System Dynamics approach to understand the role of Information Technology in the evolution of Next Generation Integrated Product Development Systems

by

Sashi K. Somavarapu
M.S. Computer Engineering (1997)
Wayne State University

B.E. Electronics Engineering (1995)
Bangalore University

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 2005

© 2005 Sashi K. Somavarapu. All rights reserved

The author hereby grants to MIT permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole or in part.

Signature of Author ______________________________
Sashi K. Somavarapu
System Design and Management Program
February 2005

Certified by ______________________________
Nelson P. Repenning
J. Spencer Standish Associate Professor of Management
Thesis Supervisor

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
FEB 15 2005
LIBRARIES

BARKER

System Dynamics approach to understand the role of Information Technology in the evolution of Next Generation Integrated Product Development Systems

by

Sashi K. Somavarapu

Submitted to the System Design and Management Program on February 10, 2005 in Partial Fulfillment of the Requirements for the Degree of Master of Science in Engineering and Management

ABSTRACT

For the automotive industry, to be competitive in the market place has to devise many strategies. Some of the prominent strategies include and are not limited to, reduction of development costs by moving in-house work to its suppliers, reduction of PD cycle time by mimicking some of the industry's successful PD processes. Some companies are also devising some complex strategies like Zero-Prototype development using computer aided prototyping and testing, currently prevalent in the aerospace and naval industries, and more recently making a move into the Lean PD systems and processes to avoid waste and increase efficiency.

However, to introduce such lean PD systems, with reduced PD cycle time, into a complex organization with many internally developed IT systems, processes and tools is a huge challenge. The organization needs to adapt to these lean environments not just structurally but also culturally. To design a lean PD organization (system) the decision makers have to foresee and understand how the system of systems may react to the change before they are implemented and/or executed.

In the past couple of decades IT systems have been a primary enabler for PD work flow processes. However, IT systems are so engraved in some PD organizations that they have turned into an engineering process mechanism. Also, some of the IT systems have served more than their life expectancy and in some cases cannot be decommissioned because these systems are so tightly coupled with the business processes.

An understanding of the internal system dynamics of these deeply engraved IT systems in the PD life cycle will help the automotive industry executives (decision makers) and IT systems architects to make the right decision when designing and deploying the new PD systems or processes.

This study provides an overview of how IT tools have evolved in the automotive industry. Extensive research was conducted to understand the different system dynamics tools used in industry – specifically in automotive product development and the software development areas. The study concludes with an explanation of how system dynamics tools can be used as a program planning and management tool.

Thesis Supervisor: Professor Nelson Repenning
Title: J. Spencer Standish Associate Professor of Management

(THIS PAGE IS LEFT INTENTIONALLY BLANK)

TABLE OF CONTENTS

Chapter 1 Introduction *13*

Background **13**

Product Development Process (PDP) **14**

The Problem **16**

Objective **17**

Thesis Structure and Scope **18**

Thesis Organization 20

Chapter 2 Systems Under Review: Product Development & Infromation Technology *22*

The Product Development Firm under Review **22**

Trends in Automotive Industry **23**

Information Technology under Review **28**

Information Technology in Product Development **28**

IT Tool Adoption 29

IT Customer 31

Chapter 3 System Analysis & Mental Models *34*

Hypotheses **34**

Interview Process **39**

Understand Identified PD Behaviors through Causal Loop Diagrams **46**

IT Tools Growth: 46

IT Tools Growth in Test Sector: 54

Possible Interventions **60**

Chapter 4 Overview of System Dynamics Tools *62*

Brief History **62**

System Dynamics in Product Development **63**

Dynamic Program Simulator 63

Software Development Model 71

Chapter 5 Case Study: Simulating the System Dynamic Models *74*

Approach **74**

Structure Analysis: 74

Explanation of the key leverage variables: 77

Current State of the System under Analysis **78**

Case Study 1: Accelerate rework discovery **79**
Results of simulation for case study 1 81
Case Study 2: Increased availability of early design prototypes **84**
Results of simulation for case study 2 85
Possible Interventions **88**
Chapter 6 Recomendations ***90***
Chapter 7 Conclusions ***94***

