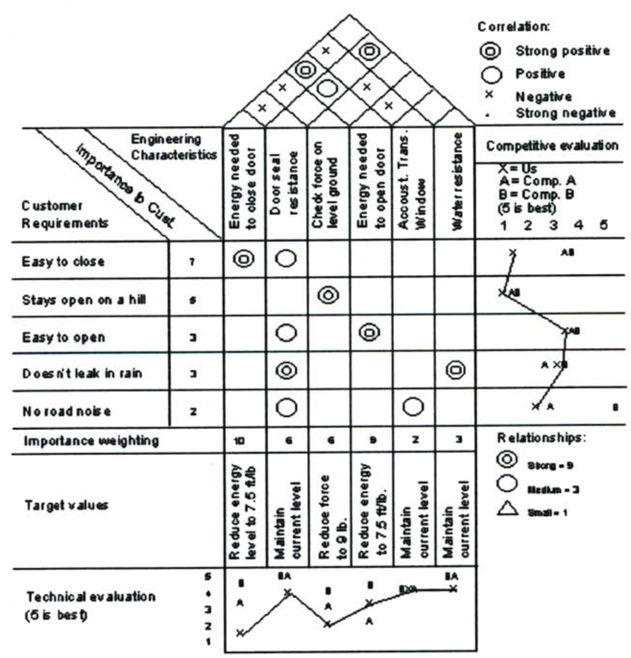

Figure 4-2: QFD Diagram Example

J. Hauser and D. Clausing, (1988) The House of Quality, Harvard Business Review, 66(3), 63-73.

A useful tool for the decomposition of appeal based scores into engineering delivery contributors X is the Quality Function Deployment matrix. The tool helps the mapping process from qualitative values given by the customer into a qualitative engineering system that can be measured and set specific targets on them. An advantage point in including the information from benchmark studies relies on the direct mapping or the local PDO delivery capabilities vs the ones offered by the competition.

Over the vehicle or system attribute level appeal score determined thru warranty or customer satisfactions indexes, the team must identify all the Xs and be able to measure them objectively, this attention requires a lot of expertise and joint efforts within the engineering team, since missing some Xs will drive the targets metrics to not really enable the development of a vehicle or system that is able to satisfy or exceed the expectations from its users. Targets should objective and be a mathematical interpolation among the leaders in the market, that gives enough information to management to correctly decide the direction that a project will take in term of the resources that need to be invested in the development of system and the ability to get high appeal scores by them.

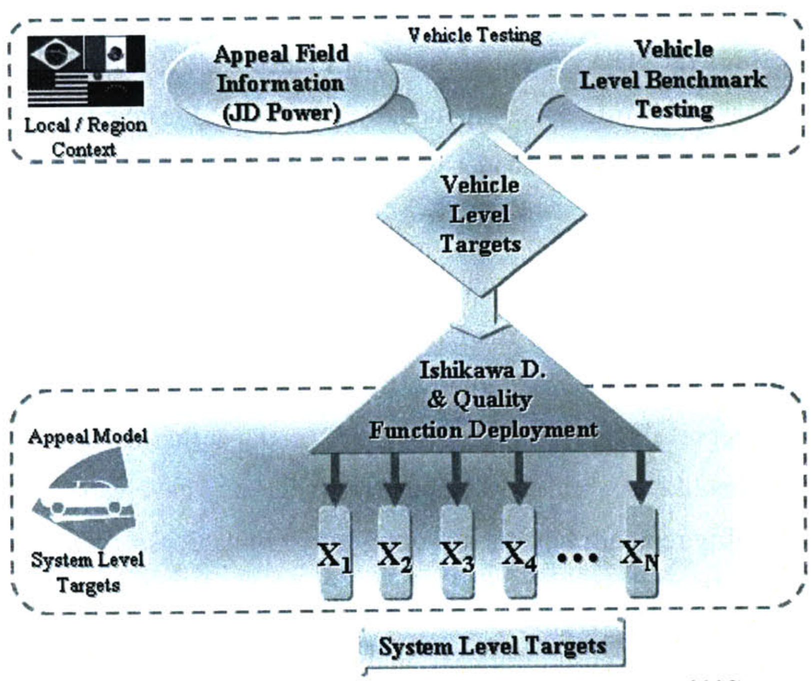

Figure 4-3: Target Development Framework (Halliday, 2008)

As shown in Figure 4-3 the start from any target setting process resides on regional appeal studies in combination with vehicle level bench mark testing in order to identify the engineering or performance value of items valued subjectively by customers.

Summarizing the idea on the framework presented in Figure 4-3 the development of targets start with the basic needs of understanding the vehicle needs which are already taken from appeal and customer satisfaction models, defined over the regional contexts where the vehicle or system will be sold. Then thru the use of system engineering, the elements in the system must be decomposed in order to determine smaller solutions or elements that deliver solution to the needs.

4.3.1. – Level Bar Setting Complexity for Targets and Requirements

It is defined that targets and performance requirements must come from appeal models and cascaded from vehicle to component levels. But organizations face the complex problem of how to channel the engineering efforts to a variable that is not always possible to correlate with appeal.

One of the elements that are difficult to correlate are quality indicators such as TGW (Things Go Wrong), and this is a distress factor between the development and design communities within any PDO. This conflict arises since the vehicle level team searches to deliver appeal targets, while the component/system focused teams looks forward to the reduction of TGW, being elements under which each team is evaluated during the annual performance reviews.

Both strategies are important since companies can no longer survive with the archaic strategies where the quality was though to be given and reliability numbers were drowning. But at the same time modern times demand appealing models, and companies need to avoid strategies of delivering good quality products that doesn't look appealing. For some companies like Toyota the previous statement may have worked so far, but for other which reliability reputation is not so strong the formula doesn't work the same.

The recommended action is to adjust performance review indicators to the improvement of quality but not at the expense of appeal, making the delivery of appeal be defined by contribution analysis to systems and components.

Although raising the bar for target and requirements setting is a methodology for ensuring the continuous growth on the products of an organization in terms of quality and appeal, an obstacle that dampens this strategy is the associated development cost required for being able to meet the targets.

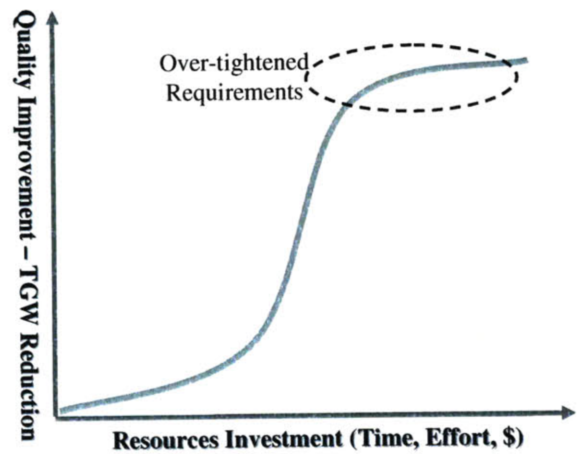

Figure 4-4: Quality Performance vs Cost Model

For example setting targets of zero TGWs push up the associated costs within the S curve described in Figure 4-4 pushes the resources required to meet the target to a region in the curve where a lot of resources need to be invested and the reflected improvement is not significantly improved and no perception may be sensed by the average customer.

Management needs a clear involvement in the process of target setting in order to involve the necessary business instances and optimize the level up to where the targets will be fixed. In many cases target levels get lowered in order to avoid costly development or final cost per unit.

But the level up to where the target will be fixed for every detected "X" in appeal models need to be the necessary to deliver the necessary for integrating a winner model. While establishing a target it is important also to verify if programs under development and future vehicles to come will be able to fulfill the targets, since it is not advisable to deliver reports to upper management showing that no program in development will meet our targets, which are based on the customer expectations. (Arbitter, 2009) This complexity is derived from two possible root causes:

- Vehicle level targets are far to strict, although they were supposed to come from the customer and benchmark driven appeal models.
- Unconscious lack of compromise by the development and engineering team to work out a systemic path for attribute performance that deliver the required level established by the targets. This could be derived by the expenditure limit of the program, but at the same time it is supposed that the appeal model and benchmark are based on the region and segment of the market to be covered.

4.3.2. – Targets and Requirements Units and Consistence

Targets used for the development of systems require methods that can be clearly correlated in order to be able to assess systems and components in a consistent way, with the purpose of collaborating within engineering organizations and suppliers with the same language and level of understanding.

Targets for systems development and the methods to asses them must represent a real customer experience, and therefore must be represented on the units that most closely resemble the real user perception.

As an example, considerer the NVH developments done in every vehicle sold in the market. The NVH attribute of these vehicles need to be measured in a consistent unit that makes possible the comparison among vehicles of the same company and other from benchmarking that at the same time makes an accurate correlation to customer perception.

For a certain measurable within the NVH groups the Ford Motor Company target specified that the truck should meet as target a maximum noise level of 66 dB while the GM competitor has measured 71 dB on benchmark testing. This target initially could have shown that the Ford truck would get a better appeal grade than its competition, but in reality it was the opposite. The frequency that was making the equivalent system at each truck brand was different producing different values of loudness, where the GM product was performing better than the Ford branded.

Even for mature companies, the need to upgrade its requirements system is critical in order to increase the level of representativeness of them. Figure 4-5 shows the difference that can be created using different systems to measure the same phenomena, where the unit that offers a closer representation of the customer perception must be chosen.

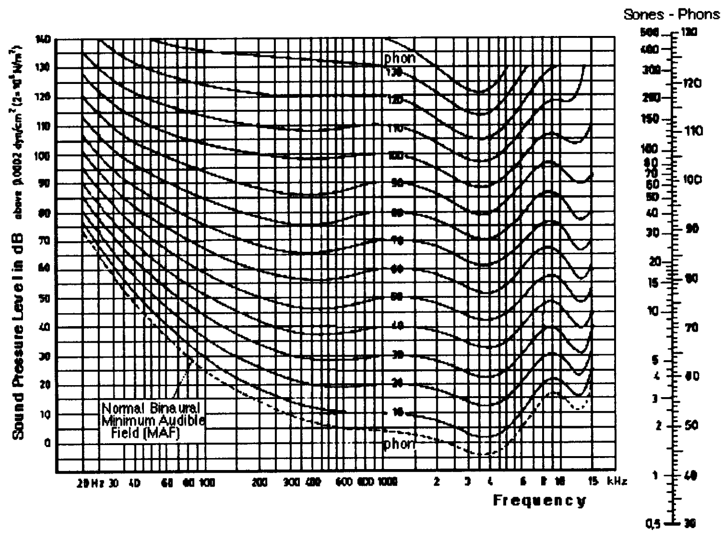

Figure 4-5: Perceived Noise Levels vs Pressure Relationship
(Phon – http://www.sfu.ca/sonic-studio/handbook/Phon.html)

In this case, a renewal of the noise requirement system is required in order to develop vehicle under targets with higher representativeness, although it will be needed to negotiate and set up an agreement among organizations and suppliers in order to have a common language for vehicle, system, and component level requirements. At least in this case the usage of sound pressure levels (dB) need to be replace by loudness levels (sones & phones)

The benefit of using loudness instead of the traditional sound pressure level resides in the existence of a linear interpolation between something loud and something quiet, since the current usage of sound pressure level (dB) fall in considerations completely dependent on frequency.

Although conversion methods regarding the units are simple and ready available, it is impossible to work out a program effectively is PDOs and suppliers engineering offices deliver results under different unit languages due the lack of clarity and confusions that may arise due unit mixing.

An imperative strategy for raising the bar in target setting is that vehicle level attributes must never fall to the level of basic of performance quality characteristics (having performance quality as a lower limit), as defined in a Kano diagram; and always aim for excitement quality characteristics in order to not only keep the products or vehicle lines competitive, but to be able to gain market share by high quality appealing products. Therefore, since time makes excitement attributes to become performance, and then basic, a culture of continuous improvement in target levels must always be on the move.

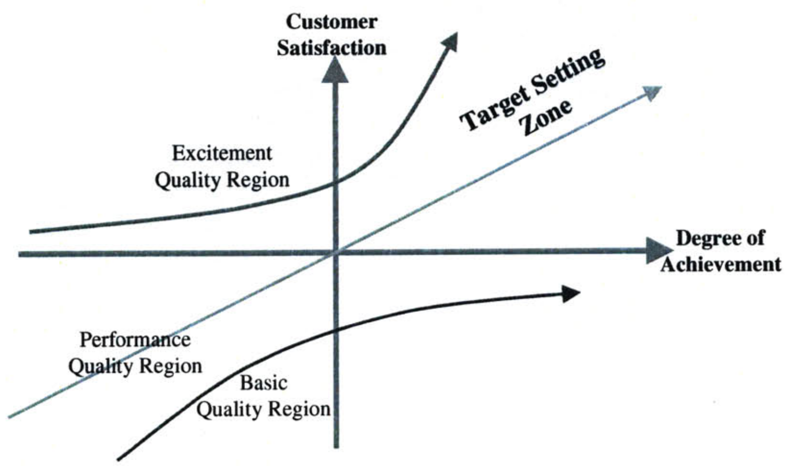

Figure 4-6: Performance Attribute Target Setting Zone

4.3.3. – Targets and Requirements Cascading

Appeal correlated targets need to be correctly cascaded to components in order to have an aligned work among design engineers for system, component, and vehicle level. But cascading a target is not as easy as cascading a requirement, since requirements may refer to functionality and reliability and often are created special for a component or system. In order to be able to properly cascade a target, refereeing it as a value correlated to appeal designated to vehicle level performance; it is necessary to understand the structure and the way each system and component contributes to the delivery of perceived performance.

This process requires the definition of critical parameter of the system which includes measurements of the transfer paths, which are the channel that transmit the inputs of components and systems to the user. As an example consider an A/C compressor, which vibration characteristics are transferred to the user in the form of audible noise and vibration on the steering wheel and seat, and this vibration has transfer paths which include the A/C suction and pressure lines, the engine, the body sheet metal structure, etc.

Proper characterizations of these paths must be measured in order to be able to predict how a different proposal of compressor could perform without the need of an actual prototype when just a reference is needed or a design of experiments is filtering its many possible options.

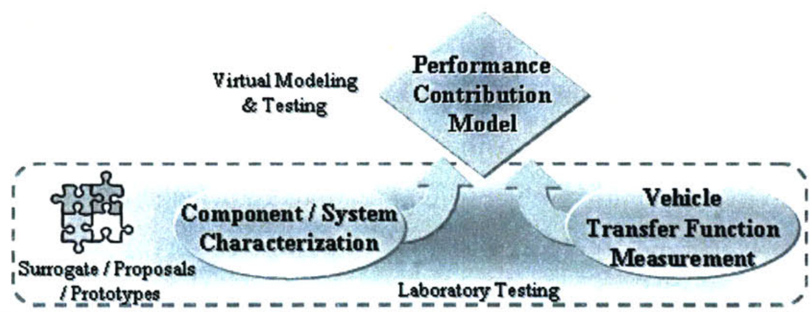

Figure 4-7: Attribute Performance Modeling Framework (Clapper, 2008)

The design of experiments technique provides an idea of the most optimal combination that will deliver the desired function. And performance contribution model serves to give predictions of the overall performance at a vehicle level stance without requiring creating that many prototypes.

It will be critical to perform as many evaluations as possible using virtual tools so as to reduce the developmental costs; nevertheless, physical evaluations will be critical to correlate computer models and perform final validations.

Once an optimal solution gets identified, it is important to perform a sensitivity analysis to understand how the different Xs or noise factors identified in a P-Diagram will degrade or change the desired output.

Emerging and young organization present a clear behavior for managing requirements where a lack of proper target cascading makes the engineering team not to be able to identify the specific targets for systems or component, or if they are identified, no proper action is directed toward meeting them toward a vehicle level target. In this cases is the behavior is to expect that when the prototype is built, the vehicle level target (if they were defined) are going to be met or the vehicle will prove to be subjectively satisfactory, in what we could call the "Home Run" or "Lucky Shot" approach.

4.4. – Previous Knowledge Retrieval

"The amount of knowledge that human beings have exposed is far in exceeds of the amount of knowledge that any person can master and efficiently apply in solving problems." (Grady, 2007)

It a common reality among product development organizations that most of the problems faced across de development of a vehicle have already been seen somewhere in the past, and even for new problems product of new combinations of issue, combinations of previous solutions may give light to the new complex problems that arise as systems grow in complexity. Complex problems require more knowledge and tools that any one person can master. (Grady, 2007)

Complex problems must be decomposed into simple problems so lessons learned can be applied, being the requirements systems one of the best alternatives for a pull ahead of knowledge.

Especially for growing organization the correct integration of design rules and requirements, with the capability to integrate at the same time the knowledge, capacities, and resources from the organization are key enablers for a future product development system. (Aguirre, 2008) Every time an issue appears on a project there is a design for the operative system in order to solve the issue but sometimes is not properly documented or its documentation is not of easy access to the engineering community.

The development and validation phase from the product development process is by far the greatest generator of solutions and lesson learned, requiring a proper system for its documentation and usage in future applications.

As an example consider the generic entertainment system of a vehicle (composed mainly of radio tuner and CD/MP3 player), which sometimes are different in a same vehicle line depending on the region in which it will be sold mainly by cost purposes. Even though there are the same, the tuning frequencies may vary as well among regions.

This creates the need of performing extensive testing over radio by local organizations and most of the times problems related within the development and validation phase are related to how the test is run, for solutions in specific systems, components, or even vehicle testing. In this case a radio was running a design verification plan with the supplier, with successful results over it bench testing methods but the system was failing on the field and on evaluations made at vehicle level.

The reason of the failure was the testing condition since the radio was tested horizontally over the bench, and on the vehicle it was installed at a certain angle which was making a difference on the loads and vibration effects over the system functionality. Examples like this need to be recorded for increased robustness and firefighting avoidance in next developments.

The goal of this part is to propose the framework of operation of a system capable of merging the current requirements databases with the previous knowledge acquired in past programs and projects. Achieving a self learning system that provides input of not only what is going to be required to verify and validate, but also that provides information of the optimal path to perform this verification and validation applying the knowledge available could be a key differentiator that enables an organization to save time applying smart testing solutions in the definition of Design Verification Plans.

Current requirements systems, at least at Ford Motor Company are managed in an electronic system that related a requirement with its testing method, which in design verification plans it also relates scheduling and resources used in the requirement verification. But there is no information about the origin of a requirement or why it is used.

At the end of a program the lessons learned are captured but not properly used in future programs, and a relational database system for capturing lessons learned is recommended in order to solve the problem.

Lessons learned and issue solutions identified must be recorded as usually but within a table format shown in Figure 4-8, relating in a table the findings and insight within the development of a program to the requirement, testing method, program characteristics, and attribute of functional part affected in order to have a close tracking of the genesis and environmental particularities of the lesson to be transmitted.

Lesson Learned and Issue Solution	Requirement ID	Requirement Key Words	Test Method ID	Test Method Key Words	Vehicle Line or Program	Attribute or Function Affected
Solution A / Lesson L. A	XX.XXXX	Steering Effort	XX.XX-R-XXX	Parking Maneuver	A 1– SUV	Chassis Steering
Solution B / Lesson L. B	YY.YYYY	Clutch Slippage	YY.YY-C-YYY	Slip Start	B2 – Car	Powertrain
Solution C / Lesson L. C	ZZ.ZZZZ	Belt Tension	ZZ.ZZ-L-ZZZ	FEAD Tension	T 3– Truck	FEAD NVH

Figure 4-8: Relational Database Proposed Structure

Requirements and testing methods have special nomenclatures within PDOs in order to be tracked and identified, which are now currently related to its names and key words in order to be easy accessible within the engineering systems. This is to be the key link among systems in order to build a bridge between the electronic storage of design verification plans and lessons that need to be applied for the successful test and development phase of a program.

Separate field must be used for every requirement and method in order for current programs in development to access previous knowledge by the requirements and methods defined within their DVP, but key words field for up-front and currently development to identify lessons learn by requirements with relation to their areas being developed.

Vehicle or project programs and functional area or attribute are identified in order to be identified by upfront vehicle development, and refer to previous programs or core areas where knowledge was captured.

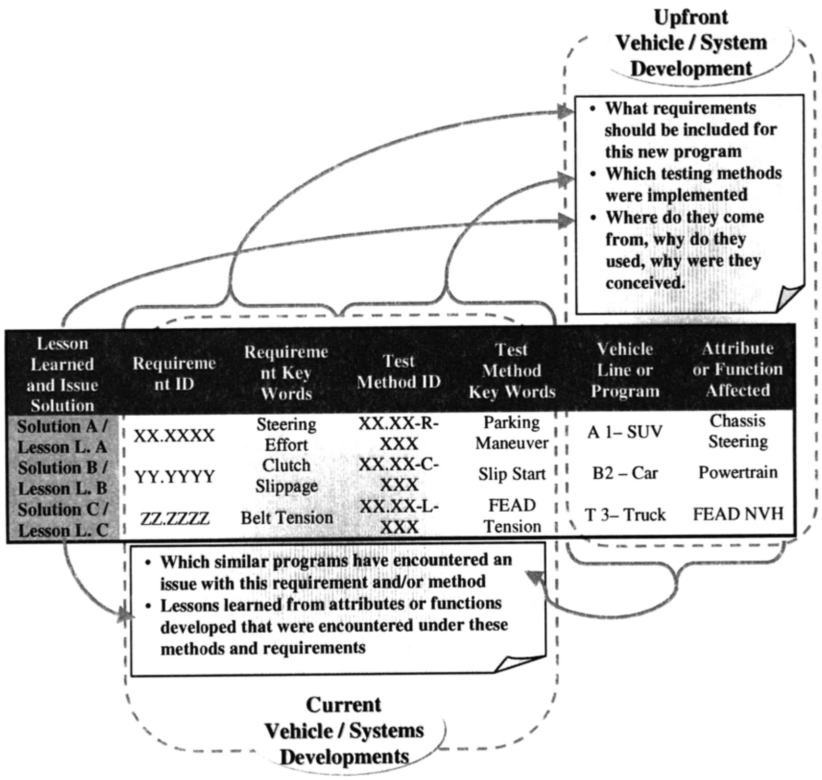

Figure 4-9: Knowledge Management Operation Framework

4.4.1. – Advantages and Risks for a Product Development System

Operating a second database fore the requirements processing and elaboration of design verification plans may add complexity and workload to a product development office within its development and validation phases. But the avoidance of firefighting due unexpected issues that could have been prevented will enable the organization to save time and resources by developing a robust DVP with less probability of leaving gaps within the programs that could represent late development or field issues. Some of the advantages identified by the use of a similar system are:

- Bridge to solve the current disconnection between design guidelines, requirements, and testing procedures; which is a common issue seen within any PDO disregarding its maturity level although logically its presence is more deeply seen in young and emerging organization that in mature ones, but still is an issue that persist consuming definition time, and creating firefighting in later states due requirements gaps.

- When a new requirement is created in order to fulfill a newly discovered need, the system will automatically provide a genesis of it, in order for other organizations to understand why the new requirement does exists and why it has to be used.
- For new programs to be developed it will serve as a tool for definition of the requirements that should be taken into account in order to have a robust, and effective system; under an efficient development and validation phase. This is done by linking the vehicle code and attribute or functions affected by the project with lessons learn from predecessor's projects and lessons from attribute or functions developed in the past.
- Clear definition of what elements need to be tested in base of past experience and engineering judgment, helping to define the optimal DVP and avoid verification gaps that lead to field issues
- Genesis of requirements in order to know which programs and reasons generated them, and be able to identify if a new program has to follow them or if it could have some tolerance. A requirement that specifies that top temperature must be below 110ºC; engineer applies it to a program but with our knowing why. Maybe the requirement comes from certain usage conditions or materials, and programs incorporating new technologies may have a different value as limit.
- Engineering judgment over historical requirements usage, enabling the PDO to simplify requirement and identify what does need to be tested and how.

The only identified risk identified is the abuse on lessons learned that could damp the evolution of new testing methods and processes. When lessons learned are carried to far they become a danger since they can become standard practices that doesn't evolve. If the market or environment is not stable make those lessons impractical and misguide from what the customer really wants.

Comparing environments and unlearning the obsolete makes the teams to tune with reality. (Harvard Business Review, 1995

4.4.2. – Needs for Future Work and Next Steps

Intensive IT work is required for the implementation of the system, creating an SQL system capable of searching information through the tables created for each program and development through a "Join[1]" command requiring to have all the lessons learned tables properly stored in the same location.

Within the field in the current requirement and DVP management system, two additional field need to be created in order to provide a link to the search results within a DVP. One dedicated to relational search within Requirements and Test Methods; and another relating Functional Areas and Attributes, and similar programs to the one currently developed. Both search fields offering an excel file with the compiled results for the relations to search.

Since most of organizations that are regional branches from corporate offices are not able to modify the current requirement management systems. Completed DVP will need to be exported and inserted into a local system in order to perform the relational search. This could represent and advantage since simpler applications such as Microsoft Access and/or InfoPath could be implemented faster and easier due the low volume of information managed just by a regional PDO.

4.5. – The Biggest Small Car

Derivatives from a global portfolio is one to the main development opportunities for growing emerging organizations since both serve to develop engineering capability with projects with an adequate complexity level and at the same time enables multinational organizations to deliver vehicles adapted to the very local needs of customers among regions increasing the success of a company both locally and in a global perspective.

The easiest and first approach for organizations is to build derivatives using components and systems already available within the company adapting to the specific environments and vehicle platforms. This was such the case of a project where a successful low cost compact vehicle from South America which was only built in manual transmission versions, needed to be further developed and redesigned for automatic transmission

[1] Join is a feature of relational databases for getting data from more than one table.

capability which was a customer need from the demanding fragmented Mexican market that required a low cost compact car but with automatic transmission capability.

This derivative was build using a South American vehicle redesigning it and developing in order to be able to use an automatic transmission system from a similar vehicle from the Asian market. The project seemed of low complexity since the vehicle were similar, coming from a same predecessor although following different evolutionary paths and the engine used was the same in family, size and shape.

The derivative was built but as strong issue was found with a FEAD[1] since a strong vibration was perceived both by audible noise and steering wheel vibration.

The study of requirements was simplified by the common characteristics among the vehicles and systems to be used, although deep analysis on requirements and targets failed to detect the items that made different the South American and the Asia engines, which has the same high level design but different elements to deliver and NVH attribute.

Although the project was a great success that not only filled a gap within a market, but boosted the growth and confidence over a young organization, it still became a valuable lesson learned in the development and requirements by the need to analyze the difference between component that apparently are the same or similar but that are built for different regions and attributes delivery.

4.6. – CASE – The Low Cost Work Truck

In emerging markets there is an urgent need of adapting global products for the specific local needs, which have to be delivered over quick response times in order to not to lose the opportunity with the product being prepared by other brand or brought overseas.

Mexico is a region with a fragmented demanding market where customers have grown more sophisticated and demanding, and previous unheard reliability levels are now the standard. At the same time customers are more aware of the attributes of the product and sensitive to the difference between products, being attracted to cheap and easy solutions to their own needs.

[1] FEAD stands for front end accessory drive, being the parts located in the front of the engine such as the band, pulleys, and components acted by them.

Ford of Mexico have endured the challenge for many years of keeping a truck platform, intensively modified year after year in order adapt new corporate designs into the Mexican environment/road conditions, to a reasonable price of the product in accordance to the specific customer needs.

The boundaries for the development are defined as an emerging secondary market, attacked under a systems oriented approach, with the objective of delivering a cost effective, reliable work truck.

The Ford of Mexico engineering team was able to identify correctly the needs for a customer demanding a work vehicle with them must only include attributes derived from the local work conditions and non daily leisure usage.

This leads to the inclusion of new requirements whose fulfillment ensure the reliability expected, while removing other requirement which exceeded other attributes characteristics that the customer in not searching in a work truck mainly related to appeal.

Flexibility in requirements adjusting, and quickness in adaptability makes a valuable quality than growing organizations in emerging markets must pursue in order to take advantage of global portfolio which in just Build-Up strategies would fail to meet the demanding needs of a fragmented market.

4.7. – Summary and Conclusions

Requirements must be adequate representations of customer needs and programs definitions on engineering units that can be understood and measured. They have to be defined at the definition phase of every program trying not to shift them or at leas not in late stages of the development.

Requirements definition must not be stated in the shape of solution but rather the output expected for the system. Requirements and targets need to be set into a boundary of specifying the desired output or behavior of the system and not to include a system solution into it, being documents with an easy evolutionary path in order to adapt to evolving customer needs.

Targets should be the best representation of the customer need, as a requirement of what must be delivered at full vehicle level, being objective and measurable. They must part from the highest level possible and quantified in engineering terms based on appeal ratings and measurements taken from competition products.

Targets and requirements must be cascades thoughtfully into system level targets with the help of the engineering community and tools such as cause and effect diagrams, and QFD in order to create the enablers for the design engineers to provide parts and components oriented to the fulfillment of vehicle level targets.

The speed, efficiency, and quality of problem solving will depend on how rapidly and efficiently the organization can access the right data, haw quickly the information moves from one source of knowledge to another, how well the organization uses the data it has to establish critical relationships and linkages, and how effectively the organization captures what it learns from experience. (Wheelwright & Clark, 1992)

"Delivery is not necessarily the best time to discover the user requirements"
– Alexander's 17th Law of Requirements

Chapter 5

=== Risk Management in Design Verification Plans ===

Perfection is not when there's nothing to add, but when there's nothing to take away.
– Antoine de Saint-Exupéry

5. – Risk Management in Design Verification Plans

The creation and development of vehicles and systems is focused under four great constrains: performance, time to market, cost, and risk. Where performance is dominant making trade off between time and cost, but risk is not very often addresses since it is not well understood (Whitney, 2004)

This chapters attempts to answer a question that is often asked to development teams "Why if the vehicle completed the test well, we are still seeing failures on the field?", in the same way, the approach of this chapter depicts a way for estimating how much is it worthy to test and how much confidence we can feel from the test results.

Testing activities are one of the highest expenditures a product development firm faces while performing a project, and therefore a validation plans involve high technological and financial risk. It is imperative to have the capability of performing informed decisions in order to ensure the robustness of a product through an effective validation plan, keeping it under the budget of a growing or economically unstable organization. Great risk on the product robustness is involved when deciding to keep the testing plans under a budget, but it is also true that in order to reach the launch of a product, it must have a positive business case which involves the development costs.

By applying Engineering Risk Benefit Analysis tools over the probability distribution for system/component failure toward Redesign & Field Failure Potential Costs (Based on warranty & TGW data) with the Test & Development costs; within a simple system optimization.

$$Test\ \&\ Development + \mathrm{Re}\,designCost + PotentialFailureCosts = Minimum$$

Equation 5-1

5.1. – Test Failure Structure

How possible a component or systems is to fail? That's a question full of uncertainty, which drives a product development organization to tests exhaustively until finding how reliable a design is. Quantifying this uncertainty is the key for being able to reduce the tests duration and sample size required to validate a product.

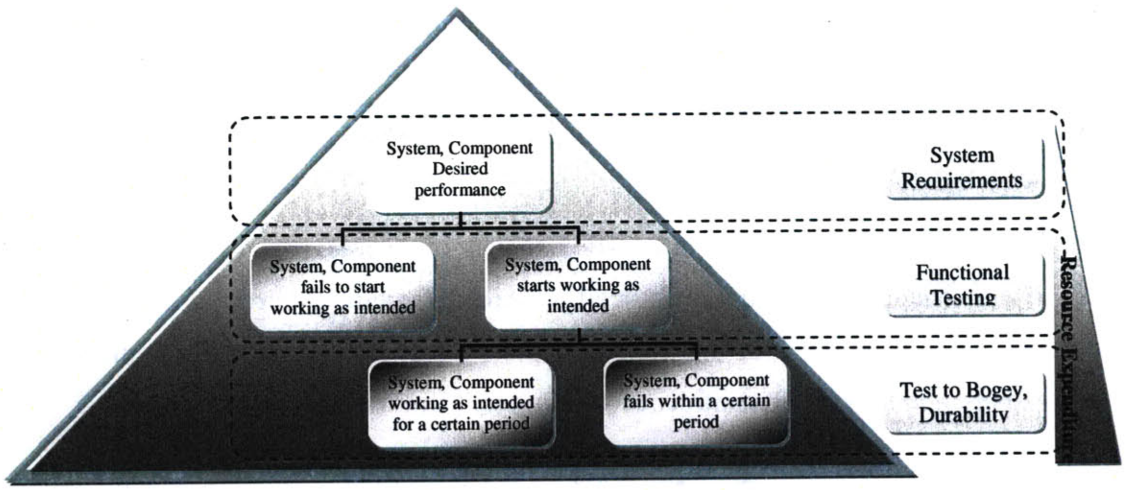

Figure 5-1: Condition Searched by Test Type/Resources Spent

Figure 5-1 pictures the quest of verification and validation is a similar way to the concept of reliability, since testing activities are in the quest of verifying that a system will perform as it is intended and that this performance will remain for a certain period.

Extensive testing is a way to ensure the reliability of a product, since the system can be tested to the whole extent of it designed life, using many samples in order to ensure the confidence of the results. But extensive testing is very costly and a product development organization must know how much value it will be obtaining from their test activities in order to efficiently allocate the resources for it.

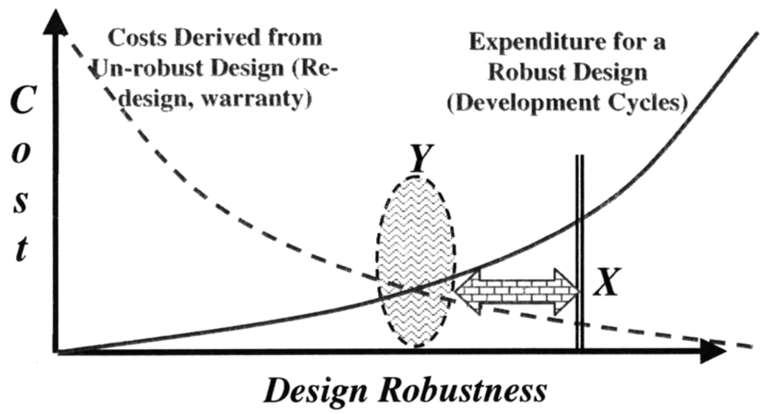

Figure 5-2: Design Robustness and Un-robustness Expenditure Charts modified from (Grady, 2006)

Figure 5-2 describes how design robustness is an investment that product development organizations must pursue but keep in balance in order to have able to sell a reliable product at an affordable price.

Part of the elements that are used for delivering design robustness, are the resources spent in development or refinement cycles, in which testing teams develop the attribute elements of a vehicle for example until reaching a point where maximum satisfaction is expected to be achieved by the customer.

On the other hand, in order to be able to reduce the time to market, products are often delivered with little development on them based on a time or resource limitation that can be spent within development cycles, resulting in unfinished products or issues undiscovered that represent lower customer satisfaction and higher failure probability of systems. In the same way, often product development cycles are stopped when the product is considered to be good enough to be introduced into the market, but not as good as the development team would have liked to deliver a product to their customers.

As mentioned by (Kleyner & Sandborn, 2007), there is an optimal life cycle cost which is shown in Figure 5-2 as the oval marked with the letter Y, but in the competitive environment that the automotive industry faces, it is important to value the benefits of a reliable brand name, such as Toyota has done over the years.

Achieving high reliability products require quite an investment on design work but also in testing work, since testing via development cycles or validation, are intended to discover any possible failure mode that could reduce it life or the satisfaction it will provide to its end users.

The need to pursue confidence and reliability in a design, to achieve a certain level of robustness, starts from the uncertainty of the probability that a system has of failing when performing in the field.

5.2. – Testing in Development Cycles and Validation Cost

Testing is one of the most resource consuming sections of the product development process, and although it is often seen as a non value added process it is yet critical and unavoidable. Optimization of this process by adopting lean principles are needed in order to keep competitive the over all product development process while not endangering the robustness of the design thru its design life. A relationship between the testing costs and the expected warranty costs due to unreliability must be clearly defined in order to keep risk at the most affordable price.

The relationship between testing costs and warranty expenditure is done under the assumption that testing guarantees that the system will perform with a certain level of reliability, by taking a sample population and verifying that it meets the requirements established for the product. At the same time it is consider that testing improves the reliability of products under the assumption that any problem found during validation is properly taken back to design and development stages, and fixed accordingly.

Validation is a significant expense that is decomposed on Equation 5-2 since it involved prototyping and instrumentation, in conjunction with the labor, materials, and energy invested in the realization of a procedure. This simplified approach (since it doesn't take into account the depreciation and maintenance of facilities and equipment) needs to be used to make trade offs and establish an optimization path to minimize costs as shown in Equation 5-1. (Kleyner & Sandborn, 2007)

$$ValidationCost = TestDuration \cdot (Labor + Materials + Energy) \cdot {}^{Samples}\!/_{Capacity}$$
$$+ Samples \cdot (PrototypeCost + InstrumentationCost)$$

Equation 5-2

5.3. – Warranty Data as a Surrogate for Design Verification Plans

It is difficult to estimate the failure rate of a component or system in the field when it is on a design stage, the typical product development organization supposes that their design and development efforts will produce a reliable system, but reality has proven that achieving a 100% of reliability is improbable. Reliability increases as more resources are used during the development of a product, and product development organizations must balance their product budgets and achieve the best possible results, keeping on a target price that the customer will pay for the product.

The moment to allocate product program resources to the testing phases is important and in order to achieve competitiveness, as product development organization must know how to keep their expenses to a minimum. Estimating the failure probability of their design and the warranty costs associated is uncertain but important in order to decide how much resources are going to be invested toward the development of a system, assuring a certain reliability level.

The warranty and quality metrics kept by a product development organization give the tool for estimating the failure probability of a future system.

When considering the usage of warranty information as a surrogate data for design verification plans, the following large assumptions have to be made:

- Every warranty claim reported represents a failure of a system in the field. This is a very important assumption which can significantly deteriorate the quality of the data to analyze, since many times a system could present a failure in the field and not be claimed at the dealer.
- On the other hand, claims reported on the system may mislead the warranty analysis on cases where a failure is reported but belongs to a different system, or it not a failure after all.
- Every dealer service group has the reports claims with the correct data (mileage, months in service, vehicle characteristics), and under the right categories of failure modes that generated the failure.
- Vehicle usage is similar among customers in order to assume that every unit is a representative of a trial or sample.

5.3.1.- Analytical Warranty Data

It is important for any product development organization to gather properly all the feedback data and information from the customer, in order to be able improve their designs and continuously evolve toward the customer needs and expectations, both in quality and performance.

For a growing organization, the right point to establish a robust warranty information gathering system is at the Local Product Control stage, since the operations at this stage relies on PVT activities.

Through a careful and extensive classification of failures, it is possible to store great quantities of information for each particular failure reported in the field.

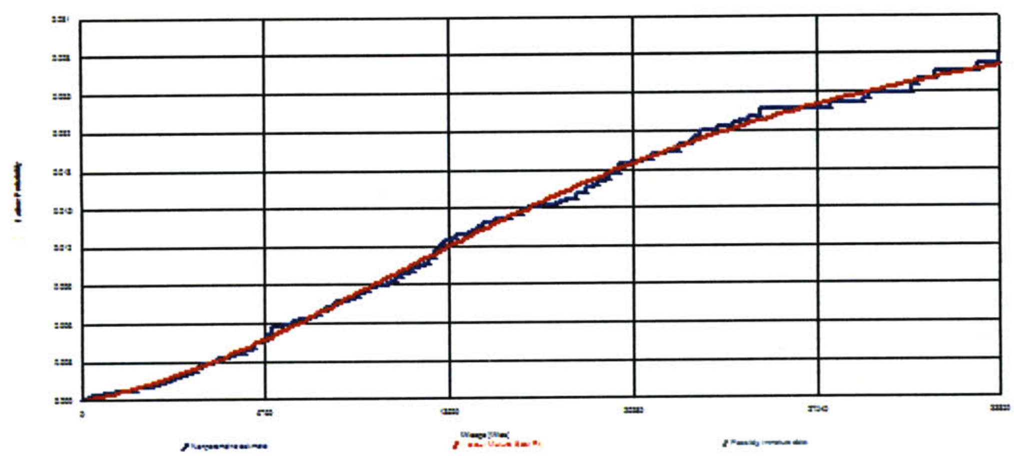

Figure 5-3: Failure Probability vs Mileage - CDF (Ford Motor Company – Analytical Warranty System) System Information Erased and Image Resolution degrade on Purpose.

Growing organizations that are branches of a more mature and developed organization may inherit systems with the required capability or even exceed the engineering needs to determine the failure probability of components and systems in the field.

In case of organizations with well developed warranty record systems or young ones keeping simple records of it, usage of warranty data in order to obtain a failure probability must be used with care due the phenomena described by (Krivtsov et al, 2003) as "data maturity".

Data maturity refers to the fact that not all the units produced for a model year under study were produced at the same time, and neither introduced into service at the same time, representing that units have different time in service and usage. Therefore the number of unit reported in claims doesn't represent all the failures that a recent model year could present, but rather only the ones presented by the units that reached the mileage or time in service point where a failure occurs.

In order for warranty data to be useful, it must be from a similar system to the one being used, which lead that the data to be used is recent since the new design may be a derivative or evolution step, in those cases the maturity of the data could be incomplete.

As an example, if we take the warranty claims gathered for the last year vehicle produced, we could find that a particular system under study presented "X" warranty claims from "Y" units that conform the volume produced. It would not be accurate to consider that $^{X}/_{Y}$ would represent the failure probability in the case of a binomial distribution. Normally it would take a lot of time for all the units of a model year to reach maturity and offer to the engineers accurate data regarding the performance of a particular system, time which is not available under the current constrains that companies have and the limited time to market available in order to remain competitive.

Analytical analysis from the units as they reach maturity is good enough to determine the parameters that describe the quality performance of the units that have achieved a certain period of time or mileage, which could statistically be used for elaborating predictions of the number of failures for the entire population at a time or mileage that is useful for requirements evaluation engineering consideration.

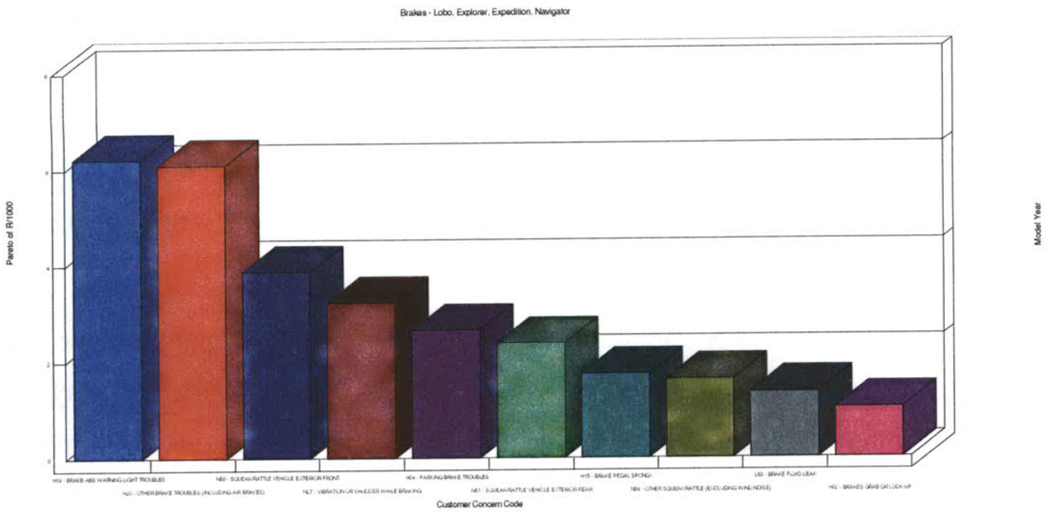

Figure 5-4: Pareto Chart from the systems with the most claims on the field for a specific product
(Ford Motor Company – Analytical Warranty System) System Information Erased and Resolution Degraded on Purpose.

Through the product concept and architecture it is possible to determine previous systems and components with similarities to the one being developed, therefore, an estimate of failure rates can be obtained through the statistic being kept by the quality office of any product development organization. Within this statement relies other important assumptions that have to be taken for using the warranty data as surrogate information:

- The system whose warranty information is going to be used needs to be similar to the one to be implemented in the products being developed. This is required in order to consider that the warranty data is at some extent representative of the behavior that the system to be developed would have, at least in its mathematical wear characteristics.
- The conditions (usage, environment, etc) between the surrogate system and the one to be developed need to be similar in order to be able to consider that the noise factor that originate the failure mode are at some extent the same.

In the case of a growing product development organization in an emerging market, this limitations doesn't represent a real disadvantage, since they represent the typical kind of project that an early PDO will be managing. As we discussed earlier, an early PDO manages projects that are derivatives or incremental versions from platforms or systems already in the field where those projects will be performing.

An important problem as noted by Kleyner and Sandborn is that the warranty statistics of a company is not long enough to cover the requirements life spans for a design. In the case of the automotive industry, the life time of a vehicle is intended to be of 10 to 15 year while quality indexes are monitored for shorter times. A solution to this problem relies on the warranty systems that the automakers implement, in order to contain the requirements and specifications of a vehicle to a specific period of time or usage of the product.

An advantage of using an analytical warranty system (AWS) is that the information of the curve that best fits the collected warranty data up to a certain point of time or mileage, is able to be used to create a prediction of the behavior of a component or system to the desired point of time or mileage.

By extending the mathematical behavior of the wear mechanisms that produce a failure, the system is able to provide failure probability data estimations for the end of warranty periods or the designed time life of the system.

Figure 5-5 shows how the Ford Motor Company AWS is able to create failure probability forecast based on the probability distribution that best fits the warranty data available for a certain system.

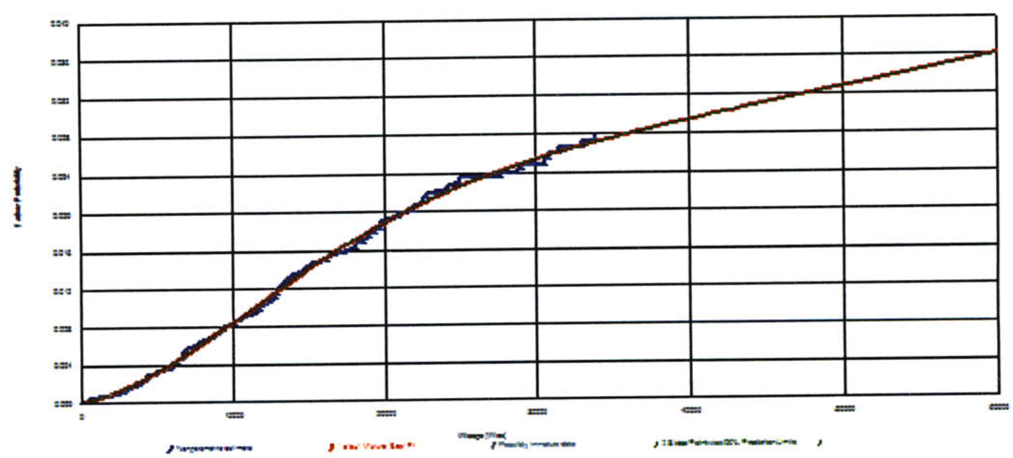

Figure 5-5: Failure Probability Prediction for End of Life Mileage - CDF (Ford Motor Company – Analytical Warranty System) System Information Erased and Image Resolution degrade on Purpose.

The ability to have a failure probability forecast enables the development team to translate the behavior seen in the field, to a full design life estimate. This estimate is of great benefit as prior or surrogate information during the design of new components or systems, where durability targets are fall beyond to the quality metrics usually managed.

For the development of verification and validation plans, having a forecast of the reliability of a system base on a surrogate enables the team to project the expected warranty costs, and decide how to focus the development efforts to the items that cause greater trouble to he company. In the same way, the engineers have a tool to estimate when the resources dedicated to testing could be reduced, considering that the expected warranty costs (as shown in Equation 5-3) don't justify an extensive and costly testing phase.

$$W_{ExpectedCost} = P_{f\_Forecast}(t_f) \cdot V_{ExpectedSold} \cdot CPU_{\mathrm{Re}\,pair}$$

Equation 5-3

Where:

W_{Cost} = Expected Warranty Costs at a time "t" (mileage)

$P_{f\_Forecast}(t_f)$ = Forecasted Failure probability at a final desired time "t_f" (mileage)

$V_{ExpectedSold}$ = Volume of units expected to be sold for specific study

CPU_{Repair} = Cost Per Unit Repaired

Although small growing organizations, or even mature product development companies may not manage sophisticated warranty analysis systems, it is possible to count with other alternatives to manually normalize the data acquired and create estimations of the probability functions that fits to the data gathered.

Commercially available software such as Minitab ® delivers a complete software tool for analyzing warranty results and be able to subtract from it parameters such as the failure rate of components and systems.

At the same point it is important to denote that warranty systems provide information with certain lagging time, since data will be generated some months after production start and every entry is subject to the time the dealer and warranty system takes to process it.

5.3.2.– Advantages from Using Warranty Information

Several assumptions have been pointed out, constrains for the use of this method, and each one of them represents a disadvantage for the extent on which it can be used. But along these constrains and disadvantages, there are powerful advantages that encourage the use of this method.

- Warranty expenditure offers a path for the growing organization to focus on the systems that require more focused testing and development, since it naturally forms a Pareto diagram as shown in Figure 5-6 of the main offenders of customer satisfaction.

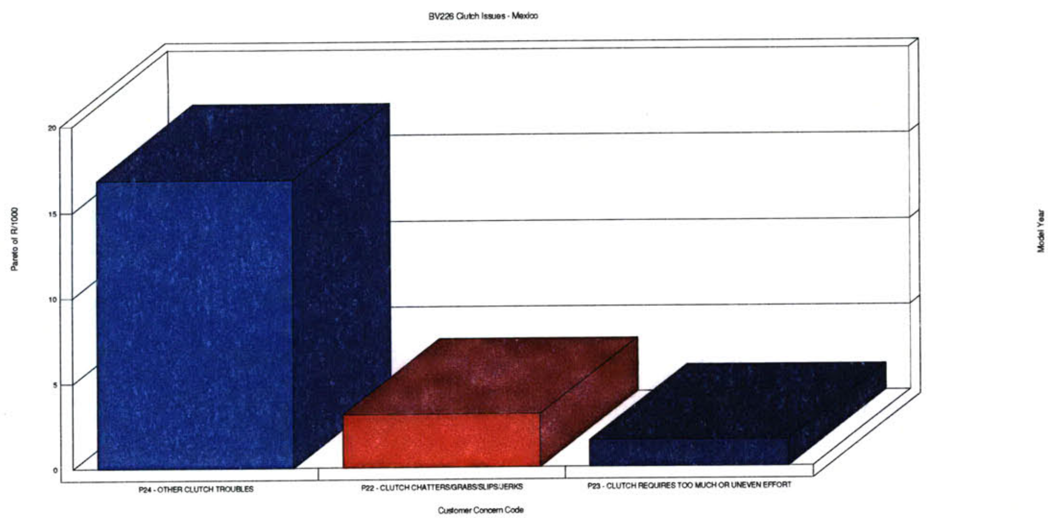

Figure 5-6: Pareto Chart from the systems with the most claims on the field for a specific product (Ford Motor Company – Analytical Warranty System) System Information Erased and Image Resolution degrade on Purpose.

- Provides the engineering organization with information about wear mechanisms trends of their designs and materials, providing useful upfront information for the design of new systems.
- Enables a path for likelihood determination for reducing sample sizes for new test developments.
- Show the actual cost paid by lack of reliability (Equation 5-4), which can be used a reason within the business case of costly test equipment, or to decide which tests could be reduced in low budget programs.

$$W_{Cost} = P_f(t) \cdot V_{Sold} \cdot CPU_{\text{Re}\,pair}$$

Equation 5-4

Where:

W_{Cost} = Warranty Costs

$P_f(t)$ = Failure probability at time "t" (mileage)

V_{Sold} = Volume of units sold for specific study

CPU_{Repair} = Cost Per Unit Repaired

5.3.3.– Typical Warranty Reliability Distribution Outputs

Warranty data permits to gather information about the previous reliability of a component or system, as well as to perform a prediction of this reliability up to a certain mileage or time is service. As mentioned earlier, this data is able to form detailed probability distributions for a specific component or system, or to simply develop Pareto diagram that reveal the items that require priority when resources are scarce. The distribution outputs from a system through the warranty analysis may describe different failure rates, which are depicted within Figure 5-7 forming what is commonly known as the bathtub curve which is the sum of the most common distributions that describe the failure rate of a system or component (Weibull, exponential, lognormal).

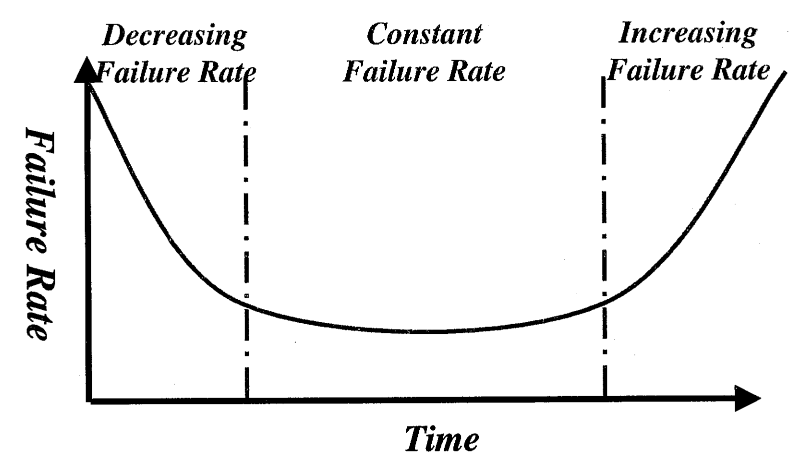

Figure 5-7: The Bathtub Curve - Failure rate VS Mileage/Time

Results from warranty analysis provide probability distributions that help to understand the behavior of the system, identifying the trends that the failure modes follow within the life of a system on the field. These representations are through mathematical models of failure rates both cumulative and in density, being the most common ones in particular for the automotive industry:

- Lognormal Distribution – The lognormal distribution is used to represent systems with increased failure rate, which behavior is reflected by wear out mechanisms. It is usually used to represent the last "increasing failure rate" section of it, being appropriate for accelerated life testing (Krivtsov et al, 2003). The formulas for cumulative distribution and probability density functions used for prediction of failure probability of components and systems at their design life times are shown in Equation 5-6 from Minitab help.

$$CDF_{Lognormal} = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi\eta} \cdot t} \cdot e^{\left(-\frac{(\ln(t)-\mu)^2}{2\eta^2}\right)} dt$$

Equation 5-5

$$PDF_{Lognormal} = \frac{1}{\eta x \sqrt{2\pi\eta}} \cdot e^{\left(-\frac{(\ln(x)-\mu)^2}{2\eta^2}\right)}$$

Equation 5-6

Where:

η = Scale Parameter

μ = Location Parameter

- Exponential Distribution – The exponential distribution is used to represent systems and components with a constant failure rate. Although it is unable to represent a complete behavior of a system or component, it is useful only for specific sections of the life of product, or for service component that have to be replaced at a fixed amount of time. It represents the flat or middle region of the bathtub curb, and not the cases of early usage and wear degradation (Krivtsov et al, 2003). The formulas for cumulative distribution and probability density functions used for prediction of failure probability of components and systems at their design life times are shown in Equation 5-8 from Minitab help.

$$CDF_{Exponential} = 1 - e^{\left(-\frac{x}{\eta}\right)}$$

Equation 5-7

$$PDF_{Exponential} = \frac{1}{\eta} \cdot e^{\left(-\frac{x}{\eta}\right)}$$

Equation 5-8

Where:

η = Scale Parameter

- Weibull Distribution – The Weibull distribution is of great use for failure analysis since it is able to represent any of the sections of the bath curb, although it is most commonly used to represent decreasing failure rates from the "infant mortality" section, being appropriate for accelerated life testing (Krivtsov et al, 2003). The formulas for cumulative distribution and probability density functions used for prediction of failure probability of components and systems at their design life times are shown in Equation 5-10 from Minitab help.

$$CDF_{Weibull} = 1 - e^{\left(-\frac{t}{\eta}\right)^{\beta}}$$

Equation 5-9

$$PDF_{Weibull} = \frac{\beta}{\eta^{\beta}} \cdot t^{\beta-1} \cdot e^{\left(-\frac{t}{\eta}\right)^{\beta}}$$

Equation 5-10

Where:

β = Shape Parameter

η = Scale Parameter

In the Weibull distribution, it is very easy to represent any region of the bathtub by only changing the shape parameter of it. As shown in Figure 5-8 the shape parameter defines the section of the bathtub that can be represented being the exponential distribution an special case of it.

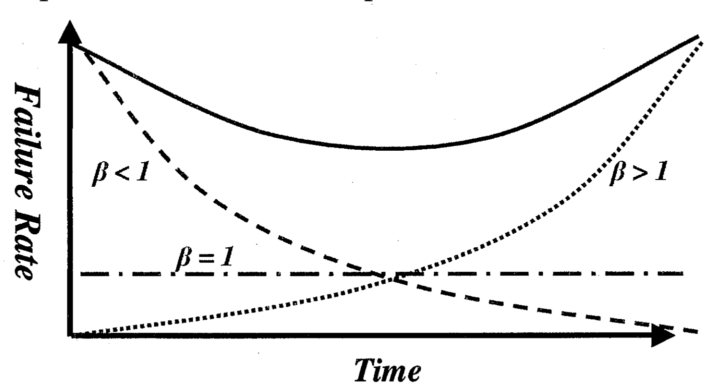

Figure 5-8: Flexibility of the Weibull distribution thru its shape paramenter

- Mixed – Weibull Distribution:

Unfortunately, using warranty data has the disadvantage of grouping every type of failure mode that could produce a failure in the component or system. This disadvantage results in distributions that represent one or more sections of the bathtub that have to be merged together in order to understand the full behavior of the system.

In the failure analysis from warranty data of almost any system or component in the automotive industry, it is possible to find that systems present an early failure rates that are different from a stabilized rate described by wear mechanisms. This is the result of a population in which not every system is equal, but we rather find subpopulations of elements that failed early in time and subpopulations that endured their design life (Attardi et al, 2005).

The mixes in the population present elements that may vary it characteristics due the following reasons, which have to be studied carefully in order to filter the warranty data to the exact parameters and time frames that are useful for studies.

- Slight design changes – Since not all units are manufactured at the same point of time, and an opportunity to find slight on-going design changes within a model year is possible. Especially when a system present reliability numbers that are lower than expected or a field issue is important enough, actions will be taken in order to correct the problem at any point of time. Cost reduction actions are often found as ongoing design changes.
- Manufacturing process change – Cost reduction actions, process change, or tooling wear are factors that may produce a mix in the reliability behavior of system and components. In the same way in a globalized economy automotive suppliers may produce the same system in several locations which will produce population variation depending the process quality and standardization.
- Usage – A system or component is subject to different ranges of loads and environment depending on the location and customer, but also standardization leads to a system or component to be used in several vehicles lines in different markets which may defer greatly the expected lifetime of the system.

In this cases, which are the most common when analyzing warranty data from the automotive industry, we find a mixture of populations where each one will present a unique failure mode described by a shape and scale factor (early or wear out) reflecting the difference between the weak units that failed to start or failed too early and the ones running thru their design lives as intended. (Attardi et al, 2005) These populations are divided by a mixture proportion "p" which can be derived from the detail expended in the analysis and recording of field failures.

Usually the early failure populations are independent of the customer usage characteristics, or difference within the derivatives where the system is used since their root cause is dependent to material or manufacturing process; but prove to be a key element for the wear out populations since this population is considered since it is more dependent on design characteristics and operating conditions (Attardi et al, 2005).

The behavior described by the mixed-Weibull distribution helps to address in a single expression the succession of event that determines the total reliability of vehicles and systems which is described by:

1. Which is the probability that the system will start to operate successfully.
2. Which is the probability that the system will continue to operate successfully for certain period of time or mileage.

Taking the data from the failure modes found in the population, it is possible to derive a distribution that is able to model several sections of the bathtub curve for failure rates. A complete description of the distribution could be found at (Attardi et al, 2005).

Equation 5-12 show the CDF and PDF functions of the mixed distribution as written in (Krivtsov et al, 2003), which eliminates the inconvenience of applying a different Weibull distribution to describe a group (Xie et al, 2002).

$$CDF_{WeibullMixed} = 1 - \left(mp \cdot e^{-\left(\frac{t}{\eta_1}\right)^{\beta_1}} + (1-mp) \cdot e^{-\left(\frac{t}{\eta_2}\right)^{\beta_2}} \right)$$

Equation 5-11

$$PDF_{WeibullMixed} = mp \cdot \left(\frac{\beta_1}{\eta_1^{\beta_1}} \cdot t^{\beta_1 - 1} \cdot e^{\left(-\frac{t}{\eta_1}\right)^{\beta_1}} \right) + (1-mp) \cdot \left(\frac{\beta_2}{\eta_2^{\beta_2}} \cdot t^{\beta_2 - 1} \cdot e^{\left(-\frac{t}{\eta_2}\right)^{\beta_2}} \right)$$

Equation 5-12

Where:

mp = mixture proportion

β_X = Shape Parameters

η_X = Scale Parameters

An different to describe the distribution is to consider that in the field there are two different kinds of product populations, one the group who presents manufacturing problems (quality defects) and another related to usage failures (reliability defects), considering that the design of the product was done to perform as intended (Majeske, 2003)

5.3.4.– Alternatives to the Use of Warranty as Surrogate Information

- CAE methods offer information that can substitute warranty analysis in the development of vehicles and system, which is best used in cases where the system to develop includes a new technology or radical innovation that makes impossible to have field data of a system with similarities.
- Although in early phases of early testing may not provide representative data from the design intended, it may increase development efficiency and help to discover innovative solutions. (Thomke, 2003) Early experimentation is a key driver to know the tendency of current technology and proposals, reducing at the same time the time to discovery of errors and rework.
- Surrogate testing data is useful specially for program that are derivatives or incremental designs from a previous product, since early testing could be performed making trial modifications to an earlier platform, making quick an early experiments thru informal prototyping. This technique is focused to the encounter of failures which help to discard ideas and options and focus to a leaner development phase. (Thomke, 2003)

5.4. – Reliability and Reliability Likelihood in Verification Plans

Small organizations doesn't have the complete expertise and resources to run the full extent of a design verification plan that would ensure the reliability of a product, this product development organizations must choose wisely their priorities based on their areas of focus, and allocate their resources to where they will capture most value while managing an acceptable risk.

The development of a design verification plan (DVP) is a methodology that requires special care since it will describe the process under which a product will be tested in order to assure certain reliability against requirements and targets with a certain confidence.

As discussed earlier, one of the main functions of a DVP is to gather the required requirements, specifications, and targets; applicable to a system in order to verify and validate its performance and quality. The performance and quality is verified through assigning specifics tests and methods.

How the verification methods are applied to a system is the other main function of the DVPs, since testing must be defined for an enough sample size to guarantee that a specific reliability is delivered by the system toward requirements. This guarantee is the confidence level or likelihood that the results shown thru testing, are representative of a specific reliability for a large projected of units to produce.

As units run in the field, test samples run a procedure, both running against a specification, target or requirement; fulfilling or demonstrating the compliance toward customer satisfaction thru quality and performance. In both cases what we see are two populations of units, either at the laboratory or in the field where some of them will fail to pass the test while the rest will be successful. This behavior is clearly described by a binomial distribution.

A vehicle, system, or component; subject to a test has to meet a certain level of performance and quality in order to be accepted and produced, considering this fact we can say that any a subject under test who meets it requirement is considered to pass or fail if it doesn't. Applying the same principle to products running on the field, we consider that a system that withstands its design life without any failure on its specifications, is considered a pass, and each product running on the field is mutually exclusive among all the others, which means that the combined probability of a sample to pass and to fail is equal to one. From (Apostolakis, 2007)

$$P_{Pass} + P_{Fail} = 1$$

Equation 5-13

5.4.1.– The Binomial Distribution

A binomial distribution represents the to possible outcomes of a system (e.g. vehicle, system, subsystem, component) subject to a test, being Pass or Fail, in form of a distribution which includes not just one sample under a probability but a population. These population of samples is also considered as trials, therefore the distribution returns the probability of X successes or failures within N trials or samples tested. As seen in (Apostolakis, 2007), when the order of the trials is unimportant the probability mass function of the binomial distribution is represented by Equation 5-14.

$$P(X = k) = \binom{N}{k} p^k \cdot (1-p)^{N-k} = \frac{N!}{k!(N-k)!} \cdot p^k \cdot (1-p)^{N-k}$$

Equation 5-14

Where:

P = Probability of k failures in N trials

p = Probability of the individual failure of a sample

N = Total number of trials (must be an integer number)

k = Number of failures or events recorded (must be an integer)

There is a clear advantage in using the binomial distribution for calculating the number of systems that will fail to meet its requirements, since the simple task of multiplying the individual failure probability against the total number of samples will only represent the mean of the population.

The binomial distribution is a good method for the verification and validation testing since tests are oriented toward searching for failure or compliance against a specification.

5.4.2.– System Reliability with a Confidence Level

The relationship between the failure probability for a system and its reliability is simple, since both are complementary as shown in Equation 5-15.

$$R = 1 - p_f$$

Equation 5-15

The binomial distribution is a useful tool that explains the relationship between the reliability of a system and the confidence or risk level in it.

The confidence or risk level, is the probability o achieving a certain level of reliability for a whole population, based on the number of samples that is feasible to test due the cost of the validation process.

The confidence level could be also defined as the probability that we will achieve at least a certain level of success, the likelihood or probability density distribution of getting a value in the specific set range. A better explanation for this can be seen in Equation 5-16.

$$C = \int_0^R f(R)dR$$

Equation 5-16

Where:

C = Confidence or Risk level

f(R) = Failure mass probability function

Equation 5-16 Represents the probability density function within the range of zero (totally unreliable) to the target reliability established, and represents the probability that the events or trials will fall within that reliability range.

Since in product validation we are interested on the success of the product, Equation 5-16 can be rearranged in order to provide the probability that the product will achieve a probability within the range of the target and 100%, in other words that the product will perform with a minimum reliability.

$$C_s = 1 - \int_0^R f(R)dR = \int_R^1 f(R)dR$$

Equation 5-17

Where:

C_S = Confidence or Risk level for success

f(R) = Failure mass probability function

There is a specific reason form making calculation of one minus the probability density function of the range from zero to the specified reliability, which relies in the ability of being equivalent to the calculation of one minus the cumulative density function of the system. This assumption is supported by Equation 5-13 since the pass or fail events are mutually exclusive and together represent the whole population under a reliability from zero to one.

The cumulative distribution function represents the probability of finding events or trials within the range of zero to a specified reliability, definition equivalent to a probability density function within the same range.

In the case of products testing, the approach shown in Equation 5-16 needs to be tailored a little bit since test samples are discrete events, countable items which can be separated as pass or fail, and represented through a Binomial distribution.

For the Binomial distribution the probability density function is described by Equation 5-18 representing the probability of achieving "k" events within a "N" population at a certain probability "p".

$$PDF_{Binomial} = \binom{N}{k} \cdot p^k \cdot (1-p)^{N-k}$$

Equation 5-18

In order to obtain the probability density function from a range of zero to the specified number of event determined by the probability instead from a point in the distribution, it is possible to use the cumulative distribution function from this binomial distribution as explained earlier, which is expressed by Equation 5-19.

$$CDF_{Binomial} = P(k \leq k) = P(k=0) + P(k=1) + P(k=2) + \ldots + P(k=k)$$
$$\Rightarrow CDF_{Binomial} = \sum_{k=i}^{k} \binom{N}{k} \cdot p^i \cdot (1-p)^{N-i}$$

Equation 5-19

Equation 5-16 can be represented discretely by a binomial distribution as shown in Equation 5-20, where we see the confidence level or cumulative density function of the results of the test. (Apostolakis, 2007)

$$C = \sum_{k=i}^{k} \binom{N}{k} \cdot p^{i} \cdot (1-p)^{N-i}$$

Equation 5-20

In the case of product validation, the requirements, specifications and targets; mean a minimal point of performance and quality to be achieved, to be demonstrated. Equation 5-16 doesn't clearly represents this, since it mostly shows the confidence of achieving a value from zero to the reliability desired, the data we want lies on the fact that the event are mutually exclusive and the combined probability is equal to one as shown in Equation 5-13. Taking this into account the confidence or risk level for success is given by Equation 5-21, as described similarly by (Kleyner, 2005)

$$C_S = 1 - \sum_{i=0}^{k} \frac{N!}{i! \cdot (N-i)!} \cdot p_f{}^{i} \cdot (1-p_f)^{N-i}$$

Equation 5-21

Where:

C_s = Confidence or risk level for success

p_f = Failure probability

N = Number of prototypes to test

k = Number of failed test samples within the population

Equation 5-21 shows us the relationship between the failure probability of a system and the confidence of finding an event within the range of success.

5.4.3.– Success Testing

A common issue when developing design verification plans is that the engineer has to play under a delicate balance of the quality, schedule, and cost. In this triangle scheduling is the task that often is run with the most assumptions which are often wrong ending in program delays and cost overruns.

When performing the verification and validation of a product, the intention of the process is to provide an error free product that will behave in the field as it was designed for. Based on the previous statement, we can make the assumption that validation is planned in a success mindset, leading to what is commonly known as Success Running.

A DVP developed under a success mindset makes the assumption that all the results done in testing will be successful, or corrected in order to be successful. Iterative loops for development and problem solving activities can not be considered by the success testing approach in order to estimate the parameters such as sample size or test duration during the elaboration of a verification plan targeting a desired level of confidence in the corporate levels or reliability established.