LIST OF FIGURES

Figure 1: Generic Product Development Process (PDP) AKA Systems Engineering "V"...............14
Figure 2: Systems thinkers' pyramid...............16
Figure 3: Thesis Approach...............18
Figure 4: Porter's Five Principles on Automotive Sector (Source: *http:// www.investopedia.com*) .24
Figure 5: Automotive market share for years 2000-2003, by country of origin (Source: *Wards Auto*)...............26
Figure 6: System dynamics Model Depicting the IT Tool Adoption in PD (*Based on Vensim molecules*)...............29
Figure 7: Typical IT Tool Adoption in PD (Source: *Vensim Model Simulation*)...............30
Figure 8: No. of Critical IT Tools Used in Different Phases of PD...............35
Figure 9: Increase in data storage needs - An industry prediction...............37
Figure 10: PD - IT Tools Customer Favorability Rating (Source: *PD Tools Survey*)...............42
Figure 11: CAE methods used during PD for design validation and verification...............44
Figure 12: PD Process Change Loop – B1...............46
Figure 13: Tools Growth Loop – R1 (Follow the bolded arrows)...............47
Figure 14: Home Grown IT Solutions Loop – R2...............49
Figure 15: Shadow IT Evolution Loop – R3...............50
Figure 16: Deviations from Standards Loop – R4...............51
Figure 17: IT Tool Growth - Derived Causal Loop Diagram (CLD) from Interviews & Qualitative Data...............53
Figure 18: Normal Physical Testing Loop - B1 & Normal CAE Testing Loop - B2...............55
Figure 19: Growth in Physical Testing IT Tools...............57
Figure 20: Growth in CAE Testing Tools...............58
Figure 21: IT Tools Growth in Test Sector - Derived Physical and CAE Test Sector Dynamics.....59
Figure 22: Phased Sub-systems (Source: *Sterman & Ford 1998*)...............63
Figure 23: Project Network (Source: *Sterman & Ford 1998*)...............64
Figure 24: Traditional Project Management Work Accomplishment Stock Flow Structure (Source: *ESD.36J Course Lecture Notes from James Lyneis (Lyneis, 2002)*)...............65
Figure 25: Typical usage of program management simulators (Source: *DPS – PA Consulting*).....68
Figure 26: Software Development Model Structure (Source: *Abdel-Hamid & Madnick*)...............71
Figure 27: Effect of Accelerate Rework Discovery on PD Efficiency...............75
Figure 28: Effect of Availability of Early Design Prototypes on Program Metrics...............76
Figure 29: Total program staff - Case Study 1...............83
Figure 30: Engineering Quality for all 3 Design Phases - Case Study 1...............84
Figure 31: Total program staff - Case Study 2...............87
Figure 32: Engineering Quality - Case Study 2...............88

LIST OF TABLES

Table 1: Sample interview questions 40
Table 2: Key interview takeaways 41
Table 3: Counter Balancing Testing methods 55
Table 4: Reference Program Definition (Source: *Program Types - Edward Esker, 2000 SDM Thesis*) 69
Table 5: Simulated results: With & without inspections during software development 72
Table 6: Current state simulation - Required input parameters 78
Table 7: Case Study 1 – Optimized key leverage variable values (*See bolded values*) 81
Table 8: Summary of program timing - Case Study 1 82
Table 9: Summary of Costs - Case Study 1 82
Table 10: Case Study 2 – Optimized key leverage variable values (*See bolded values*) 85
Table 11: Summary of program timing - Case Study 2 86
Table 12: Summary of Costs - Case Study 2 86

ACKNOWLEDGMENTS

I would like to take this opportunity to thank all the people who have made an impact in my academic career at MIT. This was one of my most memorable experiences that I will cherish for a long time. I once read a statement in a business magazine that read like this - "It is students who make a school a 'great' school" and this is absolutely true with MIT. Of course, no school exists without faculty, and MIT's faculty has clearly demonstrated that they are out of the ordinary.

I would like to first and foremost thank Marv Adams, CIO & Senior VP of Ford Motor Company for his unstinting support, mentorship, guidance and encouragement. His valuable insight, sponsorship and time have made this Thesis a possibility. Without his direction and help; I would not have achieved what I have achieved.

I am grateful to my thesis advisor, Nelson Repenning, for his guidance and coaching throughout the thesis process. I also admire his patience and perseverance for making sure that I did not lose track of my end deliverable. His system dynamics class, at MIT - Sloan School of Management, was the initial motivating factor that led me to pursue this thesis on system dynamics.

I am also thankful to my Ford Motor Company management - Nick Smither, William Houghton, Roger Mitchell, Jeremy Seligman, Tom Daigneau and David Cantegallo for their support during the course of my MIT SDM program. I am very thankful to my supervisor, Peggy Gessner, who has gone above and beyond her call of duty to adjust the workload around me and helping me finish my MIT program. I would like to express my greatest appreciation to my department manager, Steffen Mueller-Urbaniak, for indulging himself in my thesis and for his continued support through out the thesis process.

I could not have accomplished this study without the help of Don Sutherland of Ford Motor Company and Bill Dalton of PA Consulting; their great insight and experience with the Dynamic Program Simulator (DPS), a system dynamics tool, helped me achieve the amount of analysis that would not have been possible otherwise.