Success testing approach parts from Equation 5-21 making the k = 0 assumptions, which mean that we are not expecting any failure from the test to be performed:

$$C_S = 1 - \sum_{i=0}^{0} \frac{N!}{i! \cdot (N-i)!} \cdot p_f^{\,i} \cdot (1 - p_f)^{N-i} = 1 - \frac{N!}{0! \cdot (N-0)!} \cdot p_f^{\,0} \cdot (1 - p_f)^{N-0}$$

$$C_S = 1 - (1 - p_f)^N$$

$$C_S = 1 - R^N$$

Equation 5-22

Success testing gives the formula most often used for relating sample size, confidence, and reliability; for simple statistics considerations, which as was described comes from several assumptions under the binomial distribution distributions concept being feed from different failure probability distributions CDFs and PDFs for determining its parameters.

5.5. – Test Method Acceleration Factor and Duration

Equation 5-22 shows the simplified version of the binomial distribution often used for the statistic determination of sample size. This expression can be upgraded in order by the

Lipson or Parametric Binomial distribution (Kleyner & Sandborn, 2007), expand this equation with characteristics from the test method used to asses reliability:

$$C_S = 1 - R^{N(A_f L)^\beta}$$

Equation 5-23

Where:

A_f = Acceleration factor

L = the ratio between the test duration (correlated to design requirement) and the actual test time.

β = Slope of the Weibull distribution from where p_f was obtained.

The origin of the reliability function is derived from the failure probability distribution describe before, specifically from the Weibull distribution as is next shown:

$$R(t) = 1 - F(t) = e^{-\left(\frac{t}{\eta}\right)^\beta} \;\&\; R_1 = (1 - C_S)^{1/N_1} = e^{-\left(\frac{t_1}{\eta}\right)^\beta}$$

$$\Rightarrow \ln(1 - C_S) = -\frac{1}{\eta^\beta} \cdot N_1 \cdot t_1^{\beta} = -\frac{1}{\eta^\beta} \cdot N_2 \cdot t_2^{\beta}$$

$$\frac{N_2}{N_1} = \left(\frac{t_1}{t_2}\right)^\beta \;\&\; \frac{t_1}{t_2} = L \Rightarrow N_1 = L^\beta \cdot N_2$$

This equation could be expanded using the approach described previously for the management of Weibull – Mixed probability distributions as follows:

$$C_S = 1 - \left(mp \cdot R^{N(A_f L)^{\beta_1}} + (1 - mp) \cdot R^{N(A_f L)^{\beta_2}}\right)$$

Equation 5-24

When meeting a deadline of preserving a schedule becomes critical for an organization, I have been witness or at least heard the decisions of ending the performance of a test before completing the procedure and giving an assessment / sign-off for a product based on the completed portion of the test.

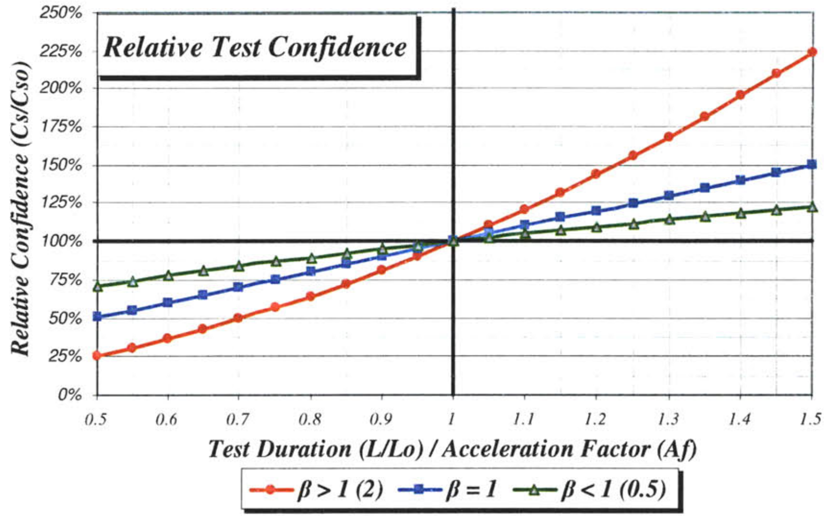

Figure 5-9: Duration and acceleration factor effect on test confidence

Equation 5-24 shows that the duration of the test and the acceleration factor affect in the same point the estimation of the confidence to achieve a targeted reliability. The most common judgment would assume that if a durability test has run the 80% of its duration we could say that the test is as good as completed when time is exerting great pressure toward the team to finish the validation of a vehicle. Over the histories and legends of a PDO we can heard of cases where nothing happened and the assessment was right and time was saved, and others where great field failures are presented and everybody wonders why the issue was not detected during the durability testing.

The values of relative confidence, test duration and acceleration factor are presented as the ration of the input or output value over the original confidence or duration in order to make the assessments in the calculations free of the number of samples to use, or the precise mileage or confidence achieved.

Figure 5-9 proves that this kind of assessment is wrong, based on the fact that some systems may perform linearly to changes in test duration or acceleration factors but in other cases this doesn't happen since different behaviors could be present depending on the shape of their failure probability distributions.

For failure modes with Weibull slopes smaller than one, the confidence is not affected as severely when a test is not permitted to be concluded and the reason is that the behavior of these kinds of slopes refers to an "infant mortality" where most of the failure occurs in early in time or mileage.

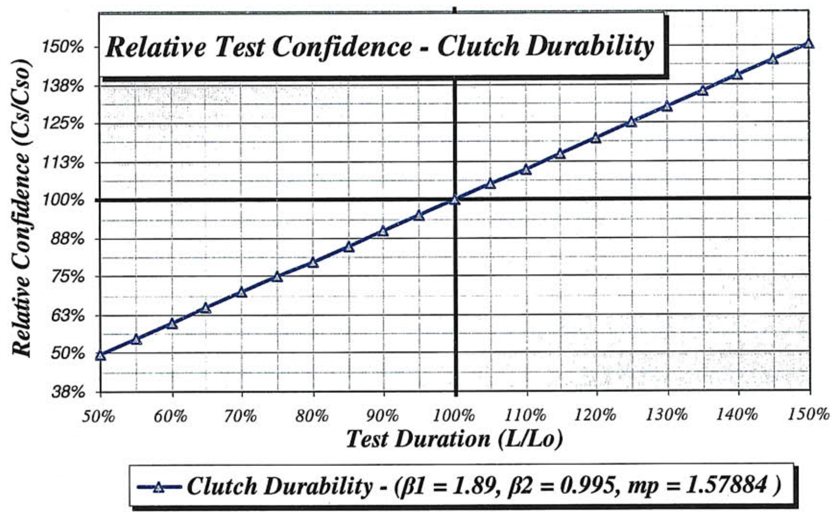

Figure 5-10: Effect of test duration on a clutch system

Cases of systems that present constant failure rates, or Weibull slopes equal or very close to one, such as the case presented in Figure 5-10. This figure represents data from an actual clutch system of a vehicle currently produced, taking the data from actual warranty claims in order to obtain its Weibull mixture parameters which mainly refer to a constant failure rate case.

This case could be one of those where an assessment establishes that stopping a durability test at 80% of its duration gives a representation equal to 80% of the full life cycle and it could be concluded that there will be no surprises in the rest of the test.

On the other hand there are other systems or attributes that doesn't behave in the same way presenting slope values that are mainly focused on wear out system behaviors and cutting off a test may represent a great loss in confidence. Figure 5-11 shows the same vehicle analyzed in the clutch case but now for the NVH performance of the brakes system in a high labor procedure that requires a significant amount of time to perform.

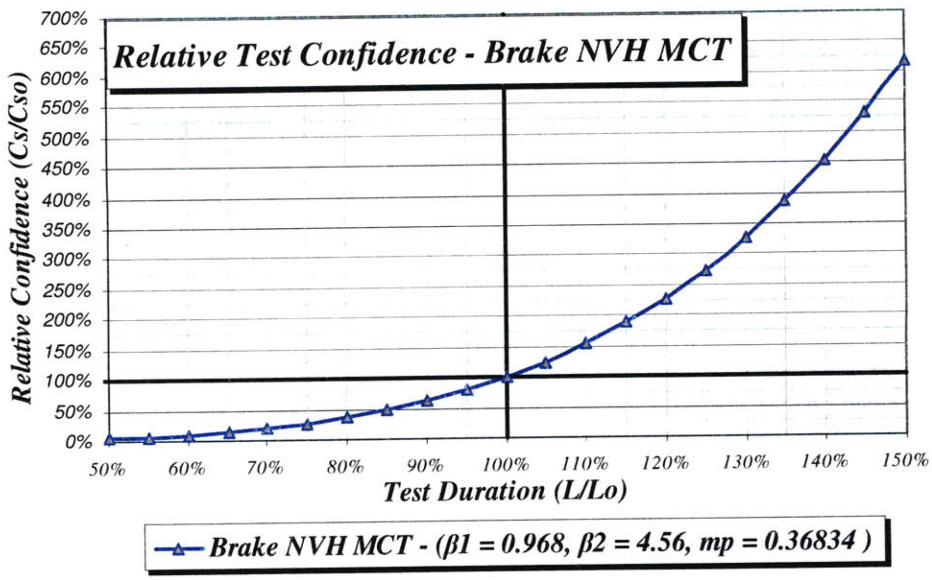

Figure 5-11: Effect of test duration on brake system noise

In this case deciding to stop a test at 80% of its duration reduces the confidence to achieve a given target reliability in 64% instead of the 20% of confidence loss seen before for the clutch. The beneficial side is that if a test is run up to finding a failure rather that completing a test time, systems with dominating slopes greater than one can give much higher confidence value by a small increase of testing duration, as show in Figure 5-11.

It will be always required to sacrifice testing time in order to achieve a milestone or to give an assessment to management, but no test could be cut off without having a proper characterization of the type of behavior that systems have, which has to be determined when the affected systems on a new derivate program are identified for the creation of testing plans.

The acceleration factor is an advantage that testing organizations can use in terms of reducing the samples size and by that reducing the prototype expenditure for testing, or when testing prototypes numbers are low, it is an enabler for reducing testing times without compromise of quality between test duration and confidence.

The acceleration factor is a test method attribute that represents how much real usage does a procedure represents, based directly on correlation factors between the inputs received by a vehicle during real customer usage and the input received during the test performance (an example are the test track durability procedures where correlation dictates that 30,000 km of test durability represents 150,000 km of customer usage). This concept is more clearly shown in alternative testing platforms like bench testing, where the controlled laboratory conditions enable system manipulations to concentrate load exerted on components accelerating fatigue and wear in reduced amounts of time.

Although not able to be tuned per test but rather defined within a test procedure, acceleration is a key factor for the reduction of prototypes, timing, or both.

5.6. – Test Sample Size

The sample size definition is a crucial point within design and verification plans since it defines most of the expenditure of design verification plans since carries the whole prototyping cost as well as the number of iterations to be done at each test sequence in order to ensure that reliability requirements are met.

The traditional analysis for estimating the sample size based on a statistical confidence of the test is shown in Equation 5-22, an approach that is not practical when cost is a main driver and extensive efforts are place in order to reduce prototype expenditure to a minimum.

Several traditional techniques can be used in order to reduce the sample size, being one the acceptance to reduce the targeted reliability of systems. This method is contradictory at some point since the purpose of a statistical analysis for determination of sample size is for ensuring a high reliability delivered, although it can be assumed that early testing will be done under low reliability assumptions with the tendency that reliability will improve as design reaches maturity.

Another important technique is increasing the duration of test time as which proves to be beneficial in terms or increasing the amount of confidence (Equation 5-9) or either designing test methods with increased acceleration factors. Equation 5-12 shows an example of how bench testing with accelerated testing factor may help to reduce the required sample size to prove statistical confidence.

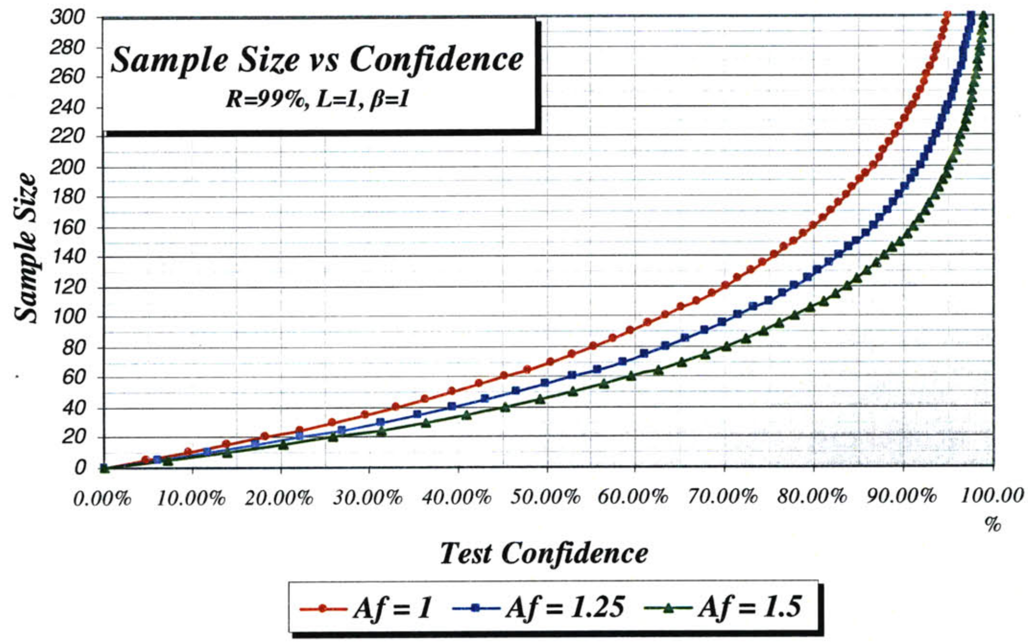

Figure 5-12: Acceleration Factor Effect on Test Sample Size Definition

Another important technique is the use of design for Six-Sigma, although this tool will not be detailed explained within this work due the extensive literature available on the field, as well as the depth of expertise and usage within the automotive industry, specifically in Ford Motor Company.

In strict theory for Six Sigma, if the number of trials is large, the binomial distribution is approximately equal to the normal distribution. In these cases the usage of Six Sigma for reduction of the sample size resides in the usage of this tool at the design phase of the product development system to ensure that each component or system will be produced under minimum levels of variation.

This statistical confidence on the characteristics of components and systems permit to the test planners to assume that a low population represents a large sample size and therefore the likelihood or confidence to achieve the target reliability can be reduced.

Linking the sample size determination with the warranty analysis to derive the failure mode distribution parameters from a component of system, alternatives for the reduction of test sample size can be found using a Bayesian approach.

5.7. – Bayesian Approach for Test Sample Size Reduction

A Bayesian approach makes use of prior information which we obtain from warranty the warranty statistics models described at the beginning of the chapter. The usage of prior information has the assumption that the new model or derivative shares similar behaviors or enough characteristics with the prior model in order to consider it useful, and include it as part of development testing for the new project.

The Bayesian approach is especially useful for emerging and growing organization since its strength is in the development of derivatives of vehicle versions from already existing platforms. In other cases where all the development is new, the Bayesian approach uses the similarities against uses and technologies, which makes more difficult the election of prior information, and have the risk that the prior information may not be accurate leading to wrong conclusions.

Fully accurate or not for product development purposes, the Bayesian approach is still a great tool for upfront product development that should be taken into account specially for emerging organizations with the main activity of developing derivatives, incremental designs, or new vehicles parting from platforms or technology already in use.

5.7.1.– Prior Information (Warranty) for Future Development Characterization

The warranty analysis framework presented within this chapter enables the determination of the failure probabilities of vehicles, systems, and components; at the end of their useful life or at the moment where their durability requirements are reached by extrapolating the warranty information from the mileage that has been reached on the field.

Although the field mileage will never be enough to provide non-parametric information of failure rates at end of life, failure probability distributions can be obtained for performing parametric estimates. These failure probabilities at end of life in serve as the prior information for Bayesian approaches, but still prior information held key for the characterization of the components still to development.

The failure probabilities distribution give us valuable data through their slope (β) and scale (η) parameters in case of Weibull and Weibull Mixed distributions, since they can be extrapolated to define the characteristic behavior of the new systems and components to design.

Within same product lines the shape parameter from Weibull distributions keeps a certain degree of consistency making it useful for upfront application into new models. The scale parameter is still useful although this parameter is dependent on the expected life of each part which could shift more easily between warranty runs due special usage and requirements level. (Kleyner & Sandborn, 2007)

5.7.2.–The Bayes Theorem

As mentioned by Vale, the Bayes theorem offers a delicate framework for using warranty information, surrogate testing information, or early testing information. By taking prior probability distributions as hypothesis, a posterior probability distribution of increased confidence by the analysis results of the priors is given. Being H1 and H2 mutually exclusive and exhaustive events where any event E is know to have a probability of success given by P(E/H1) the posterior probability of the event is given by Equation 5-25.

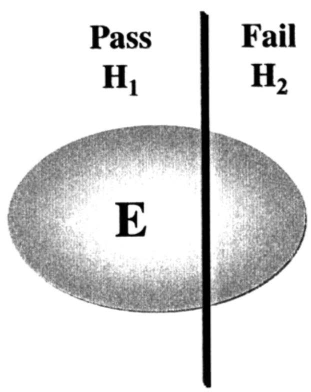

Figure 5-13: Population Segmentation (Apostolakis, 2007)

$$P(H_2/E)=\frac{P(E/H_2)\cdot P(H_2)}{P(E/H_1)\cdot P(H_1)\cdot P(E/H_2)\cdot P(H_2)}$$

Equation 5-25 (Apostolakis, 2007)

Where:

P(E/H_X) = Likelihood or confidence for obtaining the observed test data if the reliability of each unit is H_X.

P(E/H_X) = Prior distribution or that reliability H_X.

The theorem requires that the warranty data used for the analysis to be exactly mutually exclusive and exhaustive events, reason for using binomial distributions in order to have only Pass / Fail domain.

Still for automotive applications the Bayes theorem in full is not recommended for sampling size determination for the design and development of derivatives, since the existing information forming the prior information is based on warranty data consisting of thousands of units it will pull the results of any DV validation which will be less in numbers. Therefore the assessment won't serve to fully assess the real confidence of the new component since it will be biased by the weight of the priors. (Krivstov, 2008)

5.7.3.– Single Point Bayesian Approach

The single point Bayesian approach is a simplified version of the Bayes theorem under the assumption that the prior information is valuable enough that it is only required for new development to increase the existing confidence in the reliability targeted by a reduced amount of samples to test. (Vale, 1997) Therefore this is a method for raising confidence for DV methods.

This method as depicted also by Krivtsov and Kleyner relies on assuming that the warranty analysis obtained failure probability at the end of life or fulfillment of required mileage or time is equivalent as testing to failure results, considering the field failures analyzed as actual testing completed.

The basis for this assumption as described in (Vale, 1997) relies on the little point of testing extensively if we have a close to good prior confidence.

Starting again from the binomial relationships and the data found from warranty analysis, a calculation needs to be made in order to obtain the likelihood or confidence of the design to meet its target reliability based on previous information. Calculation is based on binomial likelihood taking into account failures. The value obtained must be multiplied by a percentage based on engineering judgment, where an assessment of how similar or likely is the derivative to the original platform must be applied.

The second step is to obtain the fictional number N_{prior}, which is a mathematical representation of the samples tested with the prior confidence C_{prior}, under a success testing environment.

$$N_{prior} = \frac{\log(1 - C_S)}{\log(\text{Target } R)} \cdot \frac{1}{(A_f L)^{\beta}}$$

Equation 5-26

For Mixed Weibull data applications it is recommended to recur to numerical methods for the calculation of N_{prior}.

The third step is to calculate in the same way the required N to demonstrate the target reliability with the desired confidence from the general success testing formula. Finally the reduced sample size is the number of samples to test under the described formula for success testing

$$C_S = 1 - \left(mp \cdot R^{N(A_f L)^{\beta_1}} + (1 - mp) \cdot R^{N(A_f L)^{\beta_2}} \right)$$

minus the N_{prior} calculated, assumed to be valid as testing samples used to asses reliability.

$$N_{Total} = N_{SuccessTestingl} - N_{\Pr ior}$$

Equation 5-27

This simplified approach has the constrain or requiring the warranty analysis data to be close to the component or system to be designed to be valid in order for the data to be useful. It is recommended that if the prior confidence is lower that 50% the data is not good for consideration and the project should be considered as a new design.

If the previous system population is completely unable to satisfy the required reliability targets by any confidence, the data is also not usable for Bayesian methods since the component would need to be redesigned for meeting targets.

5.7.4.- Reliability/Cost Trade Off for Minimum Development Costs

The goal of the analysis is to build enough data which in combination with enough engineering judgment, enough information is build for a product development organization for the optimal creation of design verification plans that makes the most of the robustness trade off. This information will help the product development organization to have side by side data of test and development costs vs unreliability costs as shown in Equation 5-1, to decide how much to invest in the robustness of its products by linking Equation 5-2 and Equation 5-3 in an optimization model for optimization of the samples required vs a target reliability and the confidence to reach it.

5.8. – Summary and Conclusions

As Nancy Levenson says within (Whitney, 2004), safety is being free from accidents, accidents are undesired unplanned results, a hazard is the set of conditions that lead to an accident, and risk is the level of hazard that we are willing to take in the trade offs of the iron triangle (Scope – Scheduling – Cost) for reliability environments.

Thru the study of warranty analysis, models of the failure probability of systems and components could be modeled on basis of the parameters of the functions that define them, being one of the most accurate the Weibull Mixed probability distribution since incorporates several failure modes within one single mode, which enables to gather the units that fail soon due manufacturing defects to the one that fail later due wear out mechanisms, having a more detailed and robust model.

The binomial probability distribution can be managed extensively in order to provide useful data to he engineering community for sample size estimations, as well as for assessing when is safe to stop a test or how much would we will in confidence by leaving some test to run more that is defined by the procedure.

The slope characteristics of systems and components can be translated from warranty analysis result to new projects being developed in order to make accurate assessments of the risk of not letting a test procedure to fulfill its duration to a 100%, a situation that creates serious field risks when the pressure to deliver an assessment over testing method that require extensive use of time and resources.

Using Bayesian methods, the development of derivatives or vehicle versions that depart from existing platforms, permit the reduction of sampling for DV testing by using the simplified version of the Bayes theorem while the previous version, program base, or technology/platform used as a surrogate has confidence of meeting the reliability target at least in 50%

"Nothing can be more fatal to progress than a too confident reliance on mathematical symbols; for the student is only too apt to take the easier course, and consider the formula not the fact as the physical reality."

– Lord Kelvin

As Ralph Katz always says during his class in the opening of the MIT SDM course work: "Decisions are not taken based on Data… Decisions are made based on Information"

An engineering organization needs to become data driven and use as much as possible analytical tools for the robust and accurate development of products, but problems solving is an activity that requires more experience than knowledge, and therefore no decision must be made just based on number on a screen but rather analyzing numbers with a great deal of engineering judgment based on the lessons learned that experience has written.

Chapter 6

=== System Development & Validation Testing ===

"There are three principal means of acquiring knowledge . . . observation of nature, reflection, and experimentation. Observation collects facts; reflection combines them; experimentation verifies the result of that combination."
– Denis Diderot (1713 - 1784)

6. – System Development & Validation Testing for an Emerging PDO

A robust and well defined development and validation system is a key element for achieving optimized and reliable products under the most time and cost efficient process that promotes a fast delivery to the end users.

On the third chapter we analyzed the evolution path that defines the context for an emerging and growing organization, being the case of analysis the product development office of Ford Motor Company in Mexico. Contexts, boundaries, and constrains were touched in order to understand the path that is followed by testing organizations as their product development offices grow and acquire greater responsibility.

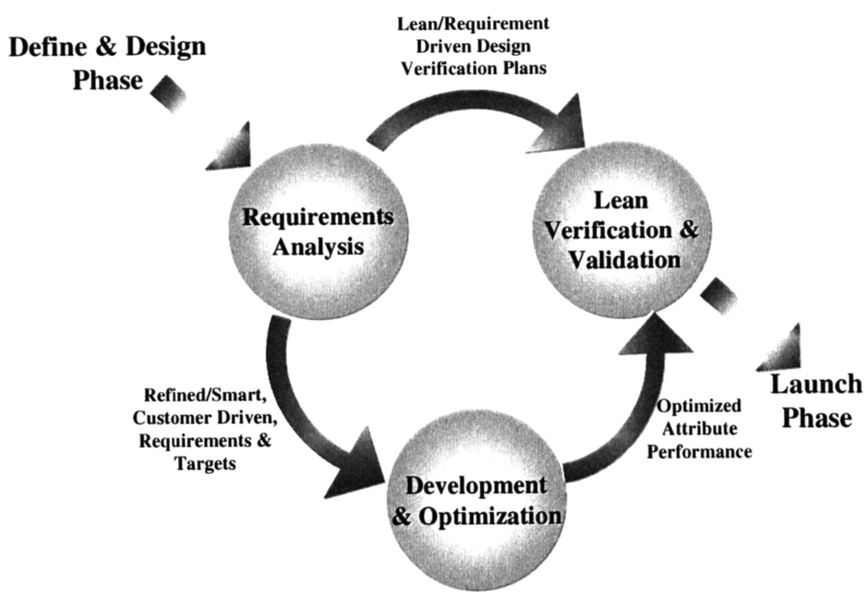

Figure 6-1: D&V High Level Operating Framework

This chapter has the intention to present and analyze a structure for the next step of growth of the development and validation phase of an emerging organization that reaches a point of maturity where the complexity not able to be handled appropriately by the current system characteristics.

The goal as described in Figure 6-1 is to set the starting point for a development and validation platform that use the available resources of an emerging organization toward the generation of refined and smart requirements and targets, enabling the creation of lean design verification plans.

6.1. – Architecture of a Development & Validation System

The current state of the Ford of Mexico development and validation structure, or the structure state being reached shown in Figure 3-5 start to present complexity problems with the old schemes producing duplication and overlapping roles and responsibilities leading to what in chapter 8 is defined as "the little empires".

As stated by Wheelwright and Clark, the right organization for developing any function and/or project is dependent on the size, types and mix of projects to be responsible for; as well as it is dependent from the competitive environment, market, organizational culture, and company history.

Taking into account what was mentioned above, Ford of Mexico requires a test and development structure with clear definition of roles and responsibilities, setting a balanced boundary toward the activities to be performed within the team but at the same time maintaining an united work cell working on a concurrent environment.

Using the same form and function attribute elements as described for a development and validation system before, Figure 6-2 presents the architecture model that involves these concepts with the current context of the emerging and growing product development organization at Mexico. This architecture is based on the current and coming challenges that it will encountered by the Mexican PDO with the goal of having a D&V ready for supporting the test needs from engineering work.

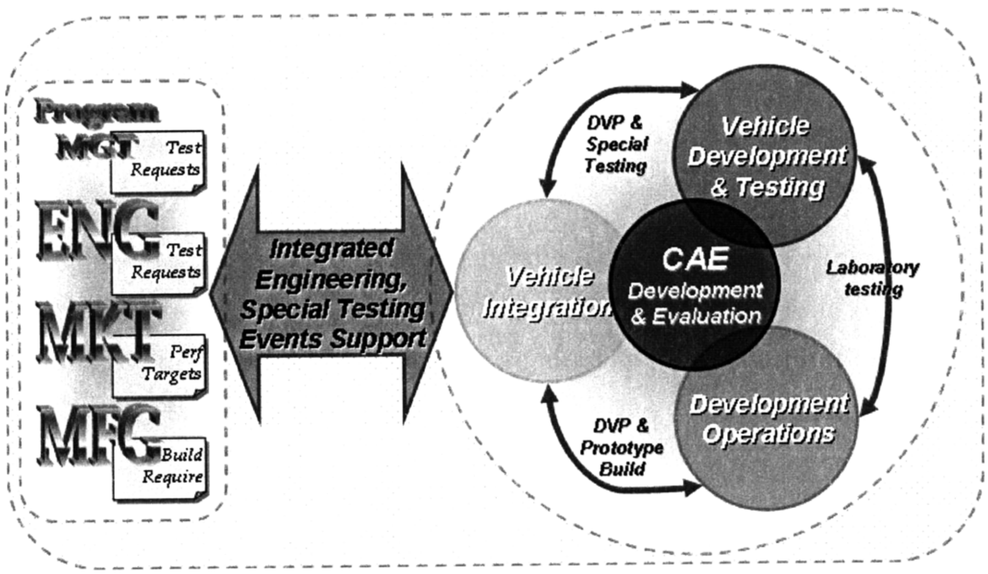

Figure 6-2: FoM D&V Recomended Architecture

The architecture model presented on Figure 6-2 works as concurrent development and validation entity, performing within work cells environments having the development of alternative testing and optimization platforms as a priority where each group must work along in the creation and correlation of models.

6.1.1.– Functions Delivery Decomposition

In mature and complex product development organizations, separated entities exist for the development and validation of their products, which is a suitable structure for organizations handling a large number of projects at the same time with a level of complexity that requires specialized teams for each task.

For the organization under analysis, specialization knowledge is still of high value for the successful development of complex projects but division of testing and development areas by function or attribute of specialty is not sustainable with the forward model of an emerging organization starting to acquire capability and forming an engineering critical mass.

Due the size, complexity, and workload to be managed; merging of the groups performing vehicle level testing and development should be merged in order also to maximize the usage of the dispersed knowledge and expertise present on each group due the increased duplication of roles that have occurred as the organization has been evolving.

As an example of these structures we could mention the structure followed by South America which is at some point ahead on the level of responsibilities and complexity handled within their projects, but still far from reaching a full level of maturity. Focusing on the brakes section, the test engineer in charge of the braking for local products grew by merging attribute development and validation roles as a vehicle dynamic engineer, being responsible for the development and validation. Similar cases among other attributes and function present that only one testing and developing entity is sufficient for balancing the resources within and organization to the work load and complexity demanded.

The current forward model for Ford of Mexico would not be able to sustain two separate groups performing vehicle testing and development activities, and the optimal solution is this concurrent engineering environment. Within this environment the human resources whose work experience and knowledge is related to the development testing for vehicle attributes (Performance, NVH, VD, Fuel Economy, etc), should merge in a single the vehicle attribute development and testing team, in order to take advantage of current experience and resources, maximizing the effectiveness of this group by concentration vehicle level activities.

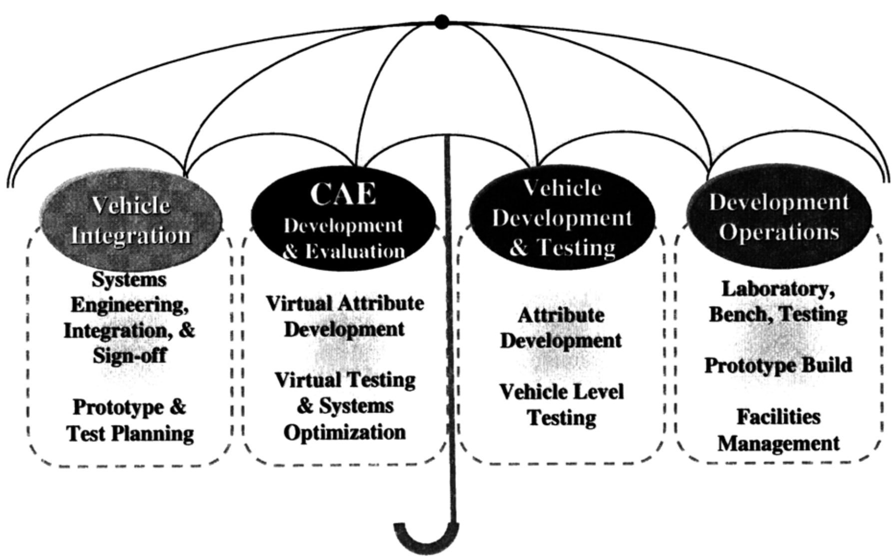

Figure 6-3: Function Decomposition of Proposed D&V Architecture

The function of the D&V remains the same, but for clear understanding of how each element shown aligns and makes use of form elements, a decomposition of function (shown in Figure 6-3) need to be generated.

The function decomposition shows what was mentioned for vehicle level testing capabilities, and as well briefly describes the roles of each group according to the functions related to them. As the strategy for moving road testing to laboratory and virtual environments grows, system and component testing will require special facilities in order to perform testing needing special human resources for the correct and efficient operation of them.

This group which we will call development operations is the one in charge of bench and laboratory testing, while at the same time being the team in charge of the facilities required for their usage.

Following the same concept as for vehicle testing, at least for the targeted level of maturity of the Mexican PDO it is convenient to merge this laboratory and bench testing operations with the rest of the facilities management required for the D&V system functioning which include development and control of facilities such as test track, engineering garages, and laboratories including instrumentation.

A common characteristic for this area is that it involves the function that requires intensive labor resources for its operation (such as laboratory operation and prototype build), making it a valuable area with great opportunities for the emerging PDO due the labor cost advantages that an emerging region offer to more developed ones.

Virtual alternative platforms follow a similar path in terms of handling attribute development and testing for the optimization of systems as will be described in the next chapter, requiring strong interrelation to the previous mentioned testing groups in order to perform accurate modeling creation and correlation.

Last but not least important is the elemental function of performing the integration of functional and attribute activities toward the deliver of a vehicle of product. An activity current constrains require merging with test planning and sign-off of design verification plans, being the entity that open and closes the circle ensuring the robustness of the project.

Planning the test and development activities is as crucial as the correct analysis of requirements, being a driver for a lean product development process.

The roles and responsibilities described within the high level architecture and its delivery of functions is based on the concept of waste elimination and creation of value within a D&V organization. As Murman et al descries the creation of value involves the elements of doing the right job and doing the job right, which for an emerging and growing organization with mixed roles and responsibilities require the restructure of function moving elements of form (human resources) to be allocated to the function where they perform the best toward the creation of value.

6.2. – Development & Validation Process Mapping

As it was said one time by the general Eisenhower, "The plan is nothing, the planning is everything", the diagram of how the D&V system would operate has little value is if the actual processing of it is not clearly mapped.

Before describing the full process followed by a development and validation I would like to make an integration break about the process for developing targets. Integrating Figure 4-3 for the development of system level targets and Figure 4-7 for the creation of attribute development models it is possible to easy derive the framework and process flow for attribute development, a process that can be done up-front for early data acquisition or for prototype vehicle development while done on physical testing.

This is a fundamental process for the accurate development of vehicle level performance, affecting directly the customer perception about the product.

The framework shown in Figure 6-4 combines each one of the system level targets into the performance contribution model where each one of the Xs detected to conform the full vehicle level perception is characterized in order to make them inputs to the model and predict the integrated attribute performance thru the measured transfer function, enabling no just the prediction of vehicle target fulfillment but also the optimization of the system.