I would also like to thank everyone who has come forward and help me in numerous ways during my academic career at MIT especially my class mates – John Chacko, Nisheeth Singh, Badari Kammandur, Frank Lanni, and numerous others who have enlighten me with their valuable guidance and business insights.

Last but not the least; I would like to offer my deepest thanks to my wife, Rekha, for her incredible support and patience throughout this two year SDM program at MIT.

(THIS PAGE IS LEFT INTENTIONALLY BLANK)

CHAPTER 1
INTRODUCTION

Background

Automotive Product Development (PD) firms inherently are in a constant churn; churn due to organizational, technological, and environmental (competitive) changes. All these changes have brought about new ways of delivering products to the market place. Firms want to develop products faster, cheaper and with greater quality. The trend in U.S. in the past decade has been to expand their market reach (share) by adding newly acquired brands to their product portfolio. Take for example; Ford bought Jaguar, Volvo and Land Rover. Daimler took over Chrysler. GM bought Saab, parts of Subaru, Fiat and Daewoo. However, leaving the merger mania behind, a few fiercely independent companies like Toyota and Honda emerged bigger and stronger than others.

To be competitive in the market place, the automotive firms have devised numerous strategies. Some of the prominent strategies include and are not limited to, reduction of development costs by moving in-house work to its suppliers, reduction of PD cycle time by mimicking some of the industry's successful PD processes. Some companies have also undertaken some complex initiatives like Zero-Prototype development using computer aided prototyping and testing, which are currently prevalent in the aerospace and naval industries, and more recently making a move into the lean and integrated PD systems and processes to avoid wastage and improve efficiency. All these strategies face considerable limitations due to the culture within the organization, resource capabilities, and lack of good processes & tools.

For instance, to introduce a lean integrated PD system, with reduced PD cycle time, into a complex organization with many internally developed information technology (IT) tools and processes takes a lot of time and effort. The organization needs to adapt to the lean environments not just structurally but also culturally. To design such a lean PD organization (system) the decision makers have to foresee and understand how the system of systems will react to the change before they are implemented and/or executed.

Product Development Process (PDP)

At a very high level PDP is a set of well defined steps taken to design, develop, build, test and validate a product based on the targets set by marketing, these targets are based on 4 Ps of marketing: product, placement, pricing & promotion and with considerable input from the customer (customer preferences in that particular market segment). Marketing also sets the product quantity and desired quality specifications which PD has to work with its intended suppliers and engineers to meet these objectives. Once designs are validated via physical testing or by much faster analytical testing; also known as CAE (Computer aided engineering), these designs have to be then released for production. Production is either carried out by the supplier who is supplying the component or by the manufacturing division of the company.

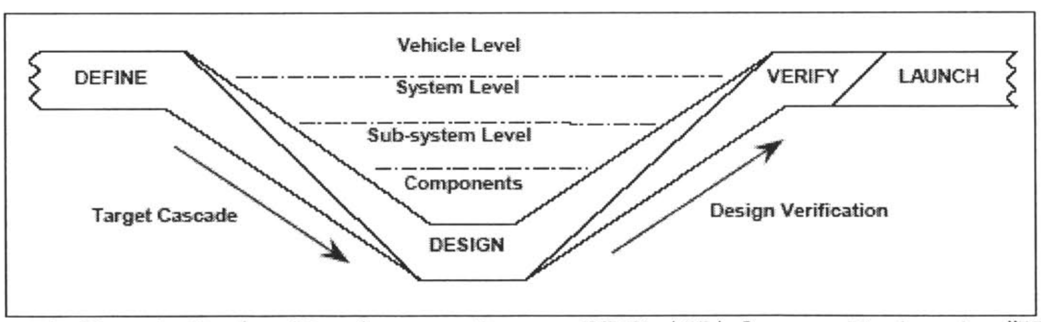


Figure 1: Generic Product Development Process (PDP) AKA Systems Engineering "V"

We have seen that products like cars and airplanes have been on the face of this earth for about 100 years now. Yet the firms that produce these products face complex issues and problems; such as complex organizational structures, product strategies, competitor pressures, personnel problems, competency issues, technology shifts, micro & macro economics and many such dynamics. The complexity of any firm can be assessed by the sheer nature of the product portfolio it offers or by the number of business entities that it manages and interacts with to develop a product.

Communication was one of the main issues that big organizations have faced in the past and even face today. Many new PD processes have been developed over the decades and still there are some voids left. Some organizations have realized the issues caused due to lack of proper communication and devised organizational structures that enable effective communication. However, there are inherent organizational structures that make it impossible to break the barriers of communication. Turf protection within an organization prevents people in these firms to exchange lessons learnt that help in making an adaptive enterprise.