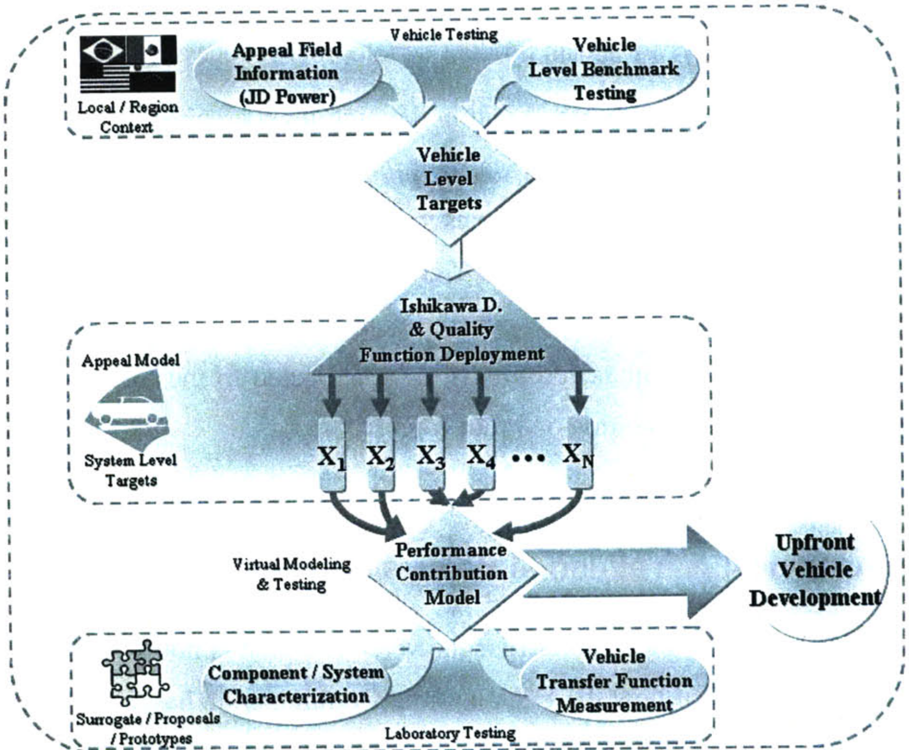

Figure 6-4: Development System Process Framework

The full development and validation process described is complex enough to be described in a single flow diagram due the number of forward dependencies of its tasks, as well as for the feedback loops representing iterations and/or rework included within the process.

The intensive feedback dependencies make a critical path method approach inappropriate since the tool is unable to represent linkages of tasks going backwards to the process flow. The representations of iterations is only achievable by using the Design Structure Matrix[1] tool, which is also the best graphical tool to describe the tasks involved within the development and validation phase of the product development process.

This tool enables the accurate representation of coupled task by either mutual input information, or later information updating (planned iterations), or problem solving activities due issue presented on task that require the repetition of a task or group of (unplanned iterations). (Weck & Haberfellner, 2006)

[1] A Design Structure Matrix (DSM) is a two-dimensional matrix representation of the structural or functional relationships of tasks. The convention used relates task inputs into task in the horizontal axis being upstream and downstream from left and right respectively; and outputs from task on the vertical axis with upstream and downstream from up and down respectively. (Weck & Haberfellner, 2006)

The process was mapped and simplified in 57 tasks as shown in Figure 6-5 and shown in detail at the appendix on Figure A-1. The task based DSM mapped was entered as the input for the MIT DSM program for analysis and partitioning, and although the iterative nature of the development process didn't allowed a significant change thru this initial partitioning method it was possible to identify the main task blocks that serve as sections for the high level view of the process.

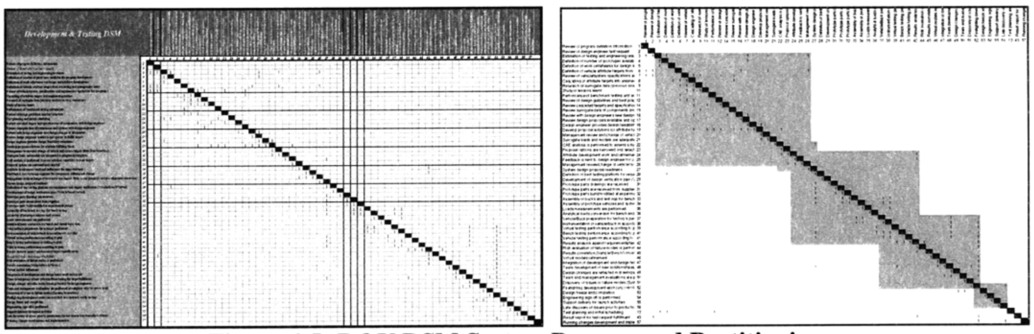

Figure 6-5: D&V DSM System Process and Partitioning

The main loops identified thru this process breaks down the process into five groups, show in Figure 6-6 and in detail at appendix Figure A-1:

- Requirements Analysis and Planning – Which includes the tasks described within the requirements analysis frameworks as well as the activities for resources estimation, scheduling and test planning.

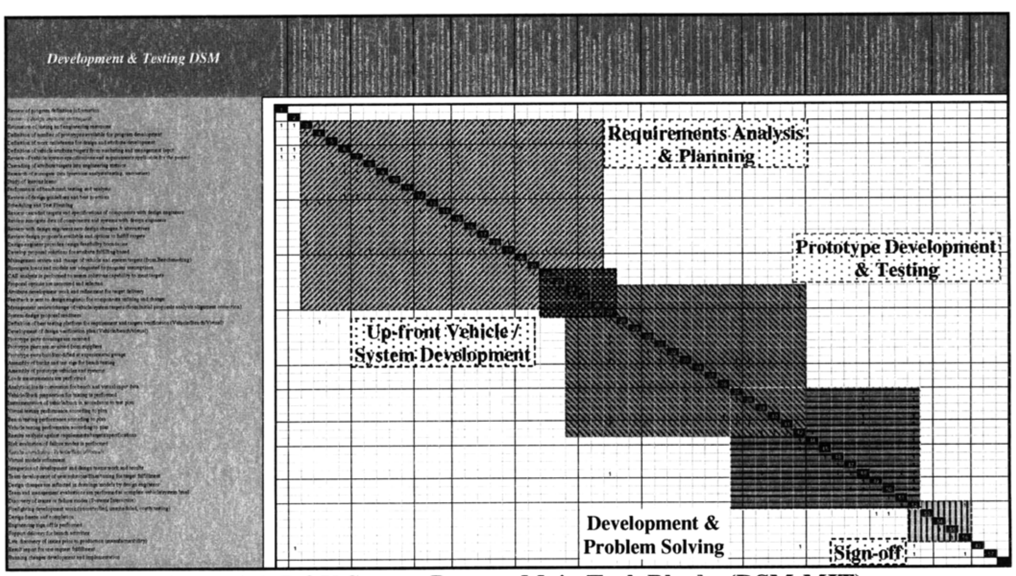

Figure 6-6: D&V System Process Main Task Blocks (DSM-MIT)

- Up-front Vehicle / System Development – Which includes the virtual simulation analysis for the engineering solution proposals and up-front development work.
- Prototype Development & Testing – Including the prototype readiness from planning to build, going through the preparation of prototypes samples to test and verification work over the different testing platforms available.
- Development & Problem Solving – Including again the testing work performed, the correlation of virtual models and development work to correct the gap to requirements and issues presented during testing.
- Sign Off – Being the final phase where design freeze as well as program sign-off is conceived, giving start to the launch phase of the product development system but being aware of possible design issue related to manufacturing and assembly.

Besides than analyzing the task based DSM for consistency and ensuring that the groups identified made sense with the flow and characteristics of each section of the process, a second method was followed in order to confirm that the loops identified were the main loops to describe the system process in its most general path.

The second method used is the identification of loops through powers of the adjacency matrix, which is done by translating the task based D&V DSM to a binary matrix were dependencies are marked as "1" while empty spaces are marked with a "0".

The binary matrix is then raised to the N^{th} power necessary to uncover the loops present on the system. The loops are identified by sequences of non zero values on the diagonal of the matrix

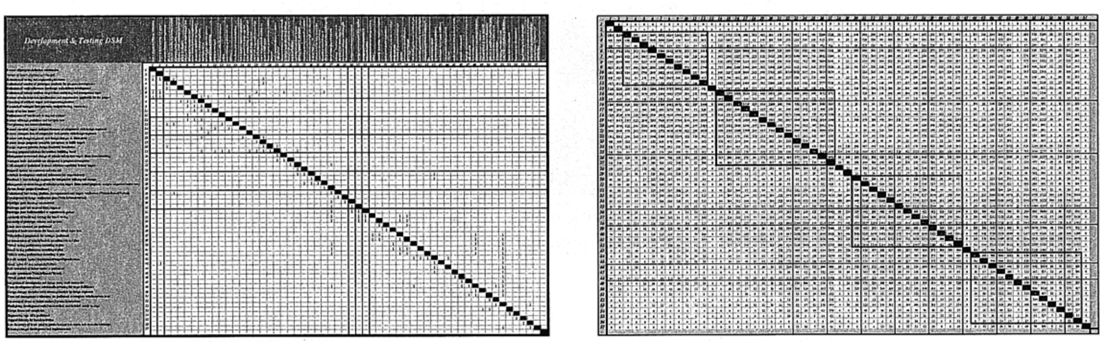

Figure 6-7: Loops Identification by Raising D&V DSM to 10th Power

The result of this analysis is shown in Figure 6-8 and in detail on the appendix on Figure A-2. The task based DSM was raised to the 10th power in order to show a similar result to the DSM-MIT program partitioning method, although through visual inspection it was possible to see that the main iteration loops created mainly due firefighting operations were not completely involved in a loop, and further power raising only groped existing loops into one big loop with our clear separation of the planned and unplanned development loops mapped.

Taking this into account as well as the uncertainty of at what point does the matrix has to be raised, make recommendable the use of the DSM-MIT program (Excel Embedded Macro) for the analysis of task based matrices.

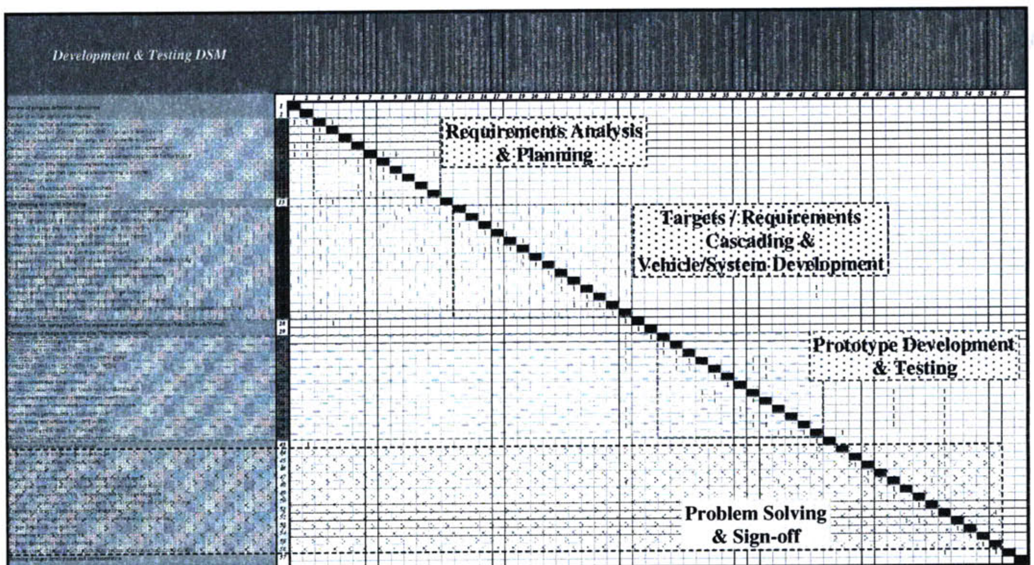

Figure 6-8: Main Loops Identification by Powers of Adjacency

The diagram reveals the importance of having a structure and well organized team covering important processes for achieving a lean process flow searching alternatives to minimize the number of iterations required as well as the resources needed during the completion of the path.

6.3. – Test Planning

Test planning is the immediate task to be performed after the requirements analysis stage of the process has been completed being complementary and of equal importance. It complements the define portion from the validation phase of any product development system, and being correctly applied has the capability of defining lean principles over the entire development and validation process, having the capability of reducing phase completion times and reducing costs both in terms of testing but as well in prototypes.

The ultimate goal of test planning is to deliver the elements for the creation of efficient and effective design verification plans under a systems integration approach for resources sharing among all vehicle functions and attributes.

6.3.1.– What Tests to Perform and how?

Design verification plans must be requirements based documents able to relate the needs for the system to fulfill with the methodology to achieve the verification.

Determining what tests to be done and it those really represent the teal requirements and specifications to validate comes from a top to down process, where cascading of targets and requirements is essential in order to integrate properly a full vehicle of system performing as it is defined to. The fulfillment of a plan which is the performing of the methods is on the contrary a bottom to top process where verification of component to vehicle level system is performed. In a brief summary, correct test planning could be defined as the process covering the following cycle (Grady, 2007)

- Verification of requirements analysis completion and cascading
- Completion of verification traceability matrices for accurate control of items to be tested and entities responsible for it
- Definition of test objectives and flow diagram preparation (scheduling & sharing), and resources preparation
- Test procedures definition
- Documentation systems readiness assurance.

An important point to consider while elaborating testing plans relies in the importance of test procedures. Test procedures choices must not only consider what is needed in that particular phase, but also capture important issues of representativeness for the entire project. (Wheelwright & Clark, 1992) Testing strategies must consider the whole development duration of the project and not be based on individual pin point activities. At the same time testing must be conducted by teams experienced enough able to reproduce real customer usage and ensure representativeness of testing methods.

The usage of generic DVPs as a start up platform for the planning process is encouraged since it avoids the usage of previous programs that not ensure that all the right considerations are taken for the new program and promotes over engineering by requirements that need not to be used.

The usage of generic DVPs still demand high analysis and verification of current level of requirements to be applied to a platform, hat or derivative.

Function or attribute based generic DVP give light to engineer of which requirement must be considered for the planning of new programs ensuring that basic considerations based on previous knowledge are taken into account, channeling the focus to the refinement of that database against specific characteristics of the program.

6.3.2.– Test Planning for Reduced Testing and Prototype Expense

Test planning as both an integral and upfront activity for the development and validation testing process have the capability of greatly reducing costs and maximizing the efficacy of the resources spent by performing the two following functions:

- Act as an integration intermediary by allocating usage periods for the prototypes available within the engineering and development teams, making time scheduling based on prototype vehicle share over functions and attributes requiring a certain prototype characteristics and that follow a process that maximizes the use of the vehicle or system from its creation to its final destruction.

  The sharing concept is simple but requires a good general awareness of the needs and functions of each engineering group in order to relate them. Taking for example the construction of a prototype vehicle which first will serve as an integration and packaging tool, sent later for non destructive vehicle level attribute of function verification such as NVH, performance, or calibration development, and finally disposing the structurally undamaged prototype by performing safety testing as the crash barrier.

- Prototype specifications delivery, in order just to build the minimum number of prototypes allowed by the budget from a program or project, and maximizing the number of prototypes available by just building toward the specification needed by the engineering group that will be using it. Taking as example prototypes dedicated to body testing just requiring a body on white[1] version of the vehicle, or functional areas not requiring a vehicle with a

[1] Body on White of BIW means a prototype where just the sheet metal structure is provided, not having any powertrain, electric, or interior trim component.

functional powertrain or even having one. This manages the concept of partial prototypes delivering what is needed for the functions to verify.

6.3.3.– Smart Testing

"To a person who only has a hammer every problem looks rather like a nail"

– Traditional English Proverb

The problem solving rush is an element that often overrides planning activities with the intention channeling the resources directly toward the solution of an emergency. This initial intention to save time toward the resolution of an issue most of the time ends on higher delays and resource expenditure.

Smart testing is nothing new and foreign to the D&V process, but rather just the same process followed with patience and analyzing the needs for every process and the outcomes the will emerge from it.

Smart testing start by planning a DVP with smart requirements, which are the one derived from appeal models and correlated to deliver the exact function desired for a system or component by being cascades from vehicle level to component level.

The basic inviolable principle is never to perform a test that is not well understood, by either even one of the following points:

- No understanding or agreement on the boundaries and context to be tested (adequate level cascading)
- No understanding or agreement on the reason why the test is being carried out (requirement)
- No understanding or agreement of what is intended to achieve by performing testing, and the limits of the assessment (goals)
- No understanding or agreement of how goals are going to be achieved (method)
- No understanding of agreement on the relationship between results and the requirement to verify (analysis)

Non-smart testing is always driven by the fear of delays, delay of not delivering an assessment on time and fear of not solving an issue a quickly as possible. The need of information, data, or values; to perform a decision is always urgent but it is important as well to dedicate enough time and resources so that the information, data, or values; really are an accurate representation of the system and its behavior.

No engineering entity does non-smart testing on purpose, but it rather is a result of poorly and not well understood planning processes, with the noble intention of acting as quick as possible. An analogy could be made between the engineering method and old chivalry tales, where the Ingenious Gentleman "Don Quixote de la Mancha" attacks with the most righteous of intentions and the noblest of purposes, but without thinking a proper plan or analysis, to the very first target his spear can point.

"Delay always breeds danger."

– Miguel de Cervantes (Don Quixote, Book IV, Chap. II)

Don Quixote could never distinguish a real enemy from a windmill, but the people around him could... therefore the importance of never disregard the opinion of third parties with fresh eyes over the process to be followed to attend an emergency.

It is mandatory not to perform any testing work that it is not well understood or when the targets/goal could not be well comprehended. The best D&V section, the best engineering and managerial organization, could charge in the quest for what they believe is best for the product, the customer, and the company; without even thinking that they may lead into a windmill and awaken the flames of firefighting.

"Never do today what you can put off till tomorrow.
Delay may give clearer light as to what is best to be done."

– Aaron Burr

6.4. – Prototyping

Prototype are (physical or virtual) are integrating tools and a common reference point for the teams communication.(Sobek, D.K., 1997)

For any product development system, prototypes are element that represents milestones, bridges and agreement sets for integration among attributes and functions, and tools for the adequate channeling of resources by process and performance assessments, and problem solving enablers. Prototypes provide PDO with effective tools to understand and correctly direct targets and requirements fulfilling.

Baldwin and Clark suggest that modularity is manifested through design, manufacturing, and use. A new manifestation or just the use adaptation may be described for a D&V system as modularity in testing which permits only to perform testing over modules instead of building full prototype vehicles, although it is very important to describe that modularity doesn't eliminate the need of full vehicle/system prototype evaluation in order to verify the correct system interactions and performance of the product as a whole. Modularity is a driver for reducing costs at bottom testing or component/system/function based levels.

As integration tools, first prototypes must focus on the integration of systems and the availability of options for further development. In this first integration step the verification that various alternatives are able to fit and integrate correctly with the systems will provide concurrent vehicle variants that in later development process may be discarded but within problem solving iterations provide built ready possible solutions that reduce time for correction, but require high design modularity in order to have several options without requiring strong changes to the surrounding systems. (Sobek, 1997)

6.4.1.– Prototype Reduction Strategies

It is desirable to have large number of prototypes built at many stages within the product development process in order to ensure that the design is evolving toward the desired performance, but this approach is costly driving the industry toward the reduction of prototypes stages and numbers to minimum possible in order to achieve competitive development costs. (Wheelwright & Clark, 1992)

Besides and accurate and effective test planning strategy, based on sharing and uptime maximization of prototypes. Several other strategies are recommended to be followed for expenditure reduction.

- Build labor costs – Prototype building are labor intensive tasks that encourage the usage of the lower labor rates available, a great advantage and opportunity for small and emerging development organizations.
- Alternative testing platforms – As described by most literature, the usage of physical prototyping and testing must not disappear but rather be greatly reduced by virtual or bench platforms that enable early prototyping and testing for increased robustness and product development speed. This alternative enables the reduction of physical build phases, increasing the representativeness at each one of them.

- Prototype tooling reduction – This strategy involves the usage of rapid prototyping for development of low volume prototype parts, expanding the concept as well to develop rapid prototyping for soft[1] tooling, avoiding the expenditure for soft tooling and pulling ahead hard tooling design which can be used for final vehicle prototype assembly for tools try outs in assembly plants.

6.4.2.– Prototyping Constrain for the Small and Emerging PDO

In emerging markets, product development organization must wisely allocate their budgets and make the most of the prototypes available for vehicle and system development.

While most well developed PDOs focus their efforts over the path of reducing their prototype expense by reducing their numbers, emerging organization battle against the constrain of the number of prototypes they can afford always reaching alternatives to have more or make a more effective use of them.

Usually for the small organization there is no constrain about expending to much on many prototypes, but rather facing the challenge to allocate the budget and have a few prototypes against none.

While labor costs for prototype building is an advantage, the build and logistics for prototype parts must be the focus by using local suppliers force and manage the construction of local developed build of parts and assembly of prototypes. Scale will be a problem that will need to be managed by the scope of the projects to handle.

This leaves as strategy for the emerging organization a very well developed test plan for maximizing the up-time and sharing of prototypes, as well as promoting the development of alternative testing platforms.

6.4.3.– Overlapping and Value Capture

Prototype stages must not overlap, since people lose track of the status of the project, which problems have been solved, and which should be given top priority in the current cycle. Ensuring that testing is completed following the production of prototypes and getting closure on each learning cycle is an important practice.

[1] Soft tooling is considered as a non-definitive tooling made for prototype development and low volume applications.

While not overlapping but establishing direct links with other phases of the product development process such as manufacturing, is a key element for right usage of prototypes and maximization of value from the resources expended on them. As Daniel Whitney says in Manufacturing by design, in (Harvard Busyness Review, 1995), most companies have operated for years in an environment where design and manufacturing communicate infrequently, if at all. In the worst cases product design is just thrown "over the wall" where the design engineer disappears after sign-off and manufacturing engineers struggles to build the products.

Use prototypes to capture and enhance knowledge, in order to continuously improve the quality, speed, and efficiency of the whole process, strengthening representativeness, and the ability to capture critical problems at each stage of development. (Wheelwright & Clark, 1992)

6.5. Documentation & Sign-off

After requirements and targets have been met a sign-off process is followed completing the design and development phases within the product development process, where design is frozen giving start to launch stages for any program or project.

The completion of the development and validation phase requires a complete set of verification reporting documentation, an item strongly encouraged where the development, technology used, or lessons learned are highly unprecedented (Grady, 2007)

A good system to store testing results and/or compliance of the systems toward it requirements and targets, is recommended but as well as the lesson learned and solutions achieved in order to fulfill them.

A final recommendation on documentation systems is not to over document details within the problem solving activities, reporting must be only done when the data reported doesn't compromises the organization thru liability issues which must be managed separately documenting what is safe as lesson learned and best practices that ensure the non repetition of issues observed without directly mentioning any program or project that present it. (Hotta, 2008)

6.6. – Could Testing Be a Value Adding Process?

Although testing is often seen as a necessary waste within many product development organizations, its importance is vital since it doesn't only serves the purpose of validating a design, but as the mean to achieve the design itself. Testing is the key driver in a product development process for cleaning and tuning a design, as well as the source of innovation for problem solutions that the paper is no longer able to see. Testing offers a unique learning path, which can lead to systemic points of view that offers overall knowledge from the entire vehicle or system, and the interactions among every system and component.

But although testing is a source of learning and provides a product development team with the path for finding and solving problems, as well as for innovating, is testing a process in which the final customer sees value and pays for it? What is needed for a customer to pay for testing?

Waste is a very famous word within lean thinking as it focuses on eliminating it. But waste can not always be eliminated and there relies the element to form waste categories: the necessary and un-necessary waste.

- Un-necessary waste is the process or element that detriments the value of the product, its performance doesn't add any value or benefit in the chain, and could be eliminated from the whole system.
- Necessary waste is also a process or element that doesn't add value to the product of process chain, but is required by the process as a system in order to avoid value detrimental or is an enabler necessary for the process to continue.

We test in order to verify and validate, that we have a winner product, as the ultimate goal of doing the things right the first time by achieving perfect execution. But if we have perfect execution and if we do the things right the first time, what if we just don't test it? We test because we can definitively do things wrong, and if by any case we didn't do the things right the first time, we want to correct it before the customer notices it.

All the ifs mentioned in the previous paragraph are elements that give uncertainty to the product, uncertainty that can be created even by the best product development organization and reduces the value of the product. Testing is the process action that takes the "ifs" away and therefore performs as value adding.

We can conclude that even on a product development organization so capable to deliver product designs that are done right the first time, the uncertainty lessens the value of the product as there is always an "if" that could put into risk the whole quality attribute for which the product wants to be identified. This uncertainty is the un-necessary waste.

Testing which often is seen as a required waste came to be as the tool or path for eliminating uncertainty, adding value to the whole product development process chain.

The etiquette of value adding process to required waste is only defined by the evolution of the customer expectations through time. This could be explained by adding the time dimension to the well known Kano diagram:

- Testing could be seen as a required waste, something basic that has to be there in order to avoid customer in-satisfaction. This one of the most commons view that our time has since today the customer expects that the products are tested, they expect them to be reliable. As an example we have the pharmaceutical industry products, since the customer expects them to be exhaustively tested to even been able to be considered saleable.

- Testing could be seen as a performance attribute that ensures an increased reliability of the product as more testing is performed, with the scope of a continuous and often never ending product improving and tuning. The more the better it is said, but we have to identify when something is good enough. As an example we have many of the 6 Sigma teaching of increased measurements in order to assure that there variance is within a certain range, or by the reliability standards within the military industries. A military product company is test more and more in order to ensure that every unit will perform as specified with a low variability.

- Testing can be seen as an excitement attribute. The automotive industry is a clear example since the data of a successful crash tests is actually data that is sold. Although engineering delivers a vehicle that is a top safety pick due to is crashworthiness characteristics, the test data is actually the element that brings the customer excitement. This data signals the product from among the rest in something considered significant and valuable for the customer.

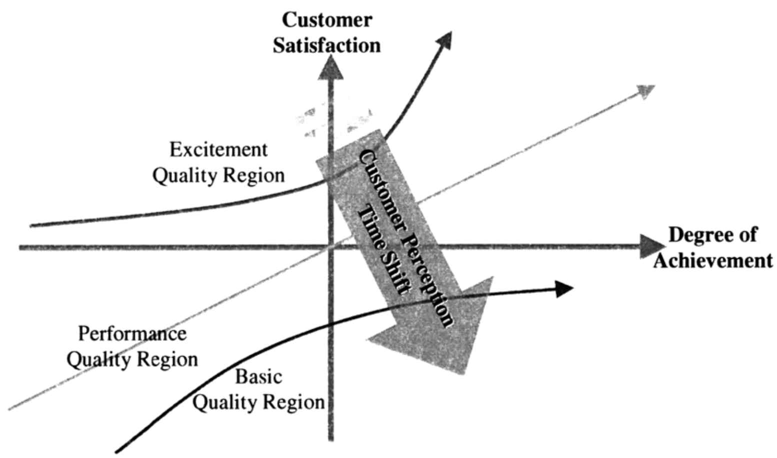

Figure 6-9: Customer Value Perception over Time

As time flows what caused excitement in the customer, this attribute becomes a performance measure, and with more time it moves to a required/basic attribute. Creativity and strategy are the key parameters for shifting time back, and produce refreshed images of testing that pull it back to provoking excitement against the customer expectations.

6.6.1.– Testing as a product attribute

In order for testing to be an adding value process, it has to be part of the product, part of the reason by which a customer is paying.

Testing gives the confidence of driving a safe vehicle, people don't expect to crash in order to feel the safe attributes of their vehicle. Testing is the indirect way in which get the feeling of the qualities of the vehicle they drive.

Bigger than a guarantee, testing is the attribute of confidence in terms of reliability, which is not referred to any particular system to be designed or developed but as the integrating activity and verification that everything will perform at the customer expects with the combined performance of vehicle attributes and system functions.

In most of the cases, people doesn't buy a vehicle by analyzing data but rather on the perception of it which is given by the performance of the system across its different attributes as well as by reliability references and information available for a certain model or brand.

The reliability confidence as an attribute is able to be perceived by the customer after a model or brand have presented it over time becoming a characteristic of the product, but this perception can be transmitted also to customer by making it aware of the testing that the product endures and the technology used to ensure its reliability.

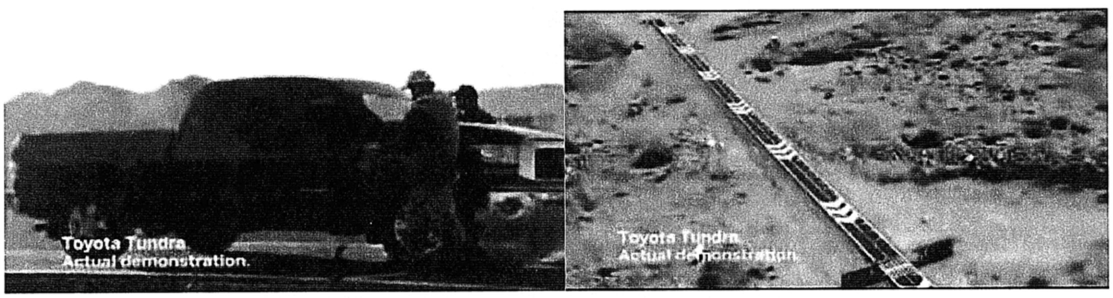

Figure 6-10: Toyota Actual Vehicle Demonstration TV Commercial (www.youtube.com)

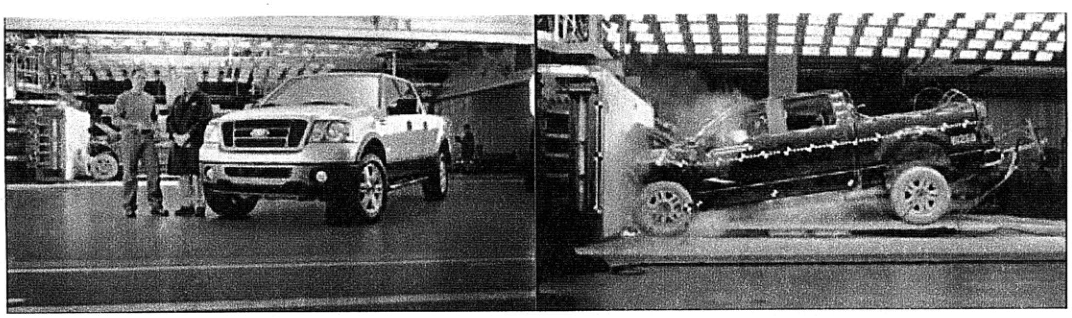

Figure 6-11: Ford Actual Testing TV Commercial (www.youtube.com)

This strategy has been started by some organizations as shown in Figure 6-10 and Figure 6-11, and could be applied with greater success in emerging or secondary markets by giving perception to the customer of the complex process followed to adapt a global portfolio to the very local needs and the testing work to ensure that the product is verified to perform under the environment where it will be used.

6.7. – Firefighting Management

Current global economic environments impose severe constrains on the budget available for the development of product both in time and resources. This leads to the planning and scheduling under a strategy that we could name as "The Home Run" strategy.

The need to develop and deliver a product within current automotive industry settings, drives development teams to work under the "Home Run" strategy in which timing has no space for failure.

The "Home Run" strategy is particularly seen at the development and testing phases of the product development process, where designed products are always expected to give positive results with the first unit tested.

There is no allocated budget for redesign in order to offer the most competitive scenario, where any unexpected issue is dealt as an emergency pulling funding from other projects or the very same profits of them. (Kleyner & Sandborn, 2007) In the best of scenarios, development and changes required to perform due failures during validation testing are considered as minor events which timing can be regained thru compressing posterior steps within the product development process having no or very little changes in final delivery dates.

Adding extraordinary resources to meet deadlines at the very end is also the most common scenario of fire fighting, which is a self-reinforcing and self-sustaining phenomenon that spread over neighboring projects when resources allocation is inadequate. (Repenning, 2001)

The cascading effect is described by Wheelwright and Clark describing that if anyone project runs into unexpected trouble and there is no slack available in terms of time and/or resources, it will be necessary to take resources from other projects cascading the firefighting issue from one project to the other. (Wheelwright and Clark, 1992) The issue effect is incremented by the addition of resources that require time to learn the project particularities and be able to perform in alignment with the project context and inner characteristics.

Although the best theoretical choice would be to estimate a percentage of resources and time within the development of a program to the solution of issues, as it was mentioned before it would make the planning of a project loose competitiveness.

Rather than that the PDO should take a convergence development process such as it is followed by Toyota where instead of focusing efforts to the development of a single solution for the delivery of function, a portfolio of some options that works with enough difference for not to depend on the very same constrains is a path for indirectly reducing test and development resources since the number of loops required to the final solution of issues toward attribute of function delivery can be greatly reduced. Instead of running the full development loop for one solution and starting it for other, this approach eliminate weaker solution in one loop leaving the stronger one for complete its development in fewer iterations.

Having different solutions with non dependency among them is also an enable to sustain requirements changes since each option carried over the development path could provide difference in their function with individual tuning or survive changes in constrains that eliminate the weaker ones not able to be adapted efficiently.

It is to expect or desire that requirement are analyzed, selected and frozen; but PDO must live under the reality that requirements are elements that change or evolve as quickly as the market evolves or new points of view/constrains are discovered. There must be a planned point for requirements freeze as there is one for design changes freeze, but the PDO must have enough options at least on the back in order to have a certain degree or readiness against changes.

Another dangerous option is the increase of milestone reporting and performance demand from management, which acts as a surveillance activity for the resources and performance of the team in a quest to contain any delay flag that could lead to a posterior firefighting condition. This strategy is dangerous since it leads to collective irrationality when the expectation of the manager is not able to be met and the perception of the team is shifted to hard working to lazy and non-competent.

Collective irrationality is also generate since engineering teams tend to minimize scheduling or resources issues with the intention of not having a project cancelled and loose their expectations, becoming a driver for firefighting. On the other hand a stricter management control increases bureaucracy and the resources are oriented to paperwork instead of development, creating again a firefighting driver. (Repenning & Black)

Summarizing, and leaving clear note that there is no magic bullet for firefighting eradication, two strategies are recommended to be followed in order to minimize the effects:

- Concurrent or parallel development of several options with enough differentiation to contain changes in requirements or issues detected during testing.
- Increase on managerial surveillance over the performance of projects being supported by expert planning teams for the correct and immediate allocation of resources with high learning curves for easy adaptation into projects.

6.8. – Summary and Conclusions

Testing in emerging and growing organization need a robust architecture where resources are optimized for quick and effective solution to development and validation iterative cycles toward the creation of a robust and competitive product in the time that is needed.

Young testing and development organization, in emerging product development entities are not able to support several groups performing vehicle testing, due the level of complexity and workload which can be better managed by a single program focused group with specialization training but in a concurrent working environment.

Test and development teams should be program oriented teams, carriers of general knowledge and focused to the customer perception, creating communication links among functions and being supported by the knowledge in depth from functional engineering entities. (Harvard Business Review, 1995)

The structure required needs a concurrent systems thinking approach, where cross functional teams organized around a programs and not a functional groups, where support from engineering functional providing knowledge and tools for quick development program actions under a general development framework based on the steps widely mentioned by Wheelwright, Clark, and Fujimoto; for effective problem solving:

- Adequate definition and framing of problems

- Generation of alternatives among parallel development and lessons learned systems
- Lean verification of alternatives (planning, methods and goals clearly defined, understood, and agreed)

Development teams require to develop within this structure as a heavy weight team as stated by Wheelwright Clark, since integration of systems and bridging among engineering teams requires a strong voice for leadership. In the same way, development teams require this heavyweight structure for the creation of ownership and commitment, creating a stronger identification with the project, and fomenting improved communication and reduced time to find solutions.

As mentioned in the requirements chapter, problem solving is more dependent on lessons learn than in actual knowledge, making the adequate documentation a basic step toward efficiency in development cycles.

Prototyping must be an integrative tools that not only joins systems and components to verify a design, but rather integrate a whole organization into the creation of a product by integrating the effort from engineering, manufacturing, purchasing, management; where several alternatives are ready available in parallel non detailed development, in order to accelerate and reduce the time within development and problem solving cycles. (Clark & Fujimoto, 1991)

"We shape our buildings; thereafter, our buildings shape us"
– Winston Churchill

Chapter 7

=== From Road to the Lab and Beyond ===

As far as the laws of mathematics refer to reality, they are not certain;
and as far as they are certain, they do not refer to reality.
– Albert Einstein

7. – From Road to Laboratory and Beyond

In our current times the companies that are able to deliver better products and reduce the time it take to release it to the market a competitive advantage that can define not only which companies will succeed but also which one will be able to remain on the market.

The pressure existing on the automotive industry due its current crisis requires more than always the reduction on resources, spending only on what is most necessary. This affects directly the testing activities, which systems are under great stress in order to reduce its expenditure, but unfortunately physical testing is a huge resource consuming activity where the options for reduction are not what the current times demand.

This is the point where new methods and technologies are created and explored in order to produce testing alternatives that enable PDOs to validate requirements with confidence at a reduced amount of resources.

The purpose of this chapter is to describe the methodology to follow by an emerging PDO for correctly adopting new testing methods for the long term reduction of the resources spent in road and vehicle testing levels.

An important objective within the methodology is to establish the tools for achieving a clear understanding of the architecture and environment of testing systems, for translation in to models for bench and virtual testing. The analysis is intended to determine the tools needed for accurate system test modeling, understanding its limits and the data required for accurate modeling. The goal will be to serve as the starting point in the process of taking tests currently done with physical prototypes, into a robust virtual environment for reduction in test time and resources with minimal loss in representativeness.

Virtual testing has not the intention of avoiding the need of a physical tests or prototypes, but to achieve well directed prototypes and tests with high level of representativeness and design confidence through less prototype phases and builds.

7.1. – Importance of Road to Lab to CAE Migration in PD Systems

As mentioned by Wheelwright and Clark, the development on new technologies that help to reduce the time and resources spent in the creation of new products has become the competitive advantage that sets the difference between the leader companies in the industry and the rest.

As seen in not a long time ago in the test and development structure of regions such as Mexico and Andina, the reliance of testing methods based on full vehicle evaluation was the iron horse of their testing capabilities, leaving the cases where a specific component needed to be tested alone to their suppliers network.

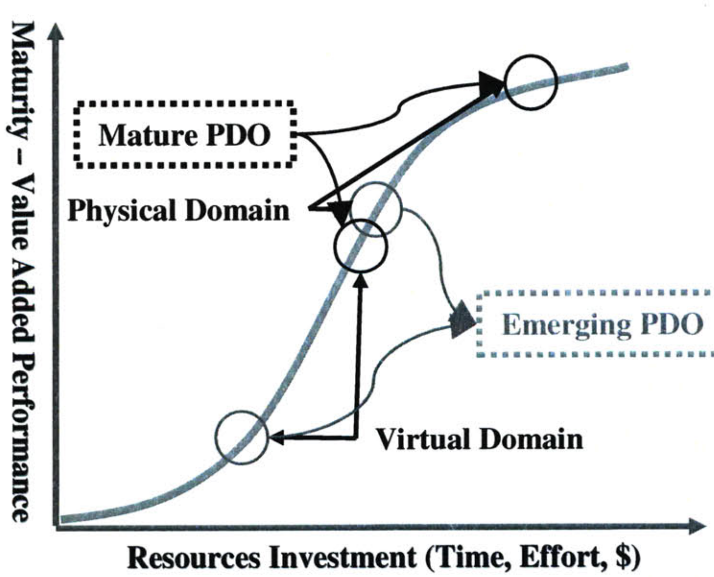

Figure 7-1: Technology S Curve between Testing Methods Domains

The issue of this testing structure is that the degree of opportunities in order to make it more effective and efficient is limited since the maturity of the domain has reached a point where only incremental innovation make the domain fit to become lean and be better planned but offer few opportunities for a radical change.

A milestone that serves to measure the level of maturity within a PDO resides in the level of adoption for new testing platforms, and the maturity degree of development of these tools over the life cycles of the testing platforms as shown in Figure 7-1.

Vehicle and road testing are testing platforms that have evolved during a large period of time making the improvement opportunities unlikely to radical innovations, as seen in the black circles. These circles shows that for mature organization there are minor changes available for improving the performance of the platform while for an emerging and growing organization there are improvement opportunities which mainly rely on adopting lean strategies for maximizing the efficiency of the testing activities as well as sophistication of measuring equipment.

On the other hand exploiting the field of alternative testing platforms such as bench testing and CAE offers a wider range of radical innovations where different use of engineering principles enhance the potential on the applications to reduce time and resources expend on testing. (Davies, 2007)

Shown in gray circles in Figure 7-1, the bench and virtual testing platforms offer greater performance increase opportunities especially for mature organizations. Emerging and young organizations see to be late adopter of this technology due the high investment required to set up the infrastructure which as shown in the figure will represent a stage of low improvement per a high investment; being the differentiating factor among PDOs with different ranges of maturity and capability.

7.1.1.– Adoption Differences between Organization Maturity Types

Emerging growing organization have the disadvantage of running against the resource and expertise boundary, which leads them to scenarios where test and development work is done with the tools found at hand. This is commonly seen in the reliance on subjective metrics for the valorization of attributes and system/component behaviors difficult to measure or isolate.

But the industry context demand action from every player disregarding their capability or maturity and is the point where adoptions of CAE tools are done within the development process of PDO. There are key differences in the path that is used for technology assumptions which can be described as follows:

Type of PDO	Emerging / Developing Organization	Mature Organization
Adoption Characteristic	Pull	Push
Behaviors	Driven by demand and opportunity, being a late adopter of proven and established tools.	Driven by knowledge and innovation, organization supply a new method to the rest.
	Fear and anxiety on uncertainty of potential benefits (in the emerging PDO context) against tools already been used successfully	Underestimation of switching costs and overestimation of potential benefits.

Figure 7-2: CAE/Bench Testing Adoption Behaviors (Davies, 2007)

7.2. – Drivers for Road to Lab and CAE Migration and Maturity

The main driver for any product development organization that intend to migrate testing work from the road to controlled environment in laboratories or CAE models is to be a data driven engineering organization.

A phenomenon strongly present in test and development organizations is the wide usage of subjective evaluations when the vehicle or system behavior is complicated to measure or even not achievable to due the equipment, facilities, or expertise constrains of the group.

Emerging and young organization have strong reliance on vehicle evaluation rating for assigning a value to the performance of a vehicle or system attribute such as NVH or vehicle dynamics, which results are not quantifiable and therefore difficult for target setting and cascading into laboratory or virtual testing.

Although attribute requirements need to be derived from appeal models that translate the user perception into engineering units, migration from road and vehicle testing need to part from engineering units and not from opinion or comments driven processes.

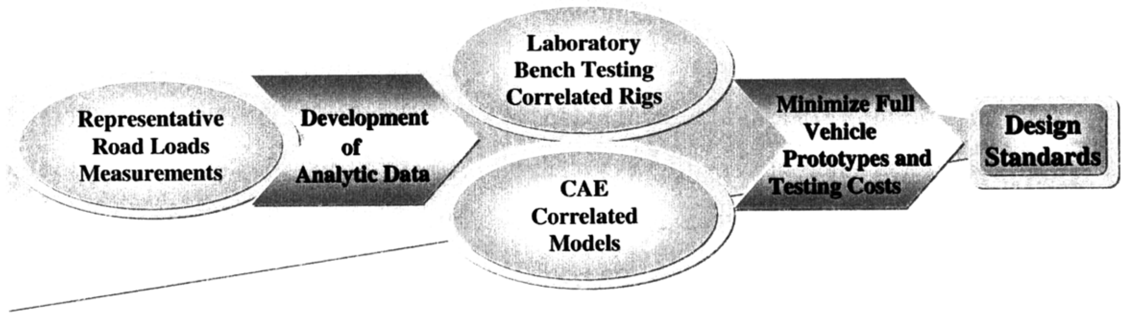

Figure 7-3: Virtual Testing Platforms Maturity Strategy

Taking into account the scenario of an emergent test and development organization with the lessons learned and needs from a developed one, the main strategy to follow in any organization to reach maturity on alternative testing platforms is described in Figure 7-3 and summarized as follows:

1. Acquire capability to measure representative road loads by either internal resources or external, and translate them into usable analytic data.
2. Process analytic load profiles as input for CAE models or test rigs where feasible.
3. Correlate the loads and models with results seen on actual vehicles being tested.
4. Rely more on correlated models, reducing test time and samples of vehicles road testing.

There are other elements that are needed during the migration path that have to be taken into special account for organizations that plan the adoption or growth of this tool for product development enhancement.

7.2.1.– Human Resources with Proper Background and Skills

In order to build and grow a CAE organization into a mature state, it is required to start it with the proper engineers with enough background and skill that are needed to build models that represent the real world as accurate as possible.

The CAE organization needs to be built with engineers with a deep understanding of models and physics or a background with experience on them in order to avoid building a critical mass of software operators. Engineers with master or PhD degree are highly desirable although it is one of the greatest challenges while building a CAE critical mass. Especially in emerging regions the availability of engineers with the required academic background or experience is very limited, making difficult to recruit the needed people due marketplace availability requiring instead a high investment on intensive training.

It is not advisable to start a CAE organization with any available engineers who can be trained as software users, since there will be that capacity gap for the creation of model which is mandatory for the creation of whole new models and not just derivatives. (Megalhaes, 2008)

7.2.2.– Model Correlation through Physical Testing

Modeling is based on a big number of hypothesis and assumptions being a difficult task when taking into account factors like manufacturing variation and calibration of vehicles for example, requiring complex crosscheck in order to assure the confidence on the elements over which an analysis shall be made.

CAE teams must be in direct contact with the physical testing teams in order to accurately process and refine models. In the same way it is desirable that the assumptions for CAE are performed within a same team, which contains both physical and virtual test personnel together sharing knowledge and lessons learned about attributes and functions. Correlation work between physical and virtual testing is fundamental for acquiring maturity, expertise in the handling of tool packs and software is not reliable unless its results have direct correlation with the physical world. (Megalhaes, 2008)

In the same way analysis results should be analyzed from both the physical and virtual side with the intention of knowledge replication about failure modes analysis. (Megalhaes & Lopez, 2008)

Still possible but with large lead times is the field performance of developments based on alternative testing platforms, since manufacturers use warranty data to compare actual field performance with pseudo-lifetime data generated in laboratory or bench test settings (Majeske, 2003)

7.2.3.– Road Loads Acquisition System Capability

Modeling is an intensive task that requires deep understanding of physics and behaviors, but in order for every model to operate it needs inputs of the same quality as the attributes that are required to build it.

The loads or inputs that are to be applied to a model need to be representative from the model they will simulate and in the same way to be measured with the quality and resolution that the model is capable to handle.

For long time the group results analysis capability has be dampened by the lack of an adequate system providing road loads or system loads for analysis, relying on surrogate loads from vehicle with certain degrees of difference and load not closely representative to the loads present on the market where the vehicle or product is going to be sold.

Although investment is considerable (close to the million of dollars) and the expertise to measure and handle analytical loads could be intensive, the ability of an organization to measure the inputs that the vehicles experiment while being used is an important driver for giving confidence to the results derived from simulation an provides the capability for taking design decisions from CAE and bench test models.

Correlation needs the developing methods in order to define procedures, work which is initiating but needs further time in order to fully mature its results. Therefore analytical road loads expertise is required to the further development of models.

7.3. – CAE Role in a Growing Product Development Organization

Organizations under development with current level of organization maturity that could not be considered of high confidence for decision taking, leaves little time to the development and growth of forecasted capabilities since all responsibilities are directed to the attendance of current engineering priorities.

There is a certain level of maturity at some attributes (e.g. Durability, Safety, Body Structures), with medium level of maturity in others (e.g. Thermal), but none with the appropriate level of confidence and many more with development opportunities (NVH, Vehicle Dynamics) which are developed and used on a need basis for the programs being developed by the organization. At least today, organizations in the path of development (Mexico, Andina, and South America) are not able to use the tool for sign-off purposes but as a risk redactor agent for development and validation purposes.

General knowledge is the advantage, since the use of general tools enable the construction of a critical mass where specialist are going to be developed in accordance to the most urgent needs of the PDO in conjunction with time and experience. The CAE engineer in an emerging organization needs to have the general basis of performing analysis in order to achieve a flexible capability, but at the same time start to develop experience in two attributes for future core development. (Lopez, 2008)

The purpose of CAE is to provide support to the design and release team on the design proposals for a system. Based on requirements the virtual analysis is run to predict the behavior on the system, changes on the proposal are submitted based on the results in an iterative process to increase the probability of the system to successfully pass it validation tests. At this point of growth and maturity of the PDO, CAE has no direct authority over designs although it is an attribute that increases as the engineering group develops it capabilities and expertise as it serves to improve design reliability and robustness.

For these organizations CAE works better as a tool for optimization and evaluation of changes predicting performance, as well as for root cause identification in problem solving tasks.

Organizations adopting new testing platforms must focus on the adoption of the current level of mature testing platforms that best fits their development needs, searching to broad their applicability and building up complementary assets. (Davies, 2007)

Even in PD organizations at mid levels of growth, there common view of CAE methods within the attribute development community is that CAE is a tool that you use if it works or not if it is not useful. There is a point of view covered by doubt where even if CAE analysis correlates with physical testing, the results may not take great more than a directional importance since the physical perception may differ from the numeric results.

Within this view CAE is not considered in an equal role since the development teams are considered to be the program owners, having CAE or core developers as entities that report analysis results to them.

This view need adjusting roles and responsibilities with a scope where both physical and virtual teams work at hand, and where each team has a generic knowledge of the other realm. Physical teams must know a genera approach to CAE in order to better understand models and its results, and virtual teams must know the physical environment in order to better know the product and how to model it.

7.3.1.– Availability of Larger Corporate Instances

Being the branch of a larger and better developed engineering organization, which could be the case of offices in emerging markets from transnational companies (Brazil, Mexico, etc), has advantages for speeding up the deployment of CAE tools and methods for the regional PDP but at the same time flexibility restrictions.

The advantages reside in the capability of following corporate procedures and requirements. A focus of following those procedures need to be strict in order to avoid the need of generation of new modeling methods, boundary conditions, and loads application. In the same way the availability of the author of the procedures is an academic resource for the right deployment of the studies. (Megalhaes, 2008)

Advantages or disadvantages from the point of view chosen is that the previous approach requires usage of the same software and licenses used in the corporate office which often requires higher expenditure in software not commonly used in the region and training in foreign locations.

7.4. – Wrong Understanding of CAE in a Growing Organization

The responsibilities of the CAE groups toward the Product Development community are undeniable but it is still required to reach knowledge more in depth in order to fulfill the capability assumptions that are demanded. This process is not clearly understood and in this path for gaining expertise and confided generates a lack of trust in the tool, and too much pressure is applied on the tool due the expectations demanded on the tool.

CAE is not a tool capable of predicting the future but rather explaining the phenomena as it is being modeled. Organization need to dedicate resource on intensive systems engineering work in order to achieve models that could accurately represent what the designers try to achieve.

Due to scheduling compromises, it is expected from CAE to be a magic wand for giving sign-off to every analysis made, disregarding the level of maturity on the application and the lack of correlation studies. The issue comes from the common misunderstanding on the tool, where the PDO believes it saves both in cost and time, but no realizing that a complete analysis assessment may take much more time than expected.

It is important therefore to have a proper judgment in the attribute where results are demanded and understand that CAE is not for evaluating everything. Rather than saying that CAE is a validation tool of next generation, it is important to understand that its real purpose and function is to be a risk reduction tool, since physical testing is still required for correlation and to give a real validation result (at least for organization in the quest for maturity). The main role and responsibility of a CAE team interacting with the design engineering department is to be a technical consultant team, aiding in the analysis and decision making for systems. (Megalhaes, 2008)

7.5. –CAE in a Mature Product Development Organizations

The place for CAE resides strongly for the alignment in the performance of attributes, narrowing the options to the ones that meet quality, costs, and functional performance in accordance to the project definition. The CAE engineering team provides valuable presence up to the point when designs are completed and final prototypes are assembled for testing.

In a developed organization the role of CAE is to be a testing platform for building confidence.

It was clearly mentioned before the background and characteristics that a CAE engineer must have, and this makes emphasis in PDOs since the computer itself is not a panacea. Commercial hardware and software is unlikely to represent a competitive advantage, it will come within an organization ability to develop proprietary software and coherently integral software, hardware, and human ware into an effective system. (Wheelwright & Clark, 1992)

Mature organizations are distinguished for creating those alternative testing platforms, making work the infrastructure and creating integrative solutions for each attribute or functional area. In the same way their characteristic role is optimizing existing architectures creating a leaner environment for all the already established platforms. (Davies, 2007)

The presence of CAE groups during launch phase of the product development process is not openly needed, since all failure modes are supposed to be avoided and no surprises are expected. But statistically this is not always the case even in mature organizations, making advisable to have a light support from CAE representatives for late problem solving.

Grow in CAE alternatives option is the development of new methods and software platforms enabling the verification and assessment of requirements in cheaper and faster alternatives ways with the robustness required. These activities are present in every organization regardless of the maturity level, but usually the mature branches in large organizations take the lead in the quest of a global CAE standardization maximizing resource in low cost countries.

7.5.1.– Key Differences between CAE Organizations Maturity Levels

Emerging & Growing CAE Organization	Mature CAE Organization
Integrated Group	**Divided Groups**
Work concentrated by time and current programs	Work concentrated toward development of new systems
Not divided in specific functions due size and work complexity	Embedded into different functional and attribute development groups
General knowledge with specialization growth forecast	Highly specialized analysts
Advantage – Easy and nimble information exchange within CAE attributes (NVH ←→ Safety)	Advantage – Easy to integrate physical and virtual functions. CAE group immersed within the attribute or functional group, and embedded into programs.
Disadvantage – CAE group not immersed or directly connected within PD community and its programs, working on a parallel set where analysis and recommendations bypass D&R help.	Disadvantage – Poor cross attribute communication, requiring a strong integration organization in order to optimize work.

Figure 7-4: Key Differences between CAE Organizations Maturity Levels

Advanced tools such as CAE enhance the efficiency of existing processes, but work best when they are embedded in new work processes tailored to their strengths. (Murman et al, 2002)

Figure 7-4 describes the main characteristics that differentiate organizations based on maturity. It is always recommended to start a group in an integral path that slowly and gradually shifts to a functional division as the engineering entity grows in size and complexity at a rate given by the projects handled.

7.5.2.– Advantages and Risks of Virtual Models Reliance

As the reliance on CAE models increases with the maturity of a product development organization, the confidence on the results given by model simulation can be increased an the number of prototypes required can be substantially be reduced with the application of the new computer based tools.

But nevertheless prototypes are still necessary, they are not an element of the product development process that will be erase through the use of virtual modeling but their quality will be greatly enhance and their stages of evolution reduced.

Even in mature and well developed organizations, the risk of overconfidence on CAE assessment may drive the product to downgrade its standards. One famous example was the 2007 redesigned model of the Toyota Camry with presented large reliability problems making Toyota to recall more units than all its North American competitors across 2007 according to Just Auto reports[1]. And although the reason could also be the Toyota ambition to commercialize a global car, it was still know the pride of Toyota by announcing the large usage or virtual tools for the development of this new vehicle.

Moderate confidence and cautious deployment drive organizations to use tools that improve initial design quality and substitute, at least in part, the expensive physical modeling building and prototype construction.

[1]JAPAN: Another Toyota recall: 470,000 cars – http://www.just-auto.com/nd.aspx?ID=92776&lk=dm
US: Survey shows Toyota reliability slippage and Ford improvement – http://www.just-auto.com/nd.aspx?ID=92775&lk=dm

There will be always question about to what extent is it worthy to simulate? Or at what extent does simulation provide results with trust and confidence? Or in which cases I need a physical prototype?

The answers are simple since alternative testing platforms must be used as long as they are a better cost effective alternative in terms of resources and time, and if the representativeness achieved is enough for making decisions. The extent of trust is directly given by the quality of the models and load inputs and the previous correlation work done between physical and testing teams.

The last question and most important one is when a physical prototype is needed, at least in the automotive industry the answer is always. At least one prototype phase that enables the PDO to validate and verify the functional and performance attributes of the product.

"It's not computer literacy that we should be working on,
but sort of human-literacy.
Computers have to become human-literate."
– Nicholas P. Negroponte

7.6. – CASE – Manufacturing Footprint as PD Growth Driver

The arrival of a major project that was going to define the manufacturing footprint for South America arrived in 1999 with the low cost version of the Fiesta, which included the construction of one of the largest and most modern manufacturing complexes of Ford Motor Company.

Ford South America had a small product development team, which got greatly involved with in the development of the vehicle with the intention of gaining expertise for new programs and derivatives and keep the LOPD work of the Fiesta.

Much of the PD organization moved to Dearborn during the development of the program, including the young CAE group, working along in the program to build a critical mass in a period that endured little more than two years. This group later became the core of the product development office of the South America operations.

With the expertise attained and a large project deployed, the growth continued to be gradual, and more engineers were trained by the core group generated. The team was able to form software or model operators within a few weeks leaving the model creation and complex simulation tasks to the core group.

The lesson learned from South America is the link that can be made between a manufacturing project and engineering, to boost the product development capabilities, bringing down key CAE engineers from Dearborn as a start up training team for local engineering, generating at the same time a new low cost country alternative for the main organization.

7.7. – Summary and Conclusions

The adoption of alternative testing platforms is a basic need for any product development organization for being able to continue the growth path of the whole engineering office by developing capacity for complex but at the same time cost effective testing.

Young emerging organizations should take a "pull" strategy for the technology adoption, using the resources available from larger corporate offices or through available tools already in the market or used by supplier of similar industries.

A PDO must first shift to be a data driven organization avoiding the usage of subjective evaluation ratings for the attributes and functional areas where alternative testing platforms are intended to be used. A common strategy among any type of organization must be implemented for achieving maturity which consists on:

1. Measurement of representative road/system loads
2. Development of analytical data for model inputs
3. Development of bench rigs and virtual model correlated through physical testing comparison
4. Creation of methods and design standards.

CAE should be understood as a tool that reduces risk and increase confidence on the design, being able to reduce physical prototypes numbers and stages as well as testing costs; but not yet as a tool for verification sign-off or a PD magic wand.

For young and emerging PD organizations it is highly recommended to start expertise as an integrated general group which will start inner specializations with a long term perspective of being divided into functional groups when the PDO has fully matured. Yet during all this process physical and virtual testing groups should always work side by side for accelerated maturity development thru sharing knowledge and enhancing models creation and correlation.

Chapter 8

=== Conflict Within PDO Teams ===

Difference of opinion leads to enquiry, and enquiry to truth.
– Thomas Jefferson

8. – Conflict within Product Development Organizations Teams

The definition of clear roles and responsibilities is a factor that generates confusion among teams within the product development process, especially in growing organizations. This phenomenon is also possible to see within any team interaction setting during product development, but this case is mainly addressed to the behaviors seen within testing and development groups due their systemic approach to problem solving methods at high vehicle levels.

This phenomenon is considered or relevance to be touched since it is a real case analysis that has been present within the product development organization under study as well as within other organization in which this pattern was able to be observed after careful detailing of the in-house case.

Even though my work experience at a PDO doesn't exceeds since the first day it was possible to perceive a forced interaction between development teams, but coworkers' anecdotes and stories make this conflict to go back almost since the two teams started to coexist within the organization. Why this so evident conflict was allowed to last for so long?

Within this pages we try to explain how is rivalry within teams generated, and why team members or teams within a same organization don't have the willingness to cooperate with other in benefit of the organization as a whole.

The behaviors to be described could happen at any point of an organization or within the daily life o people, but is a phenomenon extensively seen at the verification and validation phase of product development process, since at this step there are intensive problems solving activities and solutions development.

Usually whenever two persons or teams need to interact in the development of a project, they incur in a game in which both need to decide the strategy they will follow to cope for the goals and expectations, in interaction with the ones of others.

8.1. – The Little Empires

When the roles of responsibilities of teams is not clearly defined, people and teams in a organization doubt about the scope that they will require to fulfill in order to have the job done. In some cases, lack of organization may lead to the problem that work is actually not done because no team considered that it was under their scope. There are two possible scenarios when the lack of clarity results in an incomplete task.

- Incomplete deliverables prior to a milestone delivery
- Increased discovery rate time for issues
- Increased rework due incorrect work scope assumptions

On the other hand, the lack of clear roles and responsibilities could derive worse outcomes when it comes to the generation of rivalry among persons or teams, independently trying to cover the same part of a project.

- Intended lack of communication leading to asymmetric information among teams
- Increased effort in the problems solving process due lack of cooperation
- Work waste or effort repetition toward the same target

In growing organizations, this behavior is easier to be found since product development organization will continuously unveil new tasks and responsibilities not thought for their current organization pattern. When this situations occur and there is clear leadership on tasks, the growth of little empires within the organization makes difficult the transition step for reaching formal roles and structures.

8.2. – PDOs Turning Point

A typical characteristic of a small product development organization is that many functions can be grouped within a team responsibilities due to the workload and extent of their operations, accepting this roles in an informal way often in the scheme of "There is something to be done, who is the closest match for the task".

Growing organizations follow organizational schemes that are convenient for their size, keeping integrated modules with a general approaches to attack problems, and with a very broad coverage of activities for each group based on likelihood of doing the activity. During the growing process organizations often follow the structural pattern of a well developed one, which is the case of almost all the branches of well developed organizations. Unfortunately the process of growing involves re-structuring the organization to fit its new responsibilities is not smooth, and the informal roles change to formal structures which traditional way of teams allocation may not be the same as the new structure that needs to be implemented. This results on teams losing responsibilities while other gain them or simply new teams are formed, which in the worst scenario the result is inefficiency in the use of current expertise and resources.

Since the required experience or expertise for new formal tasks is not ready available when an organization needs to grow, scenarios like the previous mentioned are maintained since it is encouraged that any team who can contribute to the completion of task participates within projects development.

These environments where duplication of roles may be present during the formal establishment of new working structures create organizations where the roles and responsibilities are not properly divided.

The likelihood scope refers to path of packaging responsibilities into existing groups assigning them to the one that is most likely to be doing it, being difficult at some points where an activity is a multidisciplinary one.

8.3. – Simultaneous Move Games

The interaction among teams and persons within a product development organization is often a relationship where an entity reacts based on the inputs it receives from other teams and the medium. The day to day interaction is governed by the previous information a team carries, as well as the deductions and strategies each one is able to create. Therefore we can model the behavior to interacting teams and persons within product development as a simultaneous move game, where each team will act based on a personal goal in conjunction with what it believes the other teams will do.

A simultaneous move refers to an event or situation, in which players (persons or teams) are not aware of the strategy that others will follow, and they must choose an strategy relying on rationality and assuming that all the other player are rational and will move toward the strategy that is best for them (Doğan, 2008)

		Player/Team #2	
		Action C	Action D
Player/Team #1	Action A	P_1 , P_2	P_1 , P_2
	Action B	P_1 , P_2	P_1 , P_2

Figure 8-1: Model for Describing Simultaneous Move Games (Doğan, 2008)

Figure 8-1 represents a diagram in which two teams or players present the possible actions they may take during an event in which both player will act simultaneously according to a rational behavior.

In this diagram, players are entities whose decisions have an effect on the other, there could be several numbers of players for any kind of event but for demonstration purposes we will only talk about two players in this case. The game is this case is the problem solving process at product development, but it can be extended to any situation in persons or teams must cooperate in order to achieve mutual advantage as a team rather than seek a personal higher benefit.

Although the diagram presented in Figure 8-1 shows two available actions for every player, it is possible to have a large set of actions as options those players have to play. The actions available represent the set of strategies each player may choose to use, and each one of the will result in an outcome depending on the strategy followed by the others players.

The strategy that a player chooses will be combined in this simultaneous event with the strategies of other players, resulting on individual outcomes for each one of them. As described by Doğan 2004, each outcome of each player needs to be valued base on the interest and preferences, describing the benefit that each outcome will bring to their well being.

8.4. – The Prisoner Dilemma in Product Development Organizations

The Prisoner Dilemma as defined by Dixit & Skealth 2004, is the situation in which the players interacting have dominant/strong strategies but the equilibrium that comes from choosing them, results in a worse outcome for all players than the one result of choosing their dominated/weak strategies.

Game Players:

- Development Team #1
- Development Team #2

As example, these teams could be represented by a team who historically managed the responsibility for a certain activity, counting with certain general knowledge and expertise on it; and another recently created to cover a need that emerges as part of growth in the organization, and requires special attention of.

Another example could be two development teams already well established, caught in a difference of opinion about their roles and responsibilities when a new multidisciplinary task appears.

Development Teams Strategies:

- Full Cooperation – This condition establishes a game with full information exchange and partnership, making the most with all the available resources, using expertise and knowledge where it could make its best.

- Asymmetric Cooperation – This condition establishes a game in which a team/player or more will stop sharing or cooperating, in search for taking the lead of the project.

At the very first thought it would no be easy to consider rational the option of not cooperating or sharing information and joining efforts, but there are several factors that may lead to this kind of behaviors, being apparently rational at the moment a strategy is chosen. Later on this chapter we will describe the mechanism that pull a player to choose an Asymmetric Cooperation strategy, but understanding the payoffs for each outcome may also help to understand the behavior.

Development Team Payoffs Definition:

- 4 – "The Hero", "The path for an Outstanding PR[1]" – The team was able to lead and provide a solution for the issue, demonstrating expertise, ownership, and control of the task. This outcome is the result of a team taking advantage and control of the information available by not sharing full information, while the others team provide all their information toward the issue resolution. A subjective but very valuable form of seeing this reward level could be obtaining an "Outstanding" for performance review.

- 3 – "The Partner", "The path for an Excellent PR" – The team provided a solution to the problem by co-working successfully with other teams, sharing responsibility, information, and control of the task. This is the outcome result of both teams act under their full cooperation strategy. A subjective but very valuable form of seeing this reward level could be obtaining an "Excellent" for performance review.

- 2 – "The Rival", "The path for a Sufficient PR" – The team managed to solve the issue with un-easy cooperation, or managing asymmetric information with respect other teams. This is the result of both players choosing an Asymmetric Cooperation strategy, in which each tries to solve the issue, lead it, or take control of it. A subjective but very valuable form of seeing this reward level could be obtaining a "Sufficient" for performance review.

[1] PR or performance review, is the grade managed by the sponsored product development organization based on completion of goals and achievements during a year. These grades are divides in O – Outstanding, E – Excellent, S – Sufficient, I – Insufficient.

- 1 – "The Tool", "The path for an Insufficient PR" – The team was not able to have a significant role in the problem solution process, acting in a support role, without having clear involvement or complete information about the solution development. This outcome is the result of the team choosing a full cooperation strategy and not receiving reciprocal, creating feelings of being used as a tool for others teams success and credit. A subjective but very valuable form of seeing this reward level could be obtaining an "Insufficient" for performance review.

Establishing a clear ranking among the possible outcomes is necessary in order to establish the payoffs, which could be not just something tangible. For terms of analysis it is convenient to be able to assign a numerical value to the benefit obtained for each player or team, but in most cases this payoff is difficult to represent since it depends on the expectations, believes, and emotions of the players.

Figure 8-2 shows the interaction game that can be seen among persons, teams, or even within organizations that cooperate toward the success of a company, but may compete for the resources of it. Strategies for each player are shown, and the payoffs for each combination number is described in accordance to the previous numeric representation description and ordered as shown in Figure 8-1.

		Development Team #2	
		Full Team Cooperation	Asymmetric Cooperation
Development Team #1	Full Team Cooperation	3 , 3	1 , 4
	Asymmetric Cooperation	4 , 1	2 , 2

Figure 8-2: Strategies and Pay-off for Development Teams in Coopetition

Since each team will aim to maximize their profits, each will implement a dominant strategy. A dominant strategy is the one that brings the higher payoff for a player, regardless the action that any other player may choose (Doğan, 2008), and is natural for any rational player to play that strategy.

In cases where there is rivalry within two or more team in an organization, it is natural to expect them to play their dominant strategies. In this example, presenting an asymmetric cooperation environment is the dominant strategy for both development teams, which lead to a Dominant Strategy Equilibrium.

Since both development teams will follow their dominant strategies, we achieve a configuration of strategies (one for each player) such that each player's strategy is best for him, given those of the other players (Dixit & Skealth, 2004). This configuration of strategies is called the Nash Equilibrium of the game.

In this example, both development teams have the dominant strategy of deploying an Asymmetric Cooperation strategy, but the dilemma resides in both players not achieving the optimal outcome of the game but rather the equilibrium of their strategies.

		Development Team #2	
		Full Team Cooperation	Asymmetric Cooperation
Development Team #1	Full Team Cooperation	3 , 3	1 , ***4***
	Asymmetric Cooperation	***4*** , 1	**2 , 2**

Figure 8-3: Nash Equilibrium from Development Teams Prisoner Dilemma

As shown in Figure 8-3 both development teams has Asymmetric Cooperation as their dominant strategy since it has a higher payoff for them for any strategy the other team chooses: 4 over 3, and 2 over 1.

By falling in this behavior, both development teams fall in a condition in which they achieve less than if none of them would have defected. As in many projects and programs happen, the final outcome is less than it could have been if no team decides to search self benefit through their dominant strategy. In this example the dominant strategy equilibrium brings a payoff of "2", losing the payoff of "3" result from the equilibrium of weak or dominated strategies.

The existence of this type of behaviors is clearly existent within product development organizations although it is often ignored and the root cause is not clearly addressed. In most cases there is not a conscious effect of the disadvantage of the prisoner dilemma among members of an organization, and the reason relies in the fact that the work and the issues gets done and resolved in one or the other.