A systems thinker can quickly understand the complex structure and the patterns within any organization. These complex structures are like roads and bridges of a town. Once formed it is very hard to replace them. The pattern within the organization is like the traffic that flows within these structures (Roads, alleys etc.). And accidents or road jams are "events" that occur due to these patterns. A good systems thinker can digest a complex organization by laying out a framework for this organization through tools like system dynamics and identify the key events that are taking place from a set of patterns and their inherent structures. Many systems thinkers believe that events are only the tip of the "Iceberg" (refer to Figure 2). They believe that the decision makers or planners have to study the patterns within the organization to go for the (structural problems) root causes.

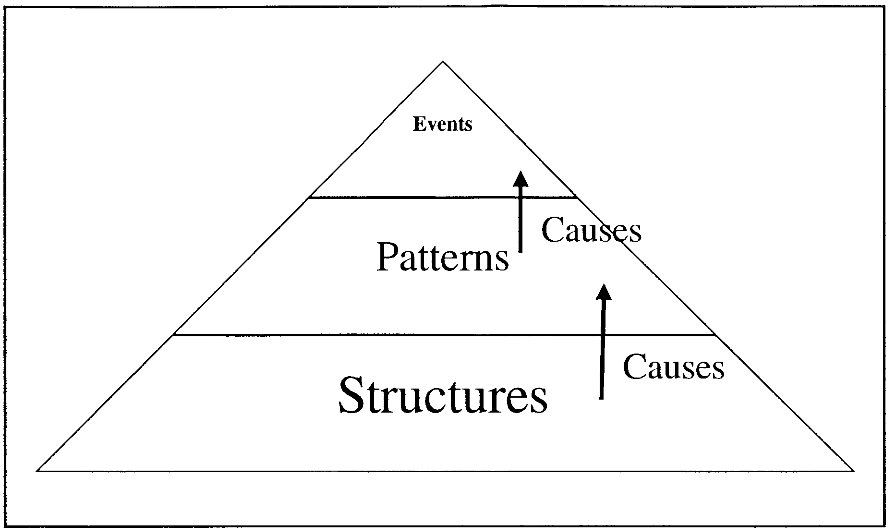


Figure 2: Systems thinkers' pyramid

Many of the examples I will be citing in this study are based on my experiences and the information I have received from the key management personnel of Ford Motor Company (My current employer). Ford Motor Company's confidential material is not shared in this study. However, the company's traditional PD behaviors are shared (I feel that these behaviors are common to other US based companies across the nation), and that is the basis of this study.

The Problem

In the past couple of decades IT tools have been a primary enabler for PD work flow processes. However, in some complex organizations, the IT tools are so embedded into the PD processes that they themselves have turned into an engineering process mechanism rather than just being a tool. Also, some of the IT tools have served more than their life expectancy and cannot be decommissioned because they are interconnected with other Enterprise IT systems.

In the company under consideration, a New PD system (NPDS) is being developed under the assumption that PD can deliver the key enablers for its success. However, the NPDS depends on IT for part of their success and the rest on process changes. Past PD systems have only delivered partial success in the implementation of the desired process and tool enablers. During the interview process with the PD engineers and process specialists, it was determined that the past PD systems have brought in many success stories, but they are still shy of their competition in terms of development costs, timing, quality & technology.

Currently there is no structured way to estimate or validate if an enabler (specifically IT enablers) can assist PD to reduce the development cycle time, reduce costs, improve quality, and increase engineering efficiency.

An understanding of the internal system dynamics of these deeply engraved IT systems in the PD life cycle will help the automotive industry executives and IT systems architects to make the right decision before they implement and/or execute any enabler into the new PD systems or processes; thereby enabling the organization to turn into a more agile and adaptive system.

Objective

The objective of this study is to gain an understanding of the role of IT in PD Systems as an enabler in the automotive organizations using system dynamics as an analysis tool. This will help us understand how IT can affect the entire product development processes, planning and life cycle.

The primary research objectives are:

1. Understand the system dynamics associated with PD systems and the associated IT process & tools through system dynamics modeling.
2. Analyze the factors affecting the firm's long and short-term decision-making process.

3. Analyze the factors affecting the firm's ability to deliver - like the organizational structure, operational procedures, tools quality and availability, staffing level, training & communication

4. Formulate recommended changes to the organization's management decision-making process and product development processes.