Although the resolution of the issues and the completion of work is the expectation of the organization as a whole, it is important to address this organizational issue promptly not only to eliminate the risk of damaging the people within the organization and reduce future effectiveness, but also to maximize the current payoffs for the organization as a whole.

As an organization composed of systems, the success of every subsystem of it contributes toward the success of the system. In this game, the product development organization receives the result of the combined effort from all its system, and therefore conflicts within them impact on the overall performance of the group.

As shown in Figure 8-4 the total payoff reflected in the product development organization is the less at the equilibrium point, and even when only one of the development teams' defects from the full cooperation strategy, the combined effort of both teams gets affected reflecting in a lesser payoff for the company.

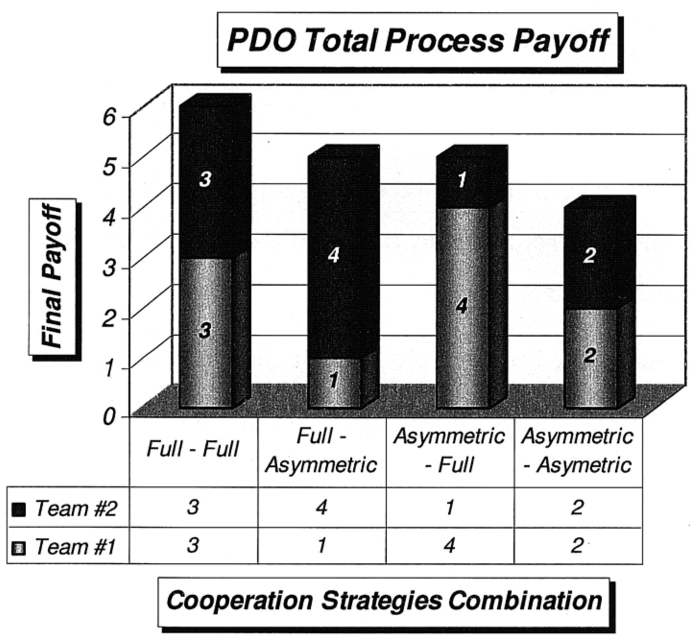

Figure 8-4: Combined System Total Payoff

One of the reasons for this issues not being addressed properly relies in the lack of awareness of the outcome that could have been produced in previous projects through improvements in the social behavior of the teams. The management team of the organization may be in a comfort state by knowing that the projects are completed and the issues resolved, obtaining payoffs around "4" to "5" according to the example, but unaware that a corrective action may lead the company to achieve a higher level, "6".

Figure 8-5: Mulaly's strategy to eliminate collective irrationality

In the automotive industry, the prisoner dilemma may lead to serious organizational problems. As an example, the new model for Ford Motor Company promotes the elimination of these silos, fostering the team cooperation and communication toward common goals.

Without addressing the root causes that drives a team to defect, lead to have organization with several teams, with one individual plan per team, that apparently are aligned to one common goal.

Figure 8-5 shows Alan Mulaly's strategy is simple and concrete, and its interpretation reveals that the company is finally aware of this kind of issues, in which the individual rationality results in a collective irrationality.

Achieving the convergence point for the dominated strategies of the development teams, a product development organization may reduce their development times and the discovery time for issues and problems. In the same way, deploying full cooperation strategies contribute to develop better products, avoiding mistakes and rework, improving team members moral by increasing the sense of ownership to the project and to the team by itself.

8.5. – Defection Drivers in the Day to Day Interaction in PD Groups

The technical reason for defection is that the strategy mix that brings the best combination for both teams and the company, relies on dominated strategies, in other words there is always a strategy that apparently will do better. Rational players should not be playing dominated strategies, and since they assume that other player are rational, they assume that others will not be playing their dominated strategies (Doğan, 2008).

It is clear that there is better path, which will offer a better payoff for both teams as well as for the product development organization in general, but each one the elements within the organization have special motives, reasons, and goals; and their interaction with other may lead to different kinds of strategies that may not align perfectly with an optimal equilibrium.

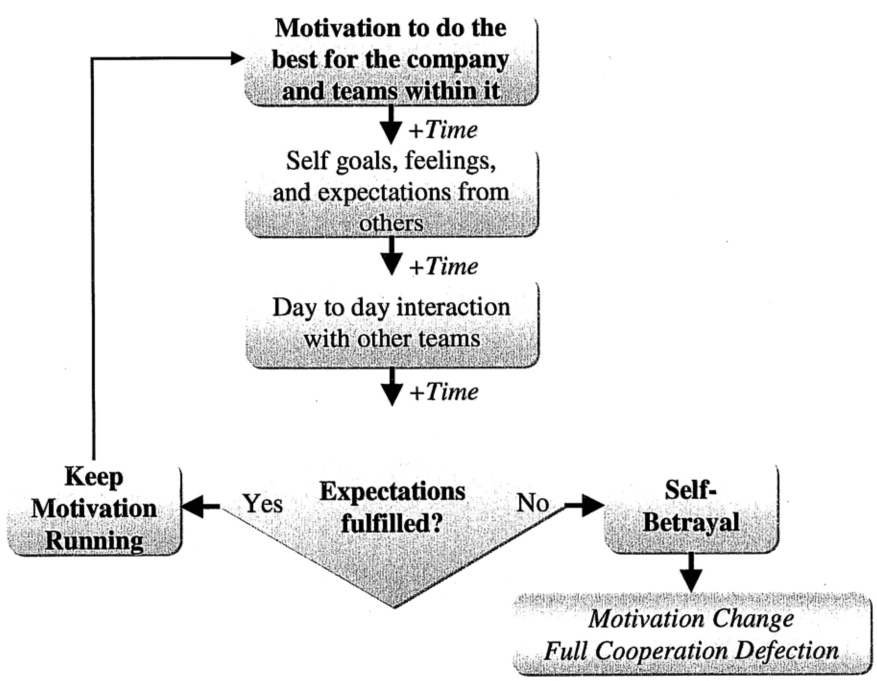

Figure 8-6: Motivation change within persons and teams modified from (Arbinger, 2002)

A behavior that is often seen in organizations is the motivation change within its members as time goes through. Each team or individual carries goals that are aligned with the company goal but accompanied by feeling and expectations, and the day to day interaction with every other member or team in the organization will result in an evaluation of expectations fulfillment at the end of a day or project.

8.5.1.– Self-Betrayal – Failed Expectations

Figure 8-6 shows the term Self-Betrayal, which Arbinger defines as the resistance to others, and to what others want or need from one to do for them. It refers to the denial phenomena, in which a player decides to act contrary to what another player, which by X or Y reasons becomes an annoyance, without regard on knowing that the action or lack of it causes a problem or disadvantage to the person or group.

Every person and team as a collective entity carries a motivation that interacts with the ones of others and may clash bringing a conflict into the group. At the end, the outcome will be compared to the original expectations making a judgment of the entities that interacted on the creation of those outcomes. If the result of the judgment made is not favorable toward the other teams in coopetition, there point of view with them drastically changes, becoming obstacles for the fulfillment of expectations, creating resentment and defection from cooperation.

As stated by Arbinger 2002 when the defection of a team member happens, their thoughts for the cause of the problems are very likely to be found in other members of the organization, the supervisor, the manager, and the coworkers with whom it has to interact. This defection point lead to blaming and keeping a certain level of resentment that instead of helping deteriorates the environment and moves the daily motivation and expectations.

The commitment to work as a team may be converted after numerous repetition of the game into problems that inhibit a full cooperation environment.

Team members may experience failed expectations and live hard moments in their day to day interaction, creating an environment over which the team results are no longer as good as they were, generating frustration and disappointment in a reinforcing loop dynamic (Hackman, 1998)

8.5.2.– Unclear Roles and Responsibilities

Although this is a clear factor within multidisciplinary activities in an organization, it is clearer to see during the growth of an evolving product development team. Growing organization present integrated and concentrated structures, where a team gives a general approach to a wide range of task, in base of the degree of likelihood to fit under them.

When new responsibilities appear, and development teams face broader objectives, the lack of roles clarity lead to an ambiguity state where there is no certainty of what kind of problems must be solved by whom, or in which degree of responsibility. The uncertainty created by unclear roles and responsibilities often creates rivalry within groups that expect to own or control these new tasks. (Lafasto & Larson, 2001)

Kanaga & Prestridge express that one of the main drivers to create conflicts within organizations and teams is the lack of clear roles, since this leads to misunderstandings due the perception difference that is created. In an attempt to get ownership of the role and responsibility that each team perceive it should have, zero cooperation is given to other teams trying to complete the same task, even reaching points where one team deliberately puts an obstacle in the path of others in order to achieve its desired positions.

Organizations with unclear roles and responsibilities may present teams that will try to engage power plays for authority, falling in conflict with other teams that either do the same or just try to complete the task as it would have been done normally. (Kanaga & Prestridge, 2002) The theme of the conflict is to win the role or responsibility through unofficial auto-assignment, and in the process each team may tend to run over the other to get the same task done, without willingness to help rival teams in the development of a project where mutual interaction is required.

8.5.3.– Dismembering Existing Structures

Growing product development organizations encounter new tasks as the project complexity they manage increases, and in the same way, operations that were generally done by the most likely organization will face the need of creating a specialized team to address them.

In an attempt to direct new tasks to cover within the exiting organizational structure, product development teams may fall into the mistake of dismantling existing structures which previously covered the task in a very general way in order to assign the responsibility to a different group.

This error wastes the experience, knowledge, and expertise gathered within the existing structure, for low as it may be, but useful for the successful development of new projects. This behavior creates a sense of uselessness by the dismantled organization reducing morale, while it also requires a lot of resources to train a new task force from scratch.

In case new group is needed to develop a task, it is acceptable to dismember an existing structure if their resources are wisely used in the creation of the new one. Management requires the ability to define the needs of this new task and locate the recourses to cover it within existent structures. The key task is to identify the members of the exiting structures who have experience or knowledge within the new task to specialize and take them in order to make full use of the time and investment that they carry from the past.

8.5.4.– Sense of Ownership – Pride

Prior to the growth stage in development teams, team players have created a sense of ownership and pride for tasks that in one way or the other they used to cover, feeling pride for the actions and results achieved in the past. This behavior often happens with several teams at the same time for a specific task, due the uncertainty of their roles, and the sense of authority within the development of it.

In the same way, when a new development team is formed to cover a necessity, it natural behavior is to develop the same sense of ownership and pride to their new job description.

When the previous two settings collide, cooperation may be broken within teams when each development team wants to keep the leadership, making them to work on separate paths in a quest to complete it earlier and better.

Unclearly defined roles within development teams, and poorly created team are the triggers for reduced cooperation when roles overlap over tasks development and testing teams fall into a conflict.

Ignorance or tolerance toward this events create only increase those feeling and expectations within development teams, generating efforts of monopolization of tasks that require several teams to cover successfully, resulting on failed, uncompleted, or reduced performance results.

8.5.5.– Lack of Trust

Within testing and developing groups in a product development organization, the pressure to deliver a product in time leads to expect positive testing results with the first trial.

Especially in small growing organizations, the testing teams don't have the equipment and facilities to conduct testing under the corporate standards that could be applicable for it, triggering mistrust from other development teams when a testing result is not what they would expect or want.

Even though a testing team may have developed local testing procedures or derivations to fulfill the validation plan, equipment is often a strong limitation that sets the boundary of what is achievable to measure and verify due the accuracy of the testing equipment or the derivations made to adapt a procedure.

In either case lack of trust form one developing entity to the results provided by other, seeds conflict and mistrust within both teams, motivating a separation among them. Reduced morale and resentment is the result emerges from the development team providing the data that is under question, in which the result is often a strategy of not sharing data or observations with the other team.

The team that doesn't trust the data generated by other strengthens their sense of ownership of the task trying to control it and perform independently, often generating high amounts of resources waste. Rework, test repetition, and unnecessary testing are the most common wastes seen by the organization when this behavior is present within testing and/or development teams.

Lack of trust to a team performance is always at hand with feelings of superiority in the same task, where the work from other team is constantly requested to be reviewed and redone, as the team considers itself to have expertise enough to know that tasks are done wrong when data is not the one expected.

In cases where teams are dispersed world wide, there is a common behavior where a country doesn't trust the testing results obtained by other, repeating the testing routines as a check point that creates duplication in the process avoiding an efficient use of resources.

8.5.6.– Quest for Rewards

Defection from cooperation among teams could be generated by a motive of outstand or excel within the organization. Although the quest for excellence is natural in every individual, group, and in a product development organization as a whole; teams may present collective irrationality when reaching excellence comes at the expense from the performance of a different team.

The collective irrationality is reached when teams in shift from cooperation to coopetition within a product development organization where the evolution of its structures finds overlapping roles that are not clearly defined.

After a coopetition environment is set, collective irrationality shifts the environment to pure competition in the independent quest of reaching higher goal blocking other teams who could interfere with it. This collective irrationality is boosted when the development projects within the organization are not enough for the total headcount assigned to them, or when the rewards systems are incapable of covering a large number of teams or individuals, regardless of their performance.

Factors that motivate defection from open and full cooperation strategies within development teams may include:

- Performance review negotiating elements
- Headcount for future years
- Assignation of new of more complex projects
- "Pride, Ego"
- Program or Project development authority

8.6. – Asynchronous Decisions on Real Case Scenarios

Something very important to notice from the prisoner dilemma described earlier is that there is a slight difference in the time that actions occur, in other words, the time to achieve the dominant strategy equilibrium in which brings the worst outcome to both teams and the organization they develop in as a whole. Defection could be from one instance to the other, but in the case of testing and development teams as well as other human interaction settings, there is a previous learning process that may retard the defection process.

The factors described above form some of the mayor causes of conflict that a could be found at the interaction points of developing teams within a young and rapid growing product development organizations.

Whenever multiple teams or individuals are assigned to perform as a team, it is a common action to expect that putting entities to work together in a project to be enough for forming a high strength team. In order to conform a team it is imperative to understand the history, experience and knowledge of ever piece, as well as having information about their expectations and thoughts.

As shown in Figure 8-7 when teams are formed and need to interact with different teams within an organization, there is a negotiation of their expectations an goals, where the rules of the game are defined with roles and responsibilities for each player. Although unclearly defined or uncertain, as long as these negotiated rules keep going, the teams will be interacting within a comfort zone where their productivity will be proportional to the degree at which their goals and expectations were able to be managed into rules and roles.

When one or more of the defection factors described before presents from one or more teams during the day to day interaction, there is a conflict between them. This conflict could be taken care immediately although with its effectiveness could be limited, such as the case of a simple apology from the defecting team, which will bring the game into its comfort productivity stage but with a damaged relation among teams. This damage is cumulative and may cause larger damage if the Defection – Simple Apology keeps repeating time after time.

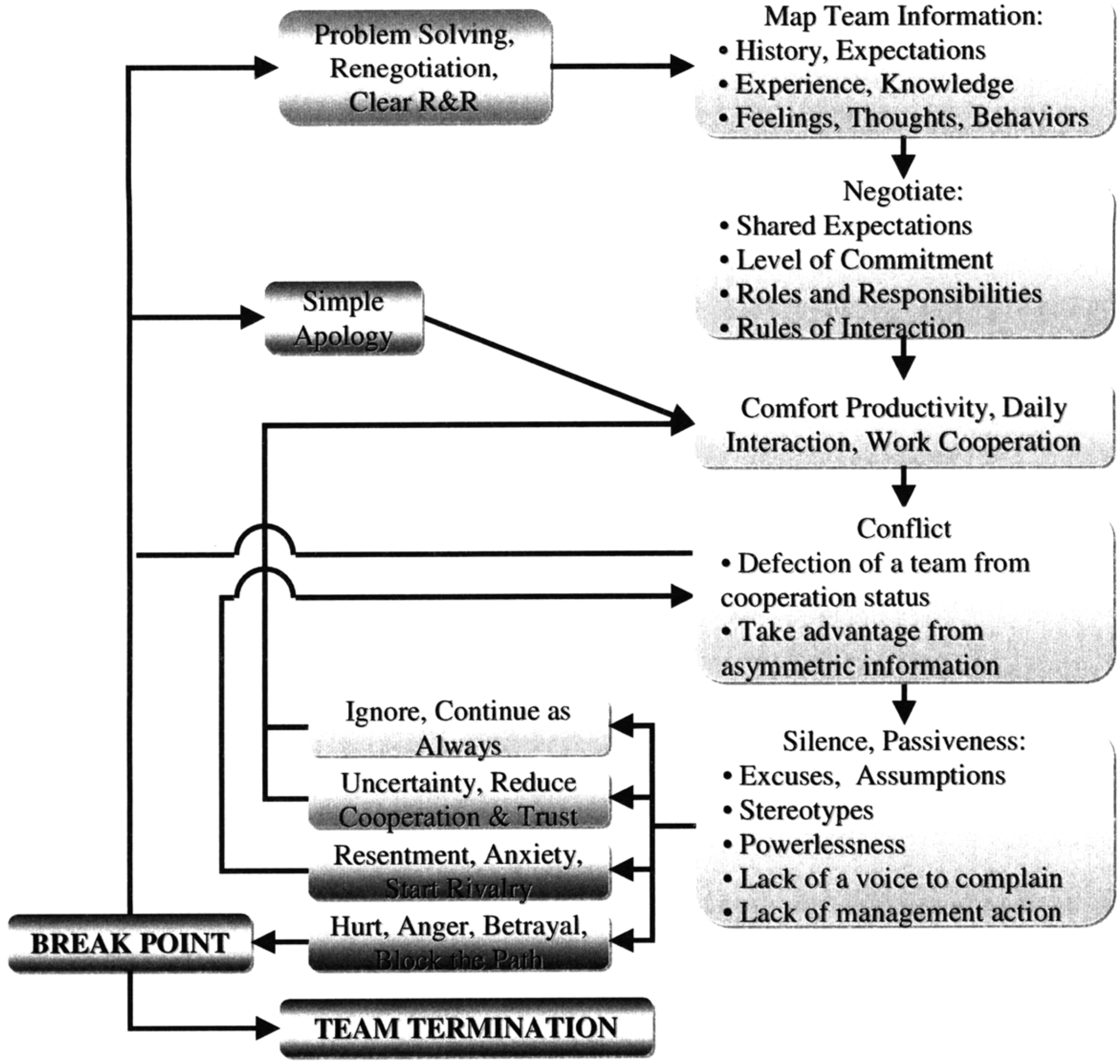

Figure 8-7: Path to teams/persons defection, from (Ely, 2008)

The Defection – Simple Apology loop shown in Figure 8-4 can be repetitive, which promotes the team to interact in a fantasy comfort environment where a deterioration of the performance achievable by the group will be diminished.

In order to break the cycle, it is required to go deeper into the conflict and be able to renegotiate the teams inner motives as well as they roles and responsibilities, managing carefully the existing resources and moving them to a location where they could perform best according to their knowledge and experience but balancing where their goals and expectations as well. As stated by Robin Ely, the renegotiation step is the path for achieving real growth and learning, and be able to achieve real solution to the problems within an organization.

As remarked by Robin Ely, when the conflict loop lead to diminishing morale across the teams facing defection as the annoyance comes into learning from the players, creating uncertainty and lack of trust for the other teams to interact. Teams facing this behavior are careful on the information that is shared and when is shared, which may lead to some delays in projects development.

Further repetition of the game creates feelings of resentment and anxiety among teams, where misunderstandings and lack of trust drive teams within a same organization to compete against each other.

Although rare, and generally presented more often within individuals rather than teams, is the generation of strong negative feeling from one entity to the other. The conditions to achieve a comfort zone productivity are lost within the conflict which drives a team or part of it to a break point. This breakpoint represents the last opportunity to act and renegotiate the game and achieve a solution of it, before the team terminates its activities. Common signs of this behavior can be often seen when several members of a team suddenly change the departments or areas where they work, or when individual resign to the organization they belong.

As the team integrity is diminished during defection and conflict cycles, the perception of the individual change within the organization, building the need to defect again as a dominant strategy. With perception changes team players no longer consider defection as something harming the organization, but as the only way left to keep it running, creating what was previously mentioned as "Collective Irrationality"

Self-perception after Failed Expectations	Collective Irrationality	Perception of Others after Failed Expectations
Victim	↔	Lazy
Hardworking	↔	Inconsiderate
Important	↔	Unappreciative
Fair	↔	Insensitive
Sensitive	↔	Hypocrite
Good Manager	↔	Lousy Managers
Good Worker	↔	Lousy Workers

Figure 8-8: Team perception change due failed expectations (Arbinger, 2002)

But what permits to defection and conflict cycles to occur? One of the main factors is lack of awareness by teams management, where the problem is ignored or superficial actions create the illusion that the problem is addressed "Simple Apology" (Hackman, 1998)

When the problem is ignored it transcends and grows, becoming to be seen as and accepted condition part of the normal day to day interaction among teams. Poor team communication status is achieved, and teams try to avoid each other as much as possible having the only the minimal interaction needed to the get the jobs done.

As noted by Hackman, to ignore a problem that repeats itself over time within an organization, is the key driver that its members use as justification for using a non-cooperative strategy.

8.7. – Strategies for Cooperation Strategies Enforcement

Game Theory concepts cover some options in order to break the prisoner dilemma, although they have to be used carefully within product development organizations in order to avoid a morale drop from team member and create defection from persons to their teams or to the organization. Game repetition, punishment/reward, and strong leadership are the strategies that a product development organization must follow in order to achieve an optimal environment for its products and processes, but the real solution is a combination of all of them in a delicate balance.

8.7.1.– Game Repetition – The double edged sword

In any product development organization, problem solving activities are frequent, and that they are event that only happen once, but repeat across the development of a program, across several programs, reaching points in the daily life discussions or participations. Therefore, the presence of this prisoner dilemma over the authority and control of problem solving activities is a repetitive game that could lead to devastating consequences within an organization if it is not addressed appropriately.

Repetition occurs naturally in every organization, but in order to make it relevant tool it is important that management and the whole organization is aware of the combined payoff possibilities as shown in Figure 8-4.

According to strict game theory concepts, the repetition of the game can lead to the solution of the cooperation issue, but considering the inner dynamics of feelings and expectations from individuals and teams, game repetition could be a trigger for defection drivers among teams if the root cause is not properly addressed.

In the day to day activities of the product development process, the repetition of the game is unknown, it is uncertain how many projects would the teams will be developing together, and it is often considered infinite for job commodity reasons. In some cases, two team or organizations will be competing for a single project or opportunity, or in a repetition of the game with know length.

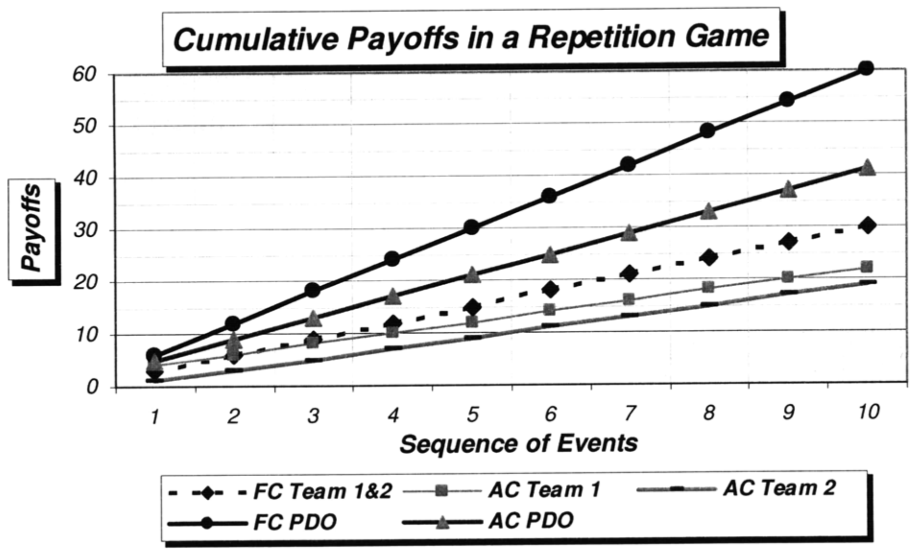

Figure 8-9: Cumulative Payoffs in a Repeated Prisoner Dilemma
FC: Full Cooperation Strategy, AC: Asymmetric Cooperation Strategy

In most cases, a product development organization may be in a comfort zone and ignore the advantages of defending a cooperation environment, such as the results are delivered and the projects completed. The cumulative payoff difference shown in Figure 8-9 explains the potential that a lean operation would bring to the product development teams interaction, and represents in the same way the associated losses product of rework such as testing repetition, parallel problem solving efforts, and unnecessary testing.

The cumulative payoffs difference results in the sum of successive payoffs described by Equation 8-1 and Equation 8-2. In this case the asymmetric cooperation payoff is dependent of grim strategy, which is the worst case scenario the group morale has reached a point in which cooperation among groups is unlikely but force in order to complete projects and solve issues.

$$PayOff_{FullCooperation} = 3 + 3 + 3 + 3 + 3 + \dots$$

Equation 8-1

$$PayOff_{AsymetricCooperation} = 4 + 2 + 2 + 2 + 2 + \dots$$

Equation 8-2

But as shown in Figure 8-6 and Figure 8-7 the extension of this game may increase stress within the product development team members increasing their tendency toward team defection.

According to Dixit & Skealth, in repetition games it is possible to observe two kinds of strategies: The grim and unforgiving strategy in which a player will defect forever whenever an opponent defects, and the Tic-for-tat strategy in which the player will follow the previous move of the opponent.

In reality it is observed that co-opetition teams neither follow any of these repetition strategies by standalone but rather a mix of them. In fact it has been observed that real life team are pretty forgiving and there could be several instances of defection by other team, in order to apply a defect grim strategy following the path described in Figure 8-7, but at the same time they behave willing to return to a cooperation strategy if the opponent shows cooperation like in a tic-for-tat strategy.

Figure 8-9 show the absolute cumulative payoffs that could be lost, but it is important to estimate the present value of the payoffs that are lost event after event in comparison to the extra payoff won by the team at the defection point (Dixit & Skealth, 2004).

If a team has an awareness of the difference in payoffs being lost, it is possible to estimate the present value of them at each event in the sequence through a compound interest analogy according to Dixit & Skealth. The theory relies that in each the effect of reinvesting the payoff lost month after month under a certain rate comparing it to the one won at the time of defection, on the basis that the present value (PV) of a certain level of payoff is not worth the same today than it is tomorrow. Equation 8-3 describes how the present value is managed period after period in order to be able to compare the payoffs to obtain in the future to the ones obtained today.

$$
\begin{aligned}
PayOff_{+1} &= PV + rPV = PV \cdot (1+r) \\
PayOff_{+2} &= PV + rPV + r(PV + rPV) = PV \cdot (1+r)^2 \\
PayOff_{+3} &= \ldots = PV \cdot (1+r)^n \\
\Rightarrow PV &= \frac{Payoff_{+n}}{(1+r)^n}
\end{aligned}
$$

Equation 8-3

Where:
PV = Present value
r = return rate (positive number)
n = number of events (hopefully infinite or unknown)

The ratio at which the payoff are affected is denotes as discount factor (δ) as shown in Equation 8-4, and defined as the fraction by which the next period's payoff are multiplied to make them comparable with the current period's payoffs (Dixit & Skealth, 2004). In management terms, and applied to payoffs that may be not tangible, the discount factor is a punishment measure that is applied in order to be able to evaluate the payoffs being laid of by choosing the teams dominant strategy.

$$
\delta = \frac{1}{1+r}
$$

Equation 8-4

Equation 8-5 and Equation 8-6 describe the infinite sum series that are product of an unknown duration game following the Full and Asymmetric Cooperation respectively. Since the discount factor is a number smaller than one by the boundary that the return rate is a positive number, the infinite sum of the discount factor converges in a specific value as shown in Equation 8-7.

$$
D\_PayOff_{FullCooperation} = 3 + 3\delta + 3\delta(\delta) + 3\delta(\delta)(\delta) + \ldots
$$

Equation 8-5

$$
D\_PayOff_{AsymmetricCooperation} = 4 + 2\delta + 2\delta(\delta) + 2\delta(\delta)(\delta) + \ldots
$$

Equation 8-6

$$1+\delta+\delta^2+\delta^3+\ldots+\delta^n=\frac{1}{1-\delta}$$

Equation 8-7

The discount rate is a value that needs to be determined within a range that will promote cooperation and avoid deterrence among the product development teams, it has to make more valuable the cooperation strategy path through the analysis of the value of the payoffs in the long term when several projects are going to be carried out by the same teams or will be needing to interact in order to complete the tasks of the organization.

Equation 8-8 shows the calculation of the discount factor range by establishing that the payoffs of full cooperation strategies need to be larger than the ones with asymmetric cooperation. The key point is to establish a value that motivates the team to deter from their dominant strategies.

$$D\_PayOff_{FullCooperation} > D\_PayOff_{AsymmetricCooperation}$$
$$3+3\delta+(3\delta)\delta+(3\delta)\delta^2+\ldots > 4+2\delta+(2\delta)\delta+(2\delta)\delta^2+\ldots$$
$$3\cdot\left(\frac{1}{1-\delta}\right) > 4+2\cdot\delta\cdot\left(\frac{1}{1-\delta}\right)$$
$$\delta > \tfrac{1}{2}$$

Equation 8-8

Current ambiguity within the automotive industry may become a driver in order to force cooperation, but at the same way inserts an interesting tool to the game, since now it is uncertain if there is going to be a new project to fight for in the future. In those cases the discount factor strategy could be applied using the probability of a next event to occur, although this method relies on high uncertainty since it is unlikely for organizations to expect the probability of disappearing and not fomenting cooperation at the same time.

In cases in which repetition of a game is know, the repetition strategies are not effective as a tool since players are able to analyze when is it better for them to defect and obtain the best payoff. Taking into account that each player is rational and considers the others to be rational, each player will know that his opponents will be playing the same strategy and defect one step earlier.

This behavior been seen through a rollback analysis, finalizes on the teams defection on the very first step, an effect commonly known as rollback equilibrium (Dixit & Skealth, 2004)

In the cases where the end of the game is known, stronger discount factors to the teams payoffs are required to be applied since there is little time in order to make it effective and therefore in order to create the desired impact in just one move, the discount factor tends to be large. Being able to have repetitive instances, with an unknown number of them, facilitates to manage discount factor to deter from dominant strategies. Applying this technique in no repetition games is difficult to achieve.

8.7.2.– Punishment and Reward System – The club and carrot path

In most organizations, the stability and good relationship between management and team member create forgiveness environments, which are often used as confidence intervals to break rules among the organization and pursue personal goals over the ones of others.

A clear behavior of the previous statement is that teams would search to change the rules of engagement to its own advantage if those rules are not properly fixed or protected.

One of the main reasons that teams or members within a product development organization decide to defect to dominant strategies in pursue of individual goals and expectations, relies in a lack of fear of punishment by the same team or its management.

Fear from punishment is a correction strategy in order to avoid teams to pursue individual goals, especially when there are no conditions for a interaction game to be repetitive (Dixit & Skealth, 2004)

Trying to apply the same approaches as in repetition events will be very hard to do, since there is not an adequate amount of time in order to feel the effect that a discount factor will have requiring them to be unfeasible in some cases. Taking Equation 8-8 as an example, players will defect in a single move, and therefore a new discount factor is obtained as shown in Equation 8-9 where we basically see that if there is no repetition of the game the discount factor must be total eliminating the payoffs in order to avoid defection.

$$D\_PayOff_{FullCooperation} > D\_PayOff_{AsymmetricCooperation}$$
$$3 + 3\delta > 4 + 2\delta$$
$$\delta > 1$$

Equation 8-9

Since the return rate must be a positive number, it is not possible to achieve discount factors larger than the unity. Instead of that a punishment to defector teams must be applied in order to set fear seeds and make a general avoidance toward defection.

The need for punishment strategies is needed within every product development organization, and as stated by Doğan, 2008, punishment strategies must be strong enough to motivate the player not to defect, but at the same time must be moderate to remain credible.

Management has the capability of establishing punishment to the teams and individual within an organization, setting examples that create fear and avoidance to defection within the rest of the whole. This objective can be accomplished by either punishment or rewards.

Punishment and reward systems must be strong enough in order to be able to break the prisoner dilemma and change the payoffs of the teams, moving the dominant strategies to the optimal equilibrium point. In order to accomplish this, a careful study of the emotional intelligence of the player is required in order to quantify the payoffs for the possible outcomes of the game. Rewards have the same effect, since they must be good enough in order to modify the payoffs of the game, motivating teams into cooperation in order to achieve them

The complexity of assignment a punishment systems relies on the value that punishments or rewards must have. In the case of punishments systems, they must be hard enough to deter from defection but not so hard in order to not damage the internal systems of the organization. A punishment that is not correctly applied, or that is hard enough that damages the inner operation of the teams lack the credibility of being applied and players will learn the situation in which management will defect from applying a punishment in order to preserve integrity.

Rewards system carries similar complexities, although their problems relies in requiring a reward that is valuable enough in order to motivate people and teams toward the desired path, but that are also manageable by the available resources of an organization. The value of a reward system must be equal or less to the amount of combined payoff lost within team defection cases, in order for the system to be sustainable and productive. Since a reward can't exceed this value, there are certain limitations that make it easy to be applied, and therefore many reward strategies fail, being commonly called "The hand shake on the back"

On the other hand, reward systems tend to nullify themselves when an equilibrium point is achieved, since most time it is not feasible to maintain a reward of all the team players during long periods of time.

Figure 8-10 shows within parenthesis, how punishment strategies can be applied to defecting player, by reducing the payoffs achieved by them in an attempt to nullify the prior dominant strategies existing within their mutual interaction in the organization. In order to be effective, the strength of the punishment must be enough to make the full cooperation payoffs more attractive, but applied with moderation in order to avoid greater damage to the organization members.

		Development Team #2	
		Full Team Cooperation	Asymmetric Cooperation
Development Team #1	Full Team Cooperation	3 , 3	1 , 4 *(-2)*
	Asymmetric Cooperation	4 *(-2)* , 1	2 *(-2)* , 2 *(-2)*

Figure 8-10: Modified payoffs through defection punishment system

Altering the payoffs create new dominant strategies for each one of the teams, in which now they achieve a larger benefit by sharing complete information and working together in the resolution of problems as well as in the development of new programs and opportunities.

		Development Team #2	
		Full Team Cooperation	Asymmetric Cooperation
Development Team #1	Full Team Cooperation	***3*** **,** ***3***	1 , 2
	Asymmetric Cooperation	2 , 1	0 , 0

Figure 8-11: New Nash equilibrium from broken prisoner dilemma

Deciding between a punishment or reward system depends on the leadership type and team players response within a product development organization. But also as mentioned by Dixit & Skealth, punishment systems are not very effective in games where the number of players is large since uncertainty of who to punish in case of defection may cause a detriment on the global moral of the group. Uncertainty arises when there is difficulty about determining the element that initiated a defection and therefore triggered the prisoner dilemma outcomes.

As a final point, the key reason in order for punishment and rewards systems to be effective is the ability of teams and team players to value a future state, which can only be assure by establishing a repetition of the game and assuring that there are always more projects to come.

8.7.3.– Strong Leadership – Pulling into the right way

Although a problem is clearly evident within an organization, often teams make no move in order to solve the un-functionality, taking a passive attitude either waiting for the other team to surrender or make a move, or accepting the condition as a day to day interaction. A strong leadership is the element required to drive this behavior from a passive state and direct them toward a solution in order to stop diminishing the teams performance.

Engineers may gather and think on the best solution to solve a technical problem, but it is clear that the most complex to solve within a product development organization are the ones involving emotions, attitudes, values, and motivations. (Lafasto & Larson, 2001) It is not an easy task, but management and team leaders require a strong skill in managing their team emotional intelligence.

As stated by (Druskat & Wolff, 2001) the emotional intelligence[1] among groups is precious and has to be nurtured by trust, group identity, and collaboration among groups. The results of achieving high emotional intelligence are stronger participation indexes by individuals, as well as higher cooperation and collaboration among groups. These results are the main drivers for better decision within product development organizations, and creative problems solutions within development and testing teams.

Nurturing the emotional intelligence among groups is a path that foments learning of behaviors (trust, cooperation, identity), and the strength of this emotional culture leads to organizational change of considerable strength. The "Simple Apology" loop works only as a reflective action that doesn't provide any kind of learning and only provides temporal changes. (Robbins & Judge, 2008)

Learning could also be done for negative behaviors such as the lack of trust or cooperation, therefore when development teams fall in a repetitive prisoner dilemma within an organization, management is the breaking point that may change the path of the game and alter the equilibrium forcing at least one player to follow a desired strategy through changing the payoffs of each player via rewards and punishments.

Learning is the path for increasing the probability of occurrence for good behaviors across interaction and performance of task among several teams. (Robbins & Judge, 2008)

The management team is responsible at some extent of setting rules of engagement across teams, as well as evaluating their behaviors, rewarding them or punishing them. Within the rules of engagement, management teams have the ability to enforce a culture or action agreements for the groups interaction, which often are called corporate strategies.

The key element is to establish that cultural or behavioral changes is not something achievable by a tablet with principles that teams must follow to interact, as also mentioned by Hackman, the real change and learning among teams is not and "add-on" to the current organizational structure but rather the establishment of a completely new one.

---

[1] The Emotional Intelligence is the ability within a group or individuals to manage feelings and emotions from the own individual and team, as well as from others, in order to obtain the optimal outcome or performance for any given task.

It is impossible to reach an utopian stage by only sewing patches to an old and dysfunctional structure.

What corporate strategies often lack is a team or leader that cascade them into the very roots of the organization, with tropicalizations [1] for each root, achieving a custom behavior of the teams an individuals with corporate alignment.

Within the automotive industry, specifically within Ford Motor Company, it is possible to see the enforcement of behavior fomenting team cooperation; where values of working as a team, and developing a global effort toward the creation of global products; resumes in strategy of merging global efforts in a single team to consolidate one global company.

But that strategy shown in the automotive industry means nothing if management doesn't cascade the values on it to the ground level, consolidating the strategy into tropicalized behaviors that individuals and teams can apply.

A point of special focus within growing organization that face roles overlapping from previous working structures to new ones, relies on the clear definition of roles and responsibilities, where new development teams are built with the resources that best fit from the previous structures. Losing expertise and knowledge, by little it is, is something not affordable by any product development organization that pretends to continue growing.

Leadership through development teams should be given to the entities whose interest best represent the interest of the groups (Hackman, 1998) and that fit the best for the task that in term of likelihood may belong to one or several teams, acknowledging that multidisciplinary interaction is the key path for the successful development of complex projects.

Within the development of the development activities, management must keep a clear watch over the conflict among teams and be aware of the extra payoffs being lost even though the work gets done. These problems must be acknowledged and addressed properly for resolution, adjusting the rules, over which teams interact in order to avoid

[1] Tropicalization is the term used when a vehicle design is modified or adapted for the local conditions of any given market where it will be sold, in order to make it optimal for the customer expectations and special conditions that must comply during its useful life.

having competition for payoffs within teams in the same organization, eliminating barriers and establishing an environment of support among them.

Management is one of the key elements to set the conditions for teams cooperation and successful interaction. When management decides not to act since the work is not done optimal but is still being completed, that unwillingness to intervene and solve the issue may create additional barriers that not-intentionally give support for teams to defect from cooperation since the message of there is no fear to defect is created.

Within teams and individuals in an organization, the existence of elements that live in a comfortable state where the assurance of a position is the only thing that matter. The need to locate these elements and pull them toward the path that provides learning and development to the organization as whole is a "must" requirement in order to build the emotional intelligence of the teams as a whole. In any team or organization it is easy to identify elements that are in a comfortable status with the only purpose of securing it, making anything necessary to maintain it, even a the expense of making the organization and other elements to loose their goals and target. This behavior described, is a driver for collective irrationality, which leadership must locate and dismantle in order to eliminate a block for teams to reach an optimal performance.

As described by Lafasto & Larson, often management teams have fear to act against conflicts and disagreements in order to not compromise the existent structure of their teams, that in one way or the other finish the job. The fear relies in not willing to enter into conflict with a senior manager, team, or individual, whose work is believed to be key, but rather obstacles the team to reach an optimal environment and payoff system. Based on this, when management is fearful to the consequence of acting against a well established structure, it becomes reluctant to meddle and act against environment where defection from cooperation is established as a condition that everyone sees comfortably.

As mentioned before, a strong leadership is key to break the prisoner dilemma within the development and testing teams of an organization, and it is also clear that a patch will not solve conflicts but rather the renegotiation of feeling and expectation of team players. Therefore, when cooperation issues reach a dangerous state of development, just an evolutionary of add-on fix will not serve but rather as a "Simple Apology" step. Here comes the point where leadership must apply a step function, a revolutionary change that shifts the current payoff systems and make the dominant strategies to change.

The type of actions required is drastic, and surely will create collateral damage to the organization structure, especially to the teams that found defection as a normal way of performing.

These drastic actions require establishing the order among conflict of well structured teams, by defining the new roles and responsibilities for each part. It is expected that the definition of new roles and responsibilities for a growing organizations, that need to move and manage teams to cover the needs of greater and more complex projects, will not be something that all teams agree or accept.

As mentioned by Hackman, revolutionary changes in organization will hurt its teams and people. People gets hurt during revolutionary changes, especially those who lead them, since they carry the responsibility for the change process consequences, even if they are successful and break the prisoner dilemma within an organization. (Hackman, 1998)

8.7.4.- Clear Roles, but Responsibilities Partnership

The completion of a program is the final goal of a PDO and development and testing team share the responsibility of delivering a system that correctly integrates and performs as customer expectations. Taking into account this common objective and responsibility, it is necessary to build partnership agreements, where teams agree to review results together being able to contribute equally to de development of projects.

Partnership among development teams is an environment in which there is no authority of one entity over the other, and where each team contributes to the delivery of the system from different ways of thinking as approaches, since by definition each team must not be able to do what the other does but be knowledgeable in a general sense in order to provide support, detect issues, and generate continuous improvement.

In software development, there is a partnership between tester and developer not working as a separate department but as a joint entity working hand by hand as a product evolves. In this environments the developer doesn't need to wait to send data and receive test results back since activities are done in parallel; and the tester is able to contribute to software development as its continuously contributes in real time with his insights and experience (Microsoft Anonymous 2008)

8.8. – CASE – Lack of Trust Increasing Development Testing Costs

Lack of trust is one of the main drivers for developing defection from cooperation strategies among development teams, which is often seen when one group doubts the accuracy of data in the results of other.

The introduction of a foreign vehicle to the market offers the advantage or quick delivery of a product to the market since it is only required to be validated against the local regulations and environment in which will operate. But the "build-up" programs has a heavy disadvantage since the importing country has limited and restricted design and development capability, making the validation phases a "home run" needed process.

During a growth step from a young product development organization in an emerging market, the need of a low cost small subcompact vehicle was evident and a full plan for quick introduction was carried away. A brief validation plan was undergoing, with high confidence on its success since the vehicle was low cost development on an existing product, already developed and launched in its home country.

During the validation of the vehicle, an issue was found with the results of power train cooling efficiency testing, which started great turmoil within the organization since the vehicle was supposed to be clean, ready, and expected to be introduced into the market.

Since the platform design was owned by another product development branch of the company, the local small organization lacked the expertise and size to defy them, which limited the weight of their voice, since the home market was not declaring any problem.

Due the circumstances and constrains for the development organization receiving the vehicle, combined with a lack of negotiated expectations and team information, the first and sustained response of the development team was that the testing was not properly done via facilities, equipment, and expertise; untrusting the validity of the results presented and requesting its repetition.

The event described triggered a loop of events, in which conflict was incorrectly addressed or ignored, on the basis that the development and testing work was still being completed over the apparent conflict that had raised among development entities. The "simple apology" cycle in combination of ignoring the situation developed within the years with reinforcing attitudes toward this behavior instead of renegotiating expectations and feeling, in the search of growth and learning.

As the product development organization grew and matured, the embedded conflict became something seen as normal within its members, where the root cause of it was long time forgotten but the behavior became part of the popular culture. Even affecting the performance of members of a development group when it was discovered that full information was shared constantly with other development groups, producing from calls of attention to lower performance grades.

The root cause of the problem was not clearly addressed and the lack of trust from testing results continued. The climax of this lack of trust issue was reached when a development team decided to send the vehicles to the corporate headquarters testing facilities within the United States, in order to repeat vehicle testing under controlled laboratory environments.

At the climax of this trust and cooperation defection cycle despite the cost of repeating a test within the local country capabilities, which could happen several times before results achieved a level of trust, the vehicle were sent to controlled laboratories in order to get trustable data.

The data obtained from the trustable facility had better fidelity, but matched the judgment of the vehicle performance against corporate requirements, resulting in a waste of resources for the program. As a indicative example, the testing cost at the controlled facility was higher that the original 1500 USD (indicative cost) per test at the local country, but adding a logistic and transportation cost of 3545 USD[1]. As a result, the defection loop among development teams can reach an increase of testing and validation cost above 8500 USD per program.

[1] Logistics and transportation cost estimate for a passenger vehicle one trip freight from Mexico City to the corporate headquarters testing facilities. Cost estimation from mid 2008.

The validation of the sub-compact low cost vehicle triggered cooperation defection among teams due a lack of trust reinforcing event, which started more than five years ago from the current model year. Although since its launch, engine overheating field warranty calls have been the main characteristic claim for this vehicle[1], the "self betrayal" consequences were not able to be removed creating rivalry among development teams, since all testing agreed to have an issue during validation, but lack of credibility among teams removed attention from the issue.

8.9. – CASE – Special Methodology Projects

In every organization measurement systems to asses the performance of their workers need to exists, systems that are conformed by objectives of different kinds in order to evaluate widely and objectively, the behaviors and delivery actions.

In a growing engineering organization the initiative to develop projects using special methodologies and approaches is strongly supported, and therefore it is expected that all the engineers deploy this methodology toward the development of more robust and reliable systems, to increase the quality of existing products, and to reduce the cost of ongoing programs.

In order to asses the methodology usage; the organization includes the delivery of at least one project every year per engineer. The deliverable has the particularity that if the engineer is assisted by other which due it involved is included within the same project, the project is only worthy of half for each one, and a third if another person is collaborating.

Over the years this measurable drives the engineering community to an isolation environment, where each one needs to develop projects by themselves or with as little help as possible from others, resulting on projects with relatively low value or that don't achieve the full potential that they could have achieved.

At first glance, testing requires engineers to work together when testing is directed to the development of a project, not necessary to be on the same area developing the task

[1] Top claim year after year since launch reported by the AWS corporate system.

together but in a work cell environment where the free flow of information enables the engineers to contribute to the development of the project with past experiences or expertise and best practices that could be applied from one project to other.

The question is simple. Is it worthy to have an organization that develops fewer projects per year, but high value projects where many engineers with different experiences and points of view are included? Or is it worthy to be able to have an organization with a huge number of projects developed?

An answer could be to have a deliverable that asks for strong and valuable involvement into projects, instead of delivering a specific number of them. Another answer is to keep working the same, and expect that projects will become more valuable with time.

Does current behavior drives the engineering community towards collective irrationality? From the testing standpoint through experience and observations without any implied judgment over the methodology, the answer could be "yes".

8.10. – Summary and Conclusion

When teams within a product development organization fall within a prisoner dilemma game, management must take quick and definitive action in order to avoid losing the optimum performance of their teams and protect the moral integrity of its human resources. Key points that management of product development should be aware are:

- Individual expectations, feelings, and goals have a major role in the team dynamics.
- Growing organizations face new tasks and responsibilities as they evolve, which need to be correctly addressed and assigned in order to avoid teams rivalry for them.
- Failed expectations due repetitive defection and conflict drives teams within a product development organization to a "Collective Irrationality" state.
- Management must be aware of teams falling in a prisoner dilemma, and be aware of the optimal payoffs that are being lost.

- Game environment should be played or depicted as unknown repetition games in order to avoid players to calculate when to defect.
- Punishment and reward systems must be established in order to modify teams payoffs and drive them to an optimal equilibrium point. And a delicate balance between their strength and credibility should be considered.
- Clear roles and responsibilities must be defined and constantly updated within product development organizations, ensuring that the knowledge and expertise of previous general structures is used to optimize the performance of the new ones as an organization grows.
- Rivalry is sane and desirable when it pushes people, teams, and organizations to continuously grow and perform better; but not at the expense of operation by handicapping them.

Chapter 9

=== Conclusions & Next Steps ===

"To measure is to know"

"If you can not measure it, you can not improve it"

– Lord Kelvin

9. – Conclusions

The intention is to briefly summarize and reflect from general perspective important aspects for the development and validation system of an emerging growing organization. Due the scope nature of this work, each chapter has specific conclusions for the themes covered which act as an extension for this integrative reflection.

Small product development organizations such as Ford of Mexico have the advantage of being fast and efficient, and as they grow and the complexity of the projects increases, a new working methodology is required for keeping this advantage while at the same time increasing capability and robustness.

Through academic and industry research and experience it has been clearly identified that most of the success of any Development and Validation phase within a product development system is strictly dependent of a correct definition phase. Through the creation of the system engineering culture implemented in early stages of every project, a framework can be followed toward the generation more efficient design verification plans, with higher levels of representativeness and design confidence.

These define phases include the analysis of requirements and development of targets, which must not define design solutions bur rather be representations of expected output linked to appeal models that correlate with the customer perception and expectation on the local region where the product will be sold.

The proper analysis of requirements for an accurate development that can be translated into alternative testing platforms requires a operational shift within the product development organization from a subjective based rating to data driven measurement process establishing metrics that promote the elaboration and correlation of models.

This requirement and targets should intend to be frozen in early stages of the development of a program, but never frozen within the process of a product development organization. Requirements and targets must be constantly evolving in order to survive the expectations shift over time within a Kano diagram, and keep always within the excitement and performance quality region, in order for the PDP to be value creating process able to always satisfy the customer shifting needs and expectation.

Value in a testing organization is the relationship between benefit and cost, which is linked to the functions and form of the testing organization. The investment and operation costs that facilities and equipment implies often make testing to look as a non-value added on the product development process, being a wrong approach since the function delivered represent the perceived attributes confidence that the final user will feel, as well as other functions that can be exploited from testing as marketing.

Although elements of form and function are similar between development and validation activities as well as their concepts, the analysis showed that the key differentiator toward both activities relies on the context in which they perform. A clearly defined context and boundaries are linked to clear roles and responsibilities, for the adequate management and performance of organizations.

Base on the evolution that have occurred within the Ford of Mexico test and development organization, and applying lessons learned from testing sectors within automotive and software companies with diverse maturity levels, a new concurrent working architecture was modeled in order to take advance of the small company characteristics present and establish the framework that will enable this characteristics to sustain as the product development organization grows in size. This system oriented and integrated thinking architecture is intended to be the pillar for maximizing the use of well defined representative prototypes, requiring less build stages for developing a vehicle over lean and data driven environments.

Another important element that any product development organization should handle across the path for maturity and projects of increased complexity is the management of risk. Engineering entities must be able to assess risk while elaborating their design verification plans, being able to build information thru data and engineering judgment, and decide which tests are really needed to be performed and for how long or with how much samples. In order to be competitive and effective in timely delivery to the market, organizations are not able to test everything and for all the time that would need, empathizing the importance of detecting what provides more value, and ensures in the same way that a reliable product is delivered to the customer.

Along the thesis structure, several testing and development principles are defined; as well as individual conclusion that complement the insights presented within this final chapter. But there is a principle of operation which I considerer to be the most important for the adequate operation of a mature mindset system focused attribute/function development and validation testing organization:

"Testing that is not understood with goals, requirements, and methods; not fully comprehended or agreed, should not be performed until clearer light shows what is best to be done"

9.1. – Future Work and Implementation

Ford of Mexico faces promising challenges to prove its value as a product development entity able to support the company as a global organization in these difficult times. But the path is not easy one, and along the way to reach engineering maturity the development and validation system currently operating may not be sufficient to support the needs of a growing product development entity.

The architecture described within this work needs to be implemented by restructuring the current one, moving the current resources where their current roles are joined with the responsibility of the group, where more value is created.

The change process will imply learning new frameworks, and replace old practices that have in operation since the organization started to this point (which have proven to be sufficient under previous local only context, small size, and low complexity managed). The new responsibilities and forward model for this product development organization will be the motivation and driver for change, but several resistance to change factors will need to be managed in order to the existing structures to fall under collective irrationality scenarios:

- Fragmentation focus rather than systems – Attempting to have the form of a developed and mature organization will make the tendency to divide tasks and functions into small isolated segments, an approach not suitable for a still small organization in comparison to well and mid-high developed ones, which will lead to "Little Empires" competing for the roles, responsibilities, and resources of projects that require systematic approaches with mutual and continuous cooperation. (Kreitner & Kinicki, 2007)
- Continue with reactive engineering approaches – Instead of having well developed plans, with proactive and creative problems solving methods that promote smart testing for every situation. Change is only accepted when there is no other option, and experience is something that is achieved when is already late. (Kreitner & Kinicki, 2007)

The organization must focus on the value of change and promote learning and cooperation under new working structures, promoting concurrent engineering work cells and punishing collective irrationality break out.

An IT support will be needed for the development of the requirements management system described in chapter four, which will be initially based on current expansion projects for the managing test request and documenting system using Microsoft Infopath™.

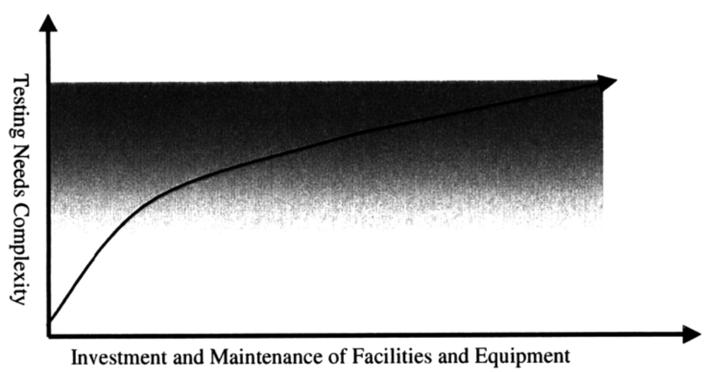

Figure 9-1: Testing Investment vs Needs Complexity

The last challenge to overcome will be the actual path to readiness for the D&V organization in order to be fully able to support the testing needs of a growing product development entity. Expertise can be obtained thru training and experience, which is required to form attribute development engineers, integrators, and planners. But equipment and facilities to support testing operation will require the completion of a careful strategy. As defined in previous chapters, the strategy to follow for the maximum readiness achievable for a testing organization is defined thru the following steps:

- Define a clear scope and goals for the programs to develop
- Identify the functional areas and attributes affected
- Identify items (equipment/facilities) to fill the capability gap
- Identify alternative test locations or prepare business case assessment
- Prioritize based on most urgent needs

Alternative testing platforms must be used to assist the verification and development of requirement assessing bench and virtual environments by the percentage of requirements being able to assessed by the tools as well as the percentage being able to be validated.

"No amount of experimentation can prove me right;
a single experiment can prove me wrong""
– Albert Einstein

This Page is Intentionally Left Blank

Bibliography & References

Aguirre, Adrian (2008)\, *Design of Product Development Systems*, Massachusetts Institute of Technology, Cambridge, MA.

Ancona, Debora & Bresman, Henrik (2007), *X–Teams: How to Build Team That Lead, Innovate, and Succeed*, Harvard Business School Press, Boston, MA

Ancona, Debora (2008), *Interviews and Insights* from the Seley Professor of Management at the MIT Sloan School of Management, and Faculty Director of the MIT Leadership Center.

Apostolakis, George (2007), *Engineering Risk Benefit Analysis*, Class notes from MIT Class ESD.72.

Arbinger Institute (2002), *Leadership and Self-Deception*, Berrett-Koehler Publishers Inc, San Francisco, CA

Attardi, Laura - Guida, Maurizio - Pulcini, Gianpaolo (2004), *A Mixed-Weibull Regression Model for the Analysis of Automotive Warranty Data*, Reliability Engineering and System Safety 87 (2005) 256 – 273.

Baldwin, Carliss Y. & and Clark, Kim B. (2000), *Design Rules*, MIT Press, Cambridge, MA

Vale, Carol A (1997), *Ford Automotive Operations Reliability Guide – Tool & Methods, Module 11*, Ford Motor Company, United States operations.

Clapper, Mark (2008), *Teachings and Insights* from the Vehicle NVH Technical Leader at Ford Motor Company, United States operations.

Clark, Kim B. & Fujimoto, Takahiro (1991), *Product Development Performance*, Harvard Business School Press, Boston, MA

Crawley, Edward (2008), *System Architecture*, Class notes and insights from MIT Engineering Systems Division class ESD34.

Davies, Michael A M (2007), *Technology Strategy*, Class notes and insights from Sloan MIT Class 15.905.

De Weck, Oliver & Haberfellner, Reinhard (2006), Systems Engineering, Springer, Draft Version

Department of Energy, *U.S. Retail Gasoline Historical Prices*, September 2008.
http://www.eia.doe.gov/oil_gas/petroleum/data_publications/wrgp/mogas_history.html

Dixit, Avinash & Skealth, Susan (2004), *Games of Strategy 2nd Edition*, Norton & Company Inc, New York, NY

Doğan, Pinar (2008), *Game Theory, Strategic Decisions, and Negotiations*, Class notes from Harvard JFK Class API–303.

Druskat, Vanessa U & Wolff, Steven B (2001), *Building the Emotional Intelligence of Groups*, Harvard Business Review, Boston, MA.

Ely, Robin (2008), *Leading Teams*, Class notes from Harvard HBS Class 2020.

Ford Motor Company, *Analytical Warranty System.*

Ford Anonymous (2008), *Interviews and observations* from product development teams at Ford North America Operations, Ford South America Operations, and Ford of Mexico.

Endo, Takahiro (2008), *Global Product Development: A Framework for Organizational Diagnosis*, Massachusetts Institute of Technology, Cambridge, MA.

Grady, Jeffrey O. (2006), *System Requirements Analysis*, Academic Press as an imprint of Elsevier, Burlington, MA

Grady, Jeffrey O. (2007), *System Verification*, Academic Press as an imprint of Elsevier, Burlington, MA

Hackman, Richard (1998), *Why Teams Don't Work, Theory and Research on Small Groups*, Plenum Press, New York, NY.

Halliday, Peter (2008), *Teachings and Insights* from the Climate Control NVH Leader at Ford Motor Company, United States operations.

Harvard Business Review (1995), *The Product Development Challenge – Competing Through Speed, Quality, and Creativity*, Harvard Business School Press, Boston, MA

Hotta, Mark-Tami (2008), *Interviews and Insights* from the World Wide Proving Grounds Chief Engineer at Ford Motor Company, United States operations.

Kanaga, Kim & Prestridge, Sonya (2002), *How to Launch a Team*, Center for Creative Leadership, Greensboro, NC

Kleyner, Andre (2005), *Determining Optimal Reliability Targets Through Analysis of Product Validation Cost and Field Warranty Data*, PhD dissertation at the University of Maryland.

Kleyner, Andre & Sandborn, Peter (2007), *Minimizing Life Cycle Cost by Managing Product Reliability Via Validation Plan and Warranty Return Cost*, International Journal of Production Economics 112 (2008) 796 – 807.

Kochar, Nand & Holmes, Eric (2008), *Interviews and Insights* from the CAE Management at Ford Motor Company, United States operations.

Kreitner, Robert &. Kinicki, Angelo (2007), *Organizational Behaviour*, Mc Graw Hill.

Krivtsov, Vasily – Frankstein, Michael – Horstmann, Ulrich (2003), *Reliability Analysis of Field Failure Data via Analytical Warranty System*, Unpublished reference to appear in Ford Technical Journal in March 2003.

Krivtsov, Vasily (2008), *Interviews and Insights* from the Global Quality Office Technical Leader at Ford Motor Company, Mexico operations.

Lafasto, Frank & Carl, Larson (2001), *When Teams Work Best*, Sage Publications, New York, NY

Levenson, Nancy (2008), *Software Engineering Concepts*, Class notes and insights from MIT class 16.355.

Lopez, Jannet (2008), *Interviews and Insights* from the CAE Manager at Ford Motor Company, Mexico operations.

Majeske, Karl D (2003), *A Mixture Model for Automotive Warranty Data*, Reliability Engineering and System Safety 81 (2003) 71 – 77

Megalhaes, Marcelo (2008), *Interviews and Insights* from the CAE Manager at Ford Motor Company, South America operations.

Microsoft Anonymous (2008), *Interviews and insights* from product development teams at Microsoft Corporation, United States Operations.

Murman et al (2002), *Lean Enterprise Value*, Palgrave, New York, NY

Rechtin E. & Maier M.W. (2000), *The Art of Systems Architecting*, CRC Press, Boca Raton, FL.

Repenning, Nelson P. (2001), *Understanding Firefighting in New Product Development*, MIT – Sloan School of Management, Cambridge, MA

Repenning, Nelson P & Black, Laura J, *Why Firefighting is Never Enough: Preserving High Quality in Product Development*, MIT – Sloan School of Management, Cambridge, MA

Robbins, Stephen P & Judge, Timothy A (2008), *Organizational Behavior,* Prentice Hall

Sobek, D. K. II (1997*). Principles that Shape Product Development Systems: A Toyota – Chrysler Comparison*, UMI, Ann Harbor, MI.

Thomke, Stefan (2003), *Experimentation Matters – Unlocking the Potential of New Technologies for Innovation*, Harvard Business School Press, Boston, MA

Torralba, Antonio (2008), *MIT Helps to Create Image Recognition Software*, MIT Tech Talk, Volume 54, Number 27, Cambridge, MA

Wheelwright, Steven C. & Clark, Kim B. (1992), *Revolutionizing Product Development: Quantum Leaps in Speed, Efficiency, and Quality*, Free Press, New York, NY

Whitney, Daniel et al (2004), *The Influence of Architecture in Engineering Systems*, MIT ESD, Boston, MA

Xie, M. Tang, Y. Goh, T.N. (2002), *A Modified Weibull Extension with Bathtub – Shaped Failure Rate Function*, Reliability Engineering and System Safety 76 (2002) 279 – 285

Zambrano, Jose & Sandoval, Miguel (2008), *Interviews and Insights* from the Test & Development (LVT & Homologation) Manager / Ford Andina Cars Brand Manager at Ford Motor Company, Venezuela operations.