Thesis Structure and Scope

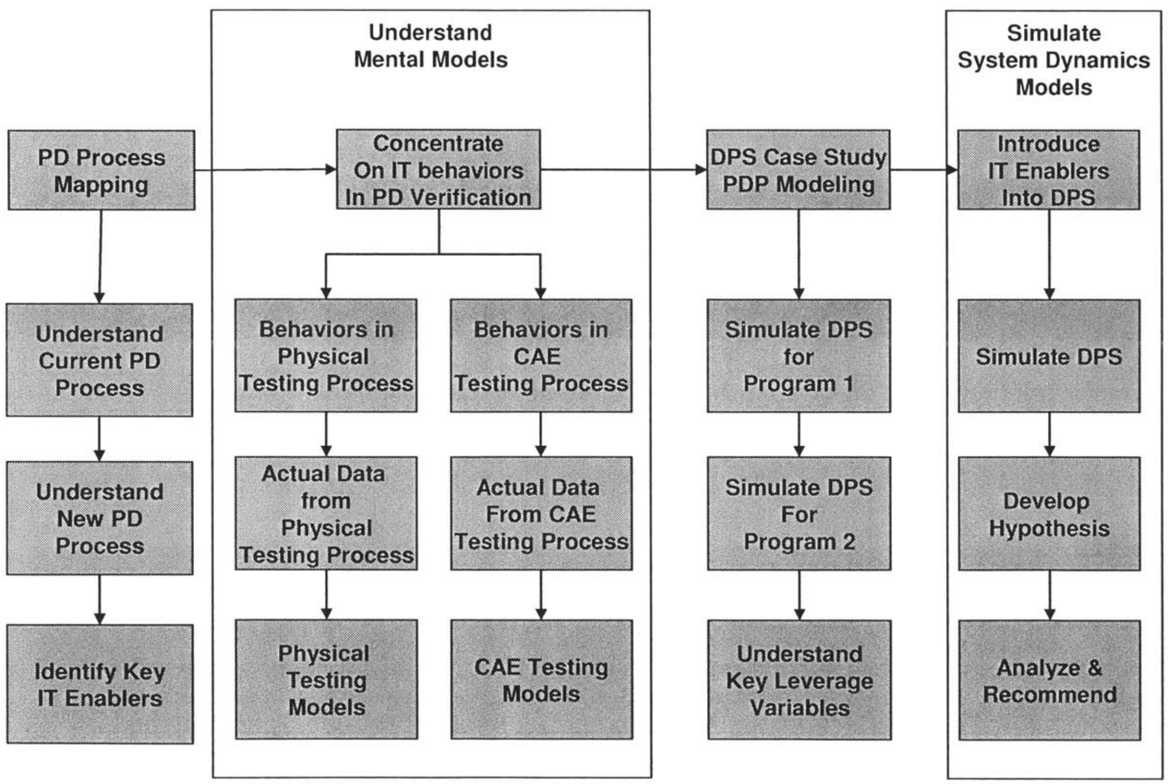


Figure 3: Thesis Approach

The foundation for this thesis was laid by developing a process map as shown in Figure 3. I have referenced the corporate product development process documents, corporate PD web site, and PD pocket cards etc. to better understand the current PD system (PDS).

An effort was made to understand the new PD system (NPDS) as much as possible. However, for the scope of the thesis I would only restrict myself to discussing couple of key IT enablers that would affect the NPDS.

Many PD and IT subject matter experts (SMEs) were interviewed through well defined question and answer sessions to understand the behaviors of PD engineers and IT tool development trends in the PD arena. Mental models were created based on the information sessions and by analyzing the qualitative customer satisfaction survey data. These mental models will help us understand the state of IT systems and the perceived quality and productivity issues related to PD-IT tools.

Dynamics Program Simulator (DPS), a Project Management Flight Simulator (system dynamics based) tool developed by PA Consulting was identified during research for system dynamics tools used in PD practice. DPS is currently used in PD vehicle program planning. Current behavioral aspects of the DPS were studied and simulated to identify the prominent or key leverage variables associated with key IT enablers in the PDS. To retain the confidentiality of DPS, the internal workings of the tool has not been documented in this study. Permission was obtained to publish some of the high level working structure of DPS.

After further investigation I have found that intensive system dynamics work was done in the area of software development. I was also able to secure the model that was developed for Tarik Abdel-Hamid project to understand the software development practices.

Several simulation runs were performed on an enhanced software system dynamics model, developed by John Tvedt (John Tvedt PhD Thesis dissertation can be found in http://www.eas.asu.edu/~sdm/tvedt/dissertation.) His thesis is based on extended version of the Abdel-Hamid & Madnick model that included inspections similar to Stage Gate or Gate reviews.

Thesis Organization

Chapter 1: This chapter gives a brief background of the problem under review, the objective and scope of this study.

Chapter 2: A brief overview of the Systems under review: PD and IT. Also, a brief history of how Systems Thinking and system dynamics are being used in automotive industry, with specific discussion around system dynamics tools used in PD and in IT software development.

Chapter 3: A hypothesis was developed based on the information gathered during the interview and data collection process. A brief explanation of the mental models developed in the area of PD and product verification.