Appendix

A. – Development & Validation System Process DSM Charts

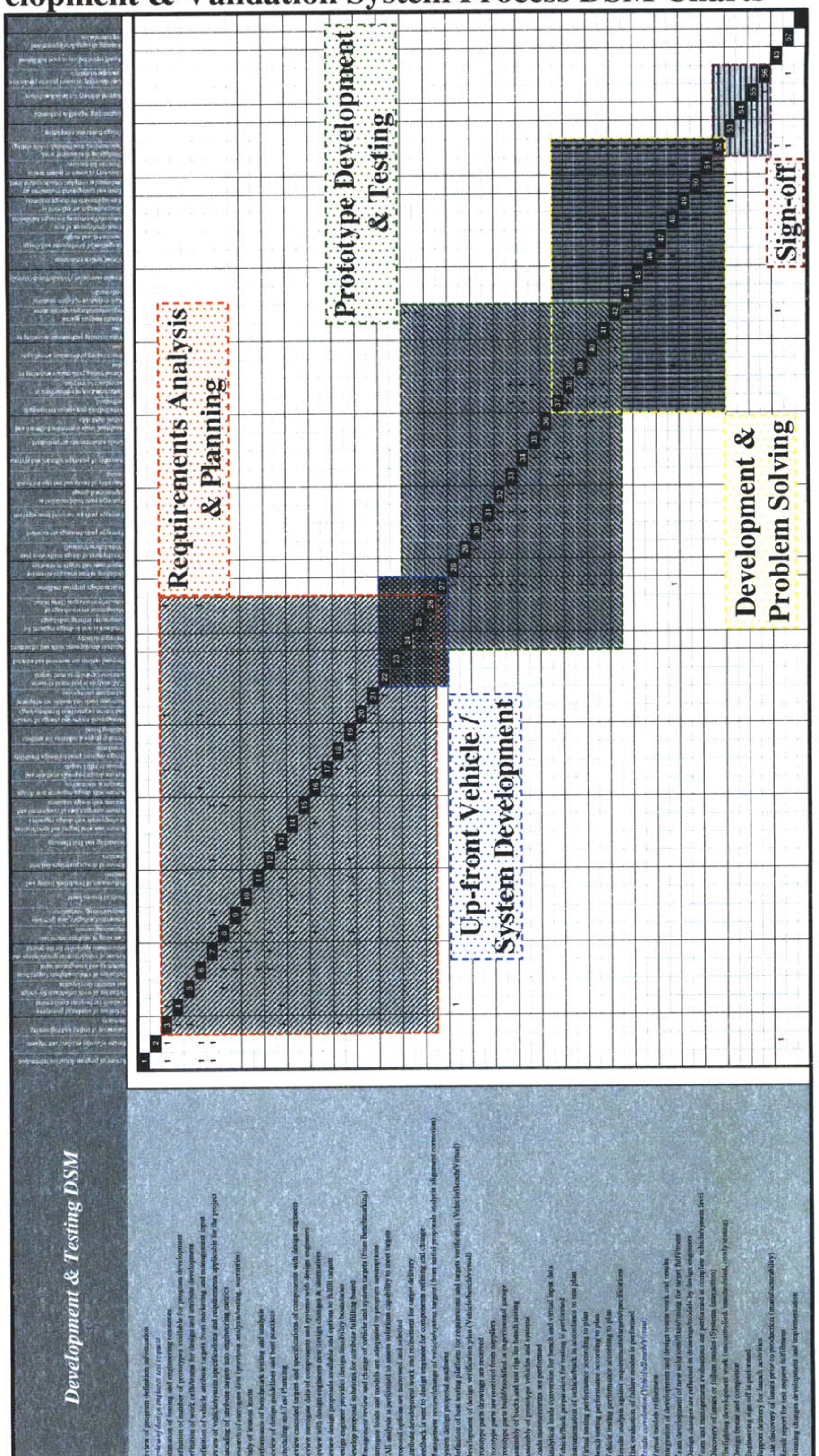

Figure A-1: Development & Validation System Partitioned DSM

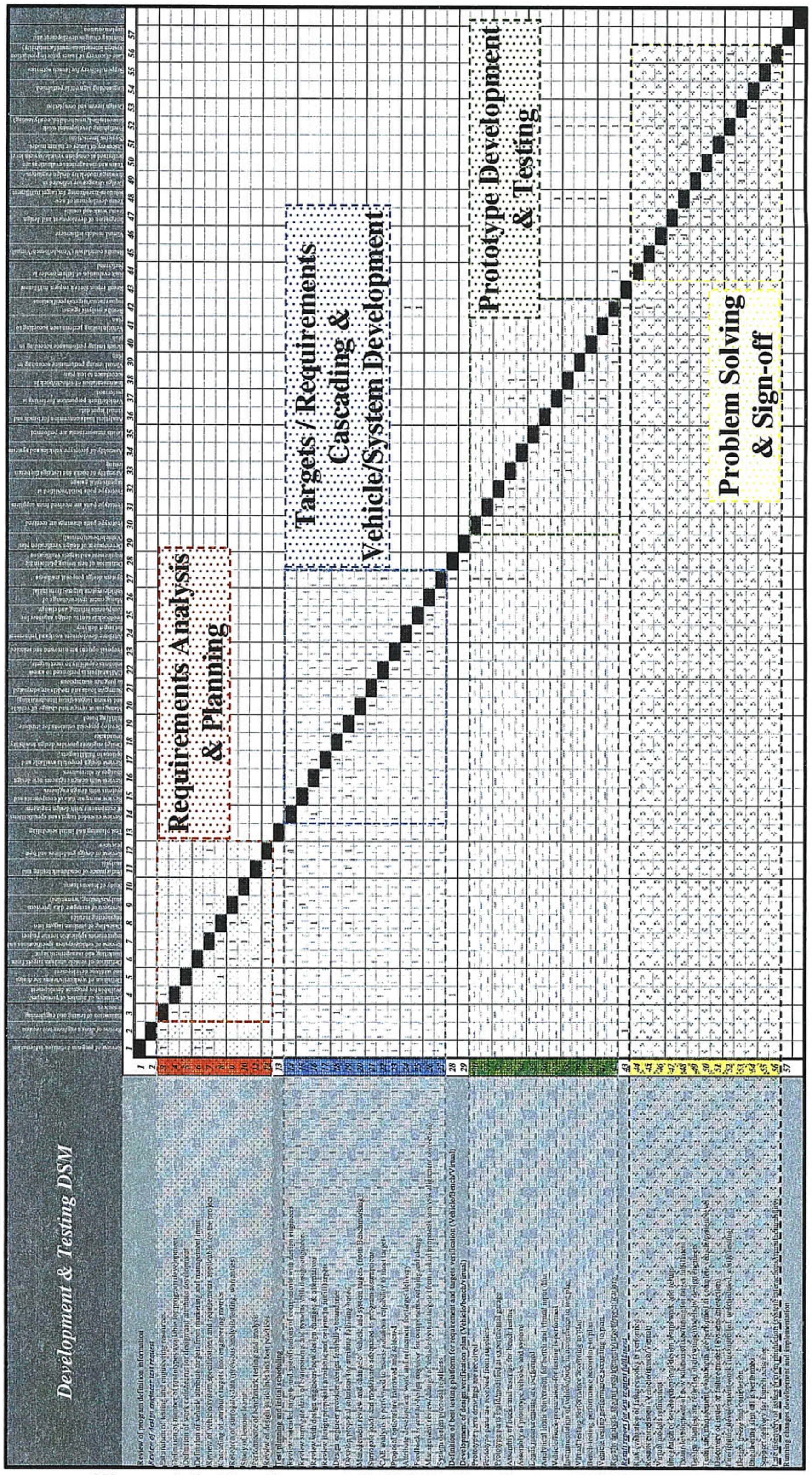

Figure A-2: Development & Validation System DSM

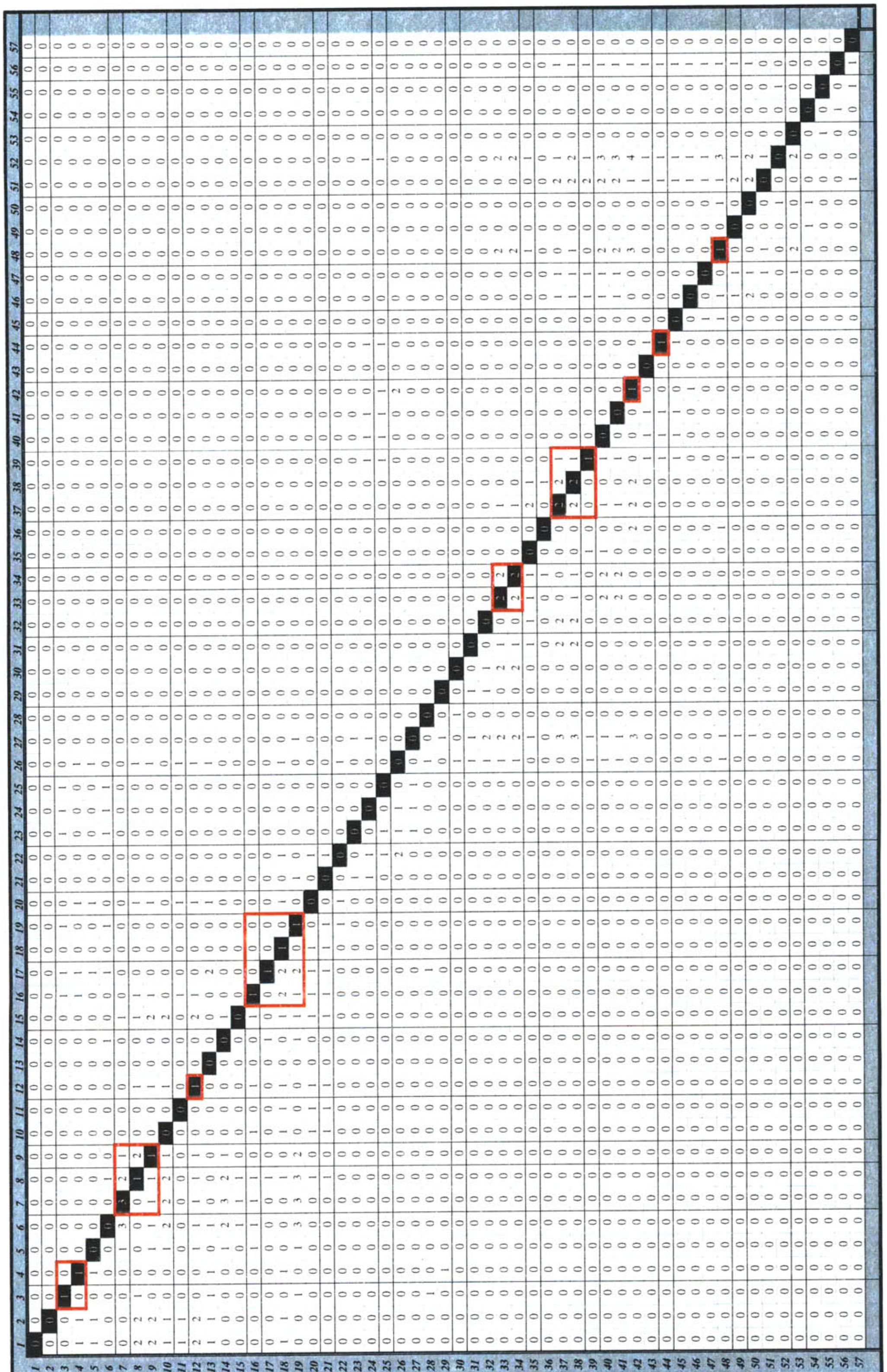

Figure A-3: Development & Validation System DSM Binary^2 Matrix

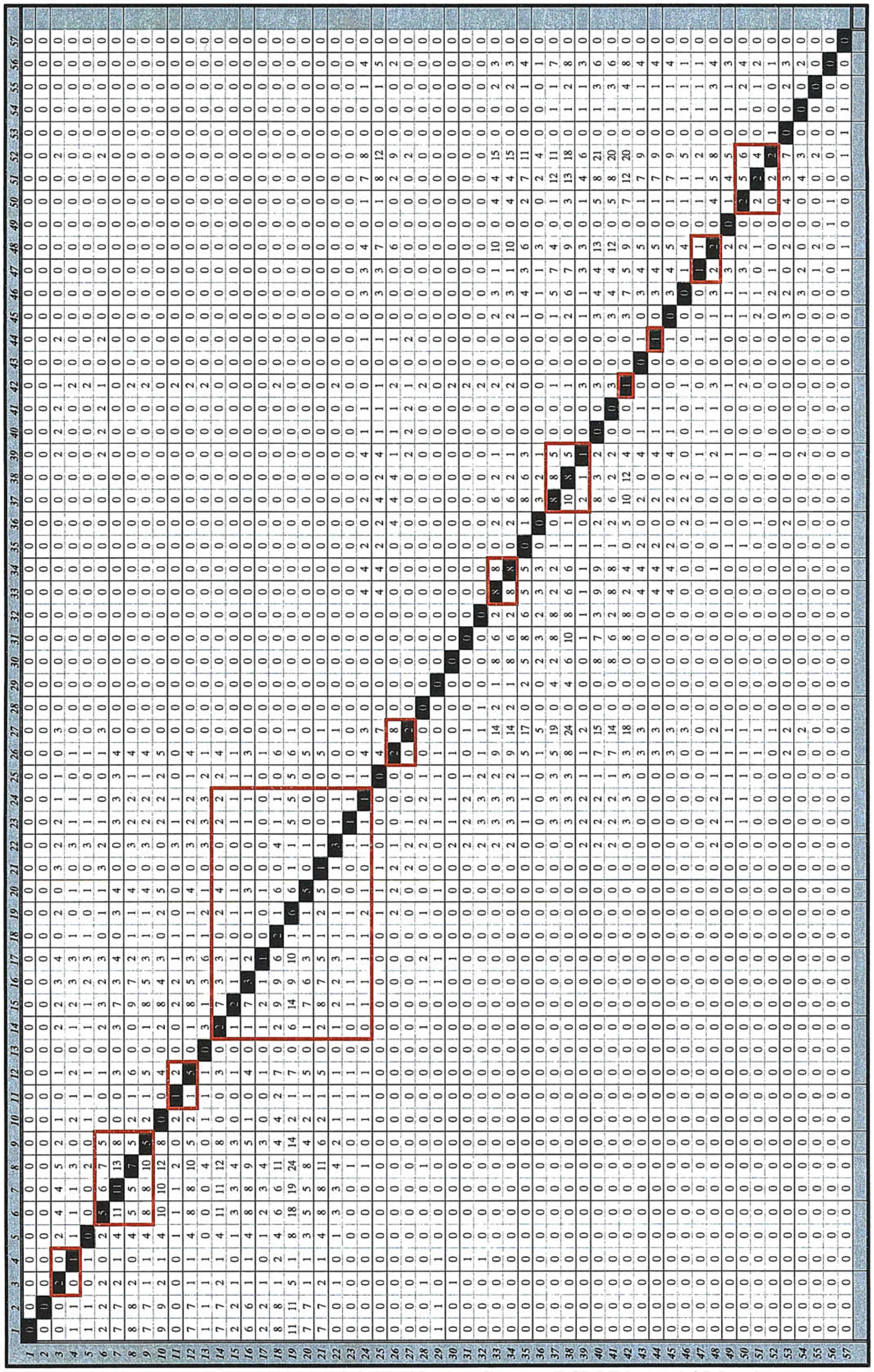

Figure A-4: Development & Validation System DSM Binary^4 Matrix

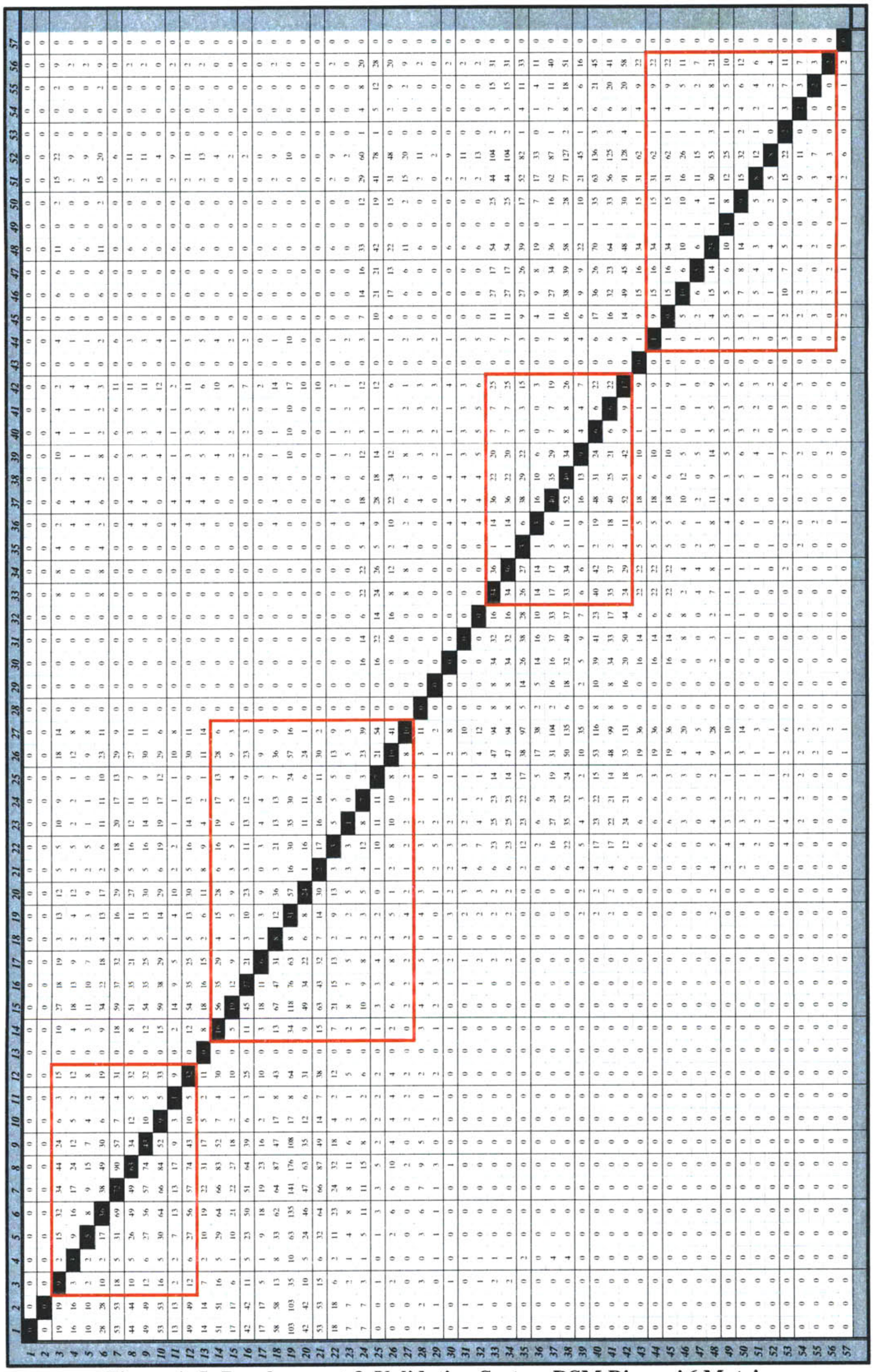

Figure A-5: Development & Validation System DSM Binary^6 Matrix

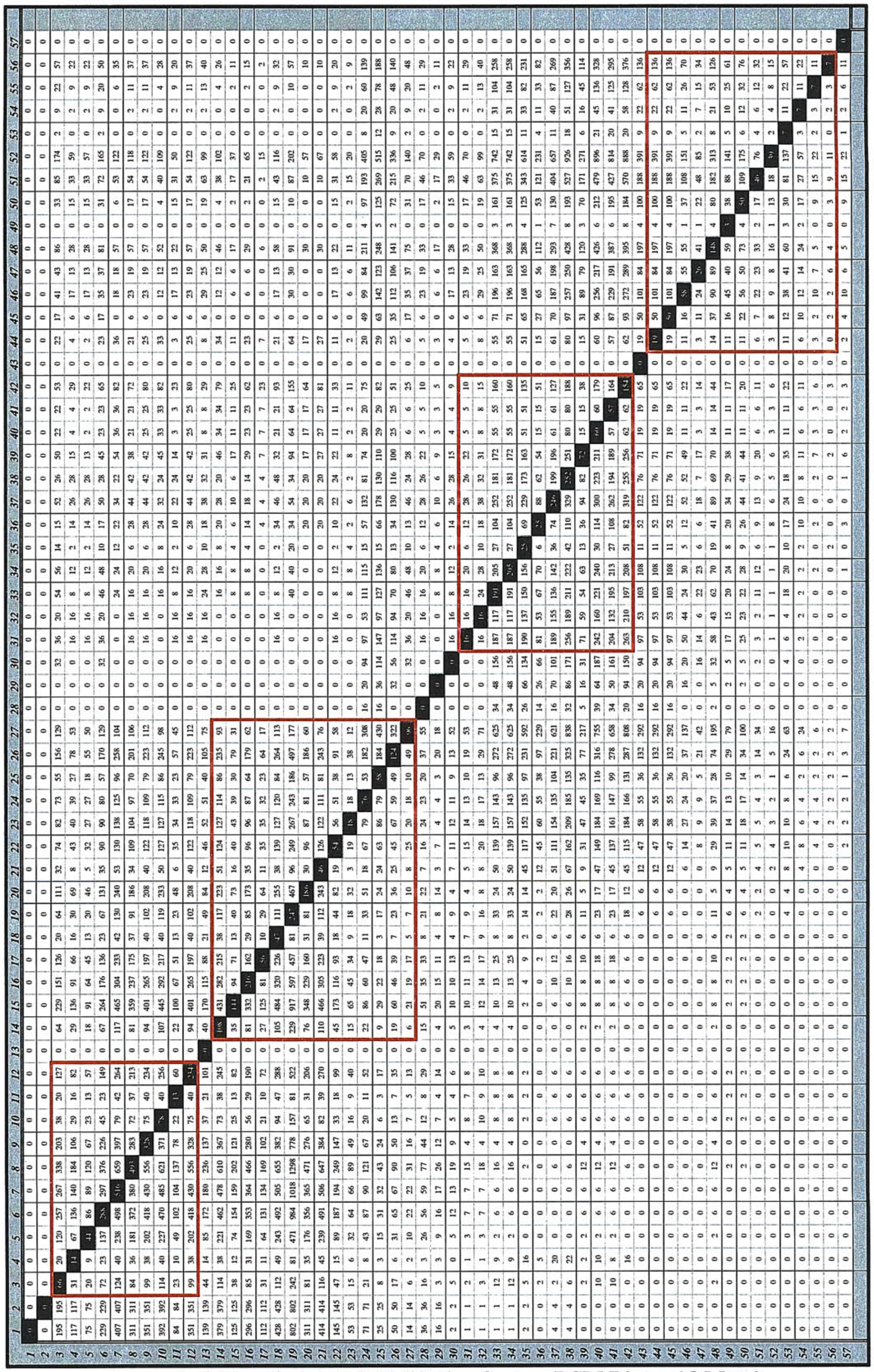

Figure A-6: Development & Validation System DSM Binary^8 Matrix

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1554	1554	468	160	914	1907	1971	2524	1503	314	163	1020	0	447	1783	1172	900	163	493	922	219	603	663	608	452	1300	1135	32	56	240	280	170	348	363	72	192	444	291	372	167	167	461	0	167	129	324	292	682	9	269	672	1396	22	57	174	467	0
4	886	886	245	97	507	1041	1070	1380	814	184	93	585	0	232	1001	663	483	93	248	528	109	315	325	298	220	631	514	0	32	56	130	110	86	100	19	86	198	182	168	76	76	229	0	76	41	135	125	234	2	89	281	532	9	22	59	189	0
5	590	590	160	66	338	688	706	914	534	128	64	395	0	151	669	447	321	64	163	355	74	219	229	211	156	452	452	0	32	56	130	110	78	92	15	72	180	168	139	53	53	167	0	53	41	129	119	205	2	87	260	467	9	22	57	174	0
6	1753	1753	515	183	1027	2131	2198	2823	1670	365	189	1161	0	491	2012	1324	1002	189	552	1047	237	666	716	659	489	1401	1143	32	56	208	260	166	318	335	68	211	446	313	377	177	177	497	0	177	118	312	274	670	9	255	658	1377	20	50	165	458	0
7	3081	3081	913	320	1807	3760	3877	4969	2951	639	335	2038	0	861	3541	2337	1770	335	935	1812	393	1072	1080	984	726	2043	1104	0	0	96	172	124	274	296	90	221	420	310	445	257	257	712	0	257	75	251	241	630	0	179	587	1313	6	35	122	399	0
8	2341	2341	668	254	1357	2811	2892	3728	2197	494	257	1555	0	638	2675	1768	1324	257	690	1389	292	826	833	763	562	1594	1046	0	32	88	214	178	204	232	59	201	406	350	377	194	194	567	0	194	76	257	237	531	2	177	564	1153	11	37	118	381	0
9	2631	2631	757	280	1528	3162	3256	4191	2477	553	288	1746	0	717	3009	1990	1486	288	781	1560	329	922	931	852	629	1777	1119	0	32	88	214	178	220	248	67	217	430	366	414	219	219	628	0	219	76	269	249	571	2	181	600	1238	11	37	122	405	0
10	2921	2921	854	307	1708	3546	3653	4692	2777	615	323	1941	0	806	3357	2216	1669	323	880	1722	361	1014	1013	927	682	1917	1014	0	0	64	152	120	228	252	78	216	390	308	405	234	234	670	0	234	64	227	211	566	0	161	533	1185	4	28	109	362	0
11	657	657	183	72	376	774	796	1027	606	134	67	431	0	176	740	491	361	67	189	395	90	237	250	227	170	489	444	0	32	56	114	94	78	88	17	68	168	150	146	65	65	177	0	65	35	118	112	200	2	74	246	458	9	20	50	165	0
12	2631	2631	757	280	1528	3162	3256	4191	2477	553	288	1746	0	717	3009	1990	1486	288	781	1560	329	922	931	852	629	1777	1119	0	32	88	214	178	220	248	67	217	430	366	414	219	219	628	0	219	76	269	249	571	2	181	600	1238	11	37	122	405	0
13	1126	1126	329	120	659	1364	1404	1797	1065	232	117	743	0	309	1287	855	639	117	323	661	152	418	436	396	291	842	727	0	32	120	202	150	178	196	39	108	286	226	241	105	105	308	0	105	75	205	199	378	2	151	419	821	13	40	99	283	0
14	2852	2852	841	297	1670	3472	3580	4593	2725	594	312	1889	0	794	3277	2161	1634	312	868	1681	358	982	990	904	669	1873	975	0	0	64	136	104	228	248	80	204	370	282	400	234	234	647	0	234	58	216	204	549	0	148	515	1148	4	26	102	349	0
15	947	947	280	99	556	1158	1193	1528	906	196	102	626	0	265	1068	717	544	102	288	557	122	329	332	302	223	629	339	0	0	32	52	36	86	92	28	67	128	92	137	80	80	219	0	80	23	76	74	195	0	54	179	405	2	11	37	122	0
16	2186	2186	636	229	1272	2639	2720	3493	2073	453	237	1445	0	600	2501	1648	1236	237	654	1290	268	746	743	679	500	1410	692	0	0	32	84	68	142	156	52	153	258	206	283	174	174	490	0	174	35	140	130	378	0	94	344	795	2	15	65	235	0
17	802	802	229	85	466	965	992	1272	752	166	85	530	0	216	915	600	446	85	237	473	96	268	266	243	179	500	216	0	0	0	16	16	32	36	14	52	76	66	91	62	62	174	0	62	6	35	31	109	0	21	94	235	0	2	15	65	0
18	3130	3130	878	339	1805	3720	3825	4928	2906	660	340	2079	0	829	3564	2351	1731	340	892	1850	358	1072	1065	981	717	2034	1156	0	32	56	210	190	174	208	53	244	440	410	414	231	231	722	0	231	71	256	226	554	2	176	574	1222	9	32	116	391	0
19	6067	6067	1787	633	3548	7381	7610	9746	5794	1241	644	3996	0	1688	6947	4571	3454	644	1818	3561	757	2067	2086	1900	1403	3954	1989	0	0	160	276	196	470	504	158	401	734	540	813	493	493	1365	0	493	121	421	407	1113	0	293	1010	2333	10	57	202	690	0
20	2244	2244	633	242	1298	2679	2755	3548	2092	476	247	1494	0	597	2563	1688	1248	247	644	1322	249	757	740	683	497	1403	642	0	0	0	80	80	88	108	34	158	242	228	246	155	155	493	0	155	30	121	101	320	0	87	293	690	0	10	57	202	0
21	3046	3046	875	323	1769	3663	3773	4846	2870	633	328	2016	0	826	3480	2285	1705	328	891	1789	345	1006	1007	926	683	1900	819	0	0	0	80	80	128	148	54	192	296	262	340	219	219	648	0	219	30	151	131	411	0	97	380	892	0	10	67	259	0
22	1163	1163	340	122	678	1405	1450	1859	1104	235	119	763	0	324	1325	872	656	119	351	689	151	300	421	383	292	807	559	0	32	56	114	94	110	120	33	88	204	170	211	106	106	270	0	106	35	142	136	262	2	82	312	595	9	20	58	207	0
23	428	428	122	47	249	513	528	678	400	89	44	283	0	116	487	324	240	44	119	253	54	151	142	129	91	292	172	0	0	32	36	20	46	48	10	33	66	44	55	33	33	106	0	33	17	35	37	105	0	31	80	207	2	9	20	58	0
24	573	573	191	82	344	718	740	963	559	140	80	400	0	167	684	464	379	80	226	380	188	573	651	598	416	1191	2168	94	150	505	667	441	628	685	94	361	872	648	626	198	198	591	0	198	295	736	629	1273	20	636	1394	2676	60	139	405	966	0
25	198	198	84	57	127	269	278	380	208	73	51	163	0	66	268	191	179	51	143	171	182	556	711	653	469	1224	2816	114	244	623	894	631	787	853	139	412	1131	861	835	229	229	650	0	229	371	979	887	1574	28	791	1899	3377	78	188	515	1293	0
26	414	414	130	76	253	525	544	720	410	114	68	302	0	127	509	349	279	68	175	271	149	424	577	528	404	976	2052	56	208	394	655	505	529	563	125	243	820	641	638	168	168	460	0	168	247	683	694	1056	20	497	1410	2317	48	140	336	940	0
27	149	149	44	17	90	182	188	253	140	42	23	109	0	46	180	127	99	23	68	107	45	149	183	173	124	404	750	32	56	208	244	150	246	259	34	125	316	213	192	51	51	168	0	51	112	247	213	470	9	224	477	940	20	48	140	336	0
28	357	357	114	37	215	453	468	597	355	72	37	234	0	110	414	278	221	37	116	211	68	154	184	164	127	344	468	[illegible]	32	88	150	114	124	136	27	53	186	146	157	54	54	127	0	54	52	153	149	233	2	105	302	525	11	29	70	199	0
29	131	131	37	16	77	158	162	215	122	36	21	95	0	35	157	110	83	21	37	79	16	68	60	57	37	127	149	0	[illegible]	32	52	36	54	60	12	27	72	52	43	10	10	54	0	10	23	52	50	103	0	46	103	199	2	11	29	70	0
30	86	86	33	9	57	121	126	167	94	25	15	64	0	34	111	81	72	15	44	58	32	61	91	82	69	161	377	0	32	56	130	110	86	100	19	46	158	142	122	30	30	61	0	30	41	135	125	174	2	89	261	406	9	22	59	169	0
31	80	80	26	11	50	103	106	147	78	27	16	62	0	29	101	80	67	16	38	60	36	77	91	81	62	193	436	0	32	88	150	114	124	136	27	53	186	146	137	34	34	81	0	34	52	153	149	233	2	105	302	505	11	29	70	199	0
32	96	96	37	13	65	137	141	197	103	39	25	81	0	40	132	109	97	25	54	74	51	113	123	110	81	255	584	0	32	120	202	150	178	196	39	80	258	198	180	44	44	115	0	44	75	205	199	336	2	151	405	704	13	40	99	269	0
33	91	91	95	67	76	176	180	299	128	95	82	131	0	64	194	193	244	82	240	194	333	865	1011	915	643	1734	4214	156	304	909	1289	893	1220	1324	227	688	1746	1317	1244	365	365	996	0	365	531	1408	1269	2473	31	1141	2771	5133	104	258	742	1892	0
34	91	91	95	67	76	176	180	299	128	95	82	131	0	64	194	193	244	82	240	194	333	865	1011	915	643	1734	4214	156	304	909	1289	893	1220	1324	227	688	1746	1317	1244	365	365	996	0	365	531	1408	1269	2473	31	1141	2771	5133	104	258	742	1892	0
35	30	30	47	84	22	54	54	126	36	65	61	71	0	23	89	108	147	61	181	141	301	755	925	828	597	1487	3761	134	327	749	1135	829	1014	1083	214	558	1532	1168	1144	348	348	862	0	348	453	1209	1150	2072	33	935	2464	4385	82	231	614	1668	0
36	5	5	18	31	2	6	6	22	4	14	14	14	0	4	16	23	41	14	61	50	117	301	366	328	231	597	1463	66	134	327	443	306	402	427	69	214	581	433	423	135	135	348	0	135	168	453	413	796	11	354	902	1668	33	82	231	614	0
37	62	62	46	96	44	100	100	195	68	87	77	105	0	34	139	155	177	77	186	155	311	758	948	844	623	1505	3936	101	344	711	1204	926	1035	1109	257	547	1616	1254	1270	370	370	860	0	370	491	1323	1315	2135	40	990	2736	4633	87	269	657	1836	0
38	64	64	72	114	50	118	118	239	80	109	99	127	0	44	167	189	237	99	263	208	422	1069	1304	1169	844	2128	5397	171	445	1055	1637	1204	1480	1590	331	804	2215	1690	1690	497	497	1230	0	497	678	1814	1738	3035	51	1405	3686	6469	127	356	926	2493	0
39	66	66	30	26	43	93	95	135	70	28	20	56	0	27	91	77	75	20	59	69	103	254	337	302	225	574	1492	31	130	293	482	365	389	425	87	196	601	477	477	126	126	316	0	126	199	541	528	806	16	407	1074	1799	45	114	271	723	0
40	74	74	84	82	55	129	131	229	94	78	70	106	0	47	153	155	207	70	235	195	376	987	1203	1087	771	2041	5035	187	402	1082	1537	1076	1415	1529	271	784	2045	1548	1513	441	441	1183	0	441	643	1710	1572	2899	45	1367	3378	6128	136	328	896	2306	0
41	72	72	79	66	55	129	131	227	94	76	68	104	0	47	151	151	198	68	221	181	331	870	1051	952	674	1810	4443	161	336	948	1347	939	1265	1373	246	715	1816	1375	1350	390	390	1048	0	390	578	1542	1407	2611	41	1242	3035	5514	125	295	814	2075	0
42	58	58	44	116	36	84	84	159	60	69	63	87	0	24	117	127	147	63	178	155	368	934	1226	1095	808	1978	5157	150	473	1020	1604	1199	1364	1446	318	703	2087	1589	1598	452	452	1131	0	452	684	1725	1736	2831	58	1341	3557	6118	128	376	888	2411	0
43	0	0	22	20	6	18	18	36	12	18	18	18	0	6	24	24	46	18	65	47	119	373	460	424	292	824	1974	94	150	475	618	407	585	637	90	327	797	589	571	160	160	466	[illegible]	160	281	708	608	1183	22	610	1338	2498	62	136	391	927	0
44	0	0	22	20	6	18	18	36	12	18	18	18	0	6	24	24	46	18	65	47	119	373	460	424	292	824	1974	94	150	475	618	407	585	637	90	327	797	589	571	160	160	466	0	160	281	708	608	1183	22	610	1338	2498	62	136	391	927	0
45	0	0	22	20	6	18	18	36	12	18	18	18	0	6	24	24	46	18	65	47	119	373	460	424	292	824	1974	94	150	475	618	407	585	637	90	327	797	589	571	160	160	466	0	160	281	708	608	1183	22	610	1338	2498	62	136	391	927	0
46	0	0	0	16	0	0	0	6	0	6	6	6	0	0	6	6	6	6	20	14	51	124	194	174	137	308	860	20	94	152	267	213	209	221	60	99	338	269	284	71	71	174	0	71	110	[illegible]	326	448	11	217	643	1014	26	70	151	430	0
47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	8	18	56	68	63	42	153	344	16	20	96	101	56	115	121	20	69	141	90	99	28	28	85	0	28	67	134	121	250	7	133	261	517	15	34	85	190	0
48	64	64	27	13	43	93	95	135	70	28	20	56	0	27	91	75	70	20	57	65	87	233	305	277	203	578	1455	32	101	307	467	335	408	449	84	236	602	464	456	116	116	310	0	116	237	568	530	886	21	482	1102	1941	53	126	313	762	0
49	14	14	8	4	12	28	28	43	20	13	11	19	0	8	29	27	27	11	22	19	33	92	118	108	79	219	578	5	32	103	178	134	169	189	49	93	242	188	214	50	50	130	0	50	99	261	257	377	[illegible]	210	516	849	25	61	141	352	0
50	14	14	8	4	12	28	28	43	20	13	11	19	0	8	29	27	27	11	22	19	41	110	148	134	100	261	731	5	48	123	229	179	194	220	55	113	300	245	269	64	64	158	0	64	123	328	320	448	12	[illegible]	642	1039	32	76	175	437	0
51	0	0	2	0	2	6	6	12	4	6	6	6	0	2	8	8	10	6	11	5	11	41	51	48	34	100	253	2	5	48	73	50	85	94	26	55	113	81	93	20	20	64	0	20	56	123	123	205	6	115	248	437	12	32	76	175	0
52	0	0	0	0	0	0	0	2	0	2	2	2	0	0	2	2	2	2	6	4	5	15	22	21	16	36	106	0	2	5	25	23	22	30	9	28	44	43	51	14	14	26	0	14	24	66	57	78	4	59	124	[illegible]	8	15	39	87	0
53	0	0	4	0	4	12	12	24	8	12	12	12	0	4	16	16	20	12	22	10	22	74	93	87	63	179	464	4	10	80	132	94	148	165	46	104	208	155	169	37	37	114	0	37	101	222	220	369	11	208	448	789	[illegible]	57	137	316	0
54	0	0	0	0	0	0	0	4	0	4	4	4	0	0	4	4	4	4	12	8	10	22	32	30	24	63	179	0	4	10	42	38	42	52	17	46	78	70	82	22	22	37	0	22	38	101	90	132	7	88	197	316	11	[illegible]	57	137	0
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	8	10	10	8	6	24	63	0	0	4	6	4	18	20	10	17	24	18	35	11	11	22	0	11	12	38	41	60	3	30	81	137	7	11	[illegible]	57	0
56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	8	6	4	2	6	24	0	0	0	2	2	2	2	2	10	10	8	11	6	6	11	0	6	10	12	14	24	2	17	27	57	3	7	11	[illegible]	0
57	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	4	9	11	10	7	18	42	0	0	2	3	2	11	14	8	11	13	11	29	9	9	20	0	9	11	34	36	42	2	29	70	109	6	11	22	50	[illegible]

Figure A-7: Development & Validation System DSM Binary^10 Matrix

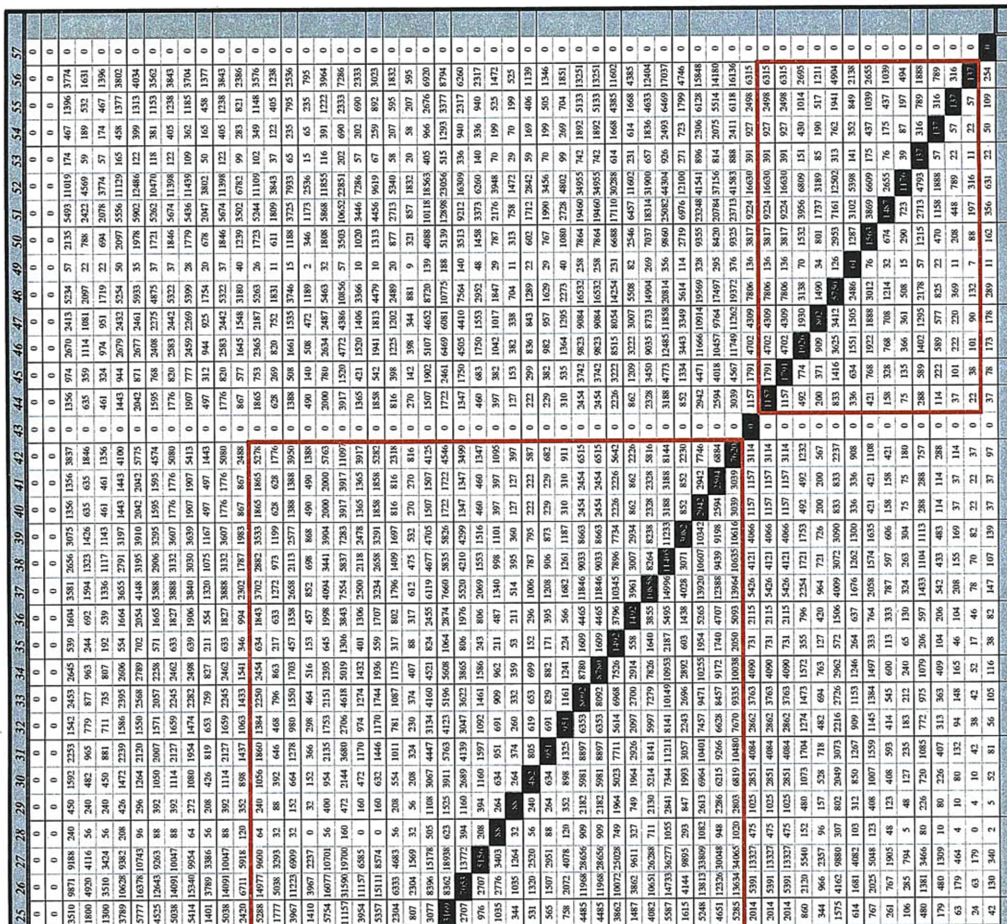

Figure A-8: Development & Validation System DSM Binary^12 Matrix