Chapter 4: A brief history of Systems thinking and system dynamics in automotive & software industry. Overview of the system dynamics tools identified in PD and in software development.

Chapter 5: Case studies based on the simulations run on DPS.

Chapter 6: Recommendations.

Chapter 7: Conclusions and closing comments.

(THIS PAGE IS LEFT INTENTIONALLY BLANK)

CHAPTER 2
SYSTEMS UNDER REVIEW: PRODUCT DEVELOPMENT & INFROMATION TECHNOLOGY

The Product Development Firm under Review

The issues identified in this study refer to the product development activities in Ford Motor Company. The dynamics of this organization are representative of many other complex product development organizations and therefore are presented in detail in this study. These dynamics may not be of small or newly formed firms or firms that were formed less than a decade ago adhering to the lean principles in PD. Even though large corporations with decades of knowledge have adapted the lean PD principles; the inherent organizational structures and behaviors have limited these companies from adapting to change quickly.

Ford Motor Company has been in the business of making automobiles for over a century. The company has seen lot of radical changes in its many years of operation. The processes and methods that were developed by Henry Ford are still being followed by the company. The incentive and financial system that existed in 1920s is still followed to its core and that might have contributed to lot of management behaviors that now exist as a major organizational barrier for change.

When Henry Ford started the development of the automobile, all the parts that made the car were designed, developed and manufactured within the company. Alfred Sloan, then CEO of GM decentralized parts-making divisions as independent profit centers to make specific categories of parts for the whole company[i]; Ford Motor Company also took similar steps. By the 1950s, Henry Ford II introduced bidding process of securing parts from independent supplier firms. The suppliers were given detailed drawings of the parts and were asked to bid for an annual contract. The lowest bidder would generally get the contract.

This activity exists even today however the method of executing this activity differs. Computers have changed the way designs are developed and transferred to the suppliers. Today, Tier one and Tier two suppliers get the designs either as CAD or CAM drawings. To take full advantage of IT infrastructure and the experience within the company, new ways of managing this supply chain is being followed. Industry experts want to standardize this process with an over arching strategy know as PLM (Product life cycle management).

Trends in Automotive Industry

The current landscape of the auto companies is very different from what it was 10 years ago. American companies have started using the manufacturing principles taught by their Japanese counterparts. The automotive supply chains have streamlined to the standards dictated by Toyotas and the Hondas. However, in the past decade the big three US automakers have lost considerable market share and the emergence of Korean automakers has brought in a new threat to both Japanese and US automakers.

Let us analyze some of the challenges specifically faced by US automotive sector[ii] in terms of Porter's five forces – New entrants, Suppliers, Buyers, Substitutes and competition.

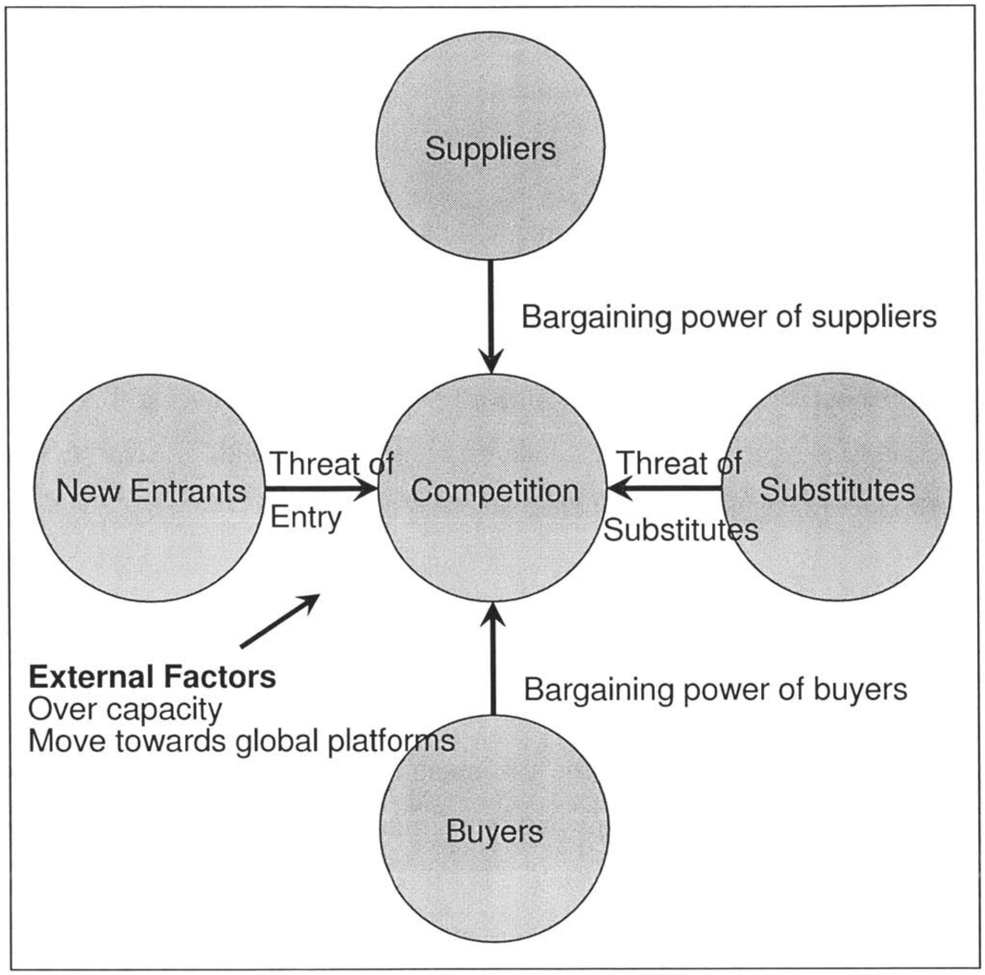


Figure 4: Porter's Five Principles on Automotive Sector
(Source: *http:// www.investopedia.com*)

1. **Threat of New Entrants:** Establishment of a new automotive company may be really expensive taking the current market conditions. However, this did not hold true when Honda Motor Co. opened its first plant in Ohio. Korean automaker Hyundai also made this threat felt by opening a new plant in Montgomery, Alabama and R&D center in Irvine, California.

2. **Power of Suppliers:** Many suppliers rely on one or two automakers to buy a majority of their products. Automotive components like tires, shocks, exhaust systems etc have become

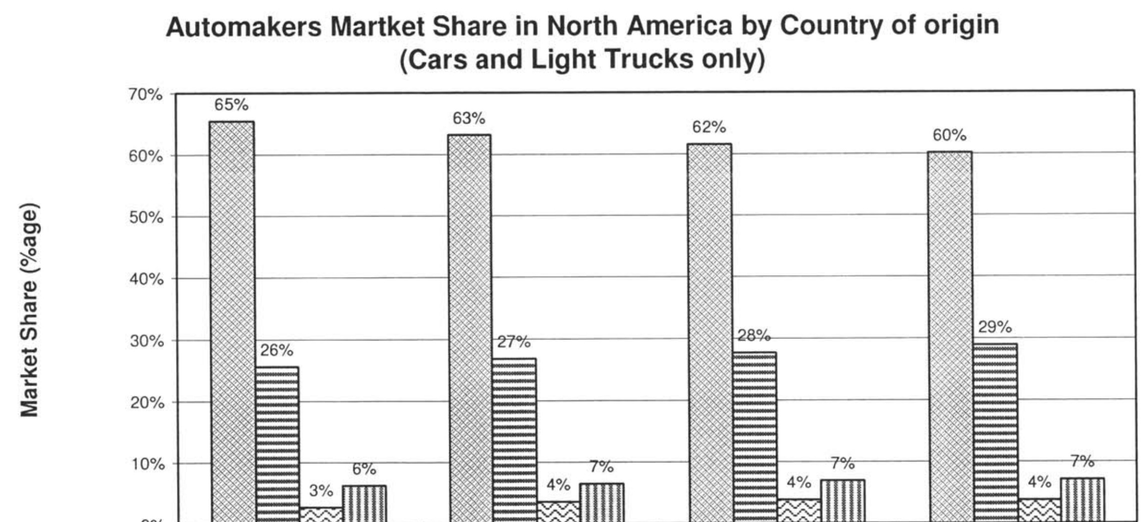
commodity products. If an automaker decided to switch suppliers because of better price or quality it may turn out to be devastating to the previous supplier's business. So suppliers hold very little power.

3. **Power of Buyers:** Historically, the bargaining power of automakers went unchallenged. The American consumer, however, became disenchanted with many of the products being offered by certain automakers and began looking for alternatives, namely foreign ones.
4. **Availability of Substitutes:** The higher the cost of operating a vehicle, the more likely people will seek alternative transportation options. The price of gasoline has a large effect on consumers' decisions to buy vehicles. Trucks and Sport Utility Vehicles have higher profit margins, but they are "Gas guzzlers" compared to smaller sedans and light trucks. Emergence of Hybrids and Fuel-cell vehicles may create a new threat to the incumbents.
5. **Competitive Rivalry:** Automakers understand that price-based competition does not necessarily lead to increases in the size of the market place, and historically they have tried to avoid price-based competition. But more recently the competition has intensified--rebates, preferred financing, and long-term warranties have helped to lure in customers, but they also put pressure on the profit margins for vehicle sales.

Apart from these five forces, firms are also facing some fundamental problem in terms of technology adoption. Technology market place is undergoing dramatic shifts. We find new gadgets that keep pouring into the market place everyday. This trend has brought in major problems to the automotive manufacturers. R&D departments have to constantly keep up with the latest gizmos in the market, both in terms of technology and consumer demanded features. This means that the firm should be in a position to adapt to the latest available technology very rapidly. However, key product strategist have to make a best valued decision to make it part of the product/platform strategy.

And then the automotive industry has seen a massive shift towards merging of major firms to gain market share as well as brand identity. Japanese and European companies were acquired by big three automakers either to inherit some of the best in class engineering practices or to compliment their product families to gain market position and market share.

For instance both Ford and GM were trying to fill in the gaps left in their portfolios and tried acquiring the luxury European brands. Ford outbid GM and acquired Jaguar, while GM went for Saab. A few years later Ford bought Land Rover and Volvo (passenger cars). However, the reverse happened to Chrysler Motors where in Daimler-Benz (European auto maker) acquired it to form Daimler-Chrysler.



	2000	2001	2002	2003
US	65%	63%	62%	60%
Japan	26%	27%	28%	29%
Korea	3%	4%	4%	4%
Europe	6%	7%	7%	7%

Year of sale

Figure 5: Automotive market share for years 2000-2003, by country of origin (Source: *Wards Auto*)

This new trend has brought in new issues with it. Traditional PD processes that were created based on the knowledge and culture within the organization now need to be adapted or modified to become integrated PD process. The business units in each of these newly acquired firms may need to adapt to this massive shift, which normally takes 5 to 10 years to make it consistent. Figure 5 shows how rapidly the North American auto manufacturers are losing the market share in terms of sales.

When Ford Motor Company acquired Jaguar Motors, Ford wanted to build its new Lincoln by sharing some of the costs with Jaguar's program team. Lincoln program team was to lead this effort following the Ford product development process that resulted in a 2 year program delay at both sides. There were many lessons learnt during that program execution and some of them will be cited in this study in the following chapters.

Taking this new challenge of integrated PD process activity into consideration we like to see what new issues prop-up in the case of implementing and devising common IT tools across Consumer Business Groups (CBGs). One other issue that we need to keep in mind is that when a new strategy is devised in an organization many new enablers are envisioned to be in place for the PD process to be successful. In many cases most of these new enablers take somewhere between 2-5 years to be fully functional and/or to deliver what they promised. One of the strategies that the auto industry started pursuing recently is the Product Life-cycle management (PLM) strategy.

PLM is basically a system or set of systems intricately tied together to maintain information of the product through different stages of its product creation process, from design inception to its end of life. For PLM strategy to work effectively, an organization should have well defined processes, methods and tools. And to enable these processes, methods and tools, a well defined standards based IT infrastructure needs to be in place. And the suppliers who serve the organizations may have to invest considerably to adhere to these standards. If this IT enabler is not in place, PLM strategy is sure to be a failure.

Information Technology under Review

We are in the digital age and most of us have lost touch on how to use paper and pencil. Nowadays even emails are accepted and are becoming more prevalent in large organizations for major project approvals. Inter-company mail that was heavily used prior to the emergence of email is slowing fading away. Paper documents are the thing of the past in most large organizations. Large firms encourage their employees to use online paychecks to save costs. And a perception has been reached by many organizations that anything that is computerized will cuts costs further. This perception led to the development of the new age problems; the growth of IT systems and IT tools.

Information Technology in Product Development

IT tools have come a long way in the PD business and can be used in a variety of ways. A few areas where IT tools are extensively used in PD are - work flow management, process management, project management, design (CAD, CAM), analysis (CAE), real-time systems, process automation etc.

The tightening of IT budgets has meant that enterprise IT managers are particularly keen to 'do more with less'. At present, PD organizations spend significant proportions of their IT budget – on average around 75%[iii] – on costs associated with keeping IT systems operational. Consequently, enterprise IT managers are looking for effective infrastructure management products that will generate returns by reducing support costs, increasing the productivity of the IT support function and generally consuming IT resources more efficiently. In particular, solutions that ensure prompt and effective IT infrastructure problem resolution will generate a strong demand from all organizations. IT managers will also benefit from solutions that can detect whether applications are not running at their full potential and are, therefore, losing valuable revenue.

As the role of IT in business service delivery increases, the emphasis placed on the IT department to deliver and manage the systems that support the organization also increases. Although this dynamic has been in place for a number of years it has recently become particularly marked given the renewed focus on the role of IT in a difficult economic environment. The impact of this trend is that the vendors have been forced to evolve their infrastructure management solutions from being purely technology-centric to becoming increasingly focused on business services.

IT Tool Adoption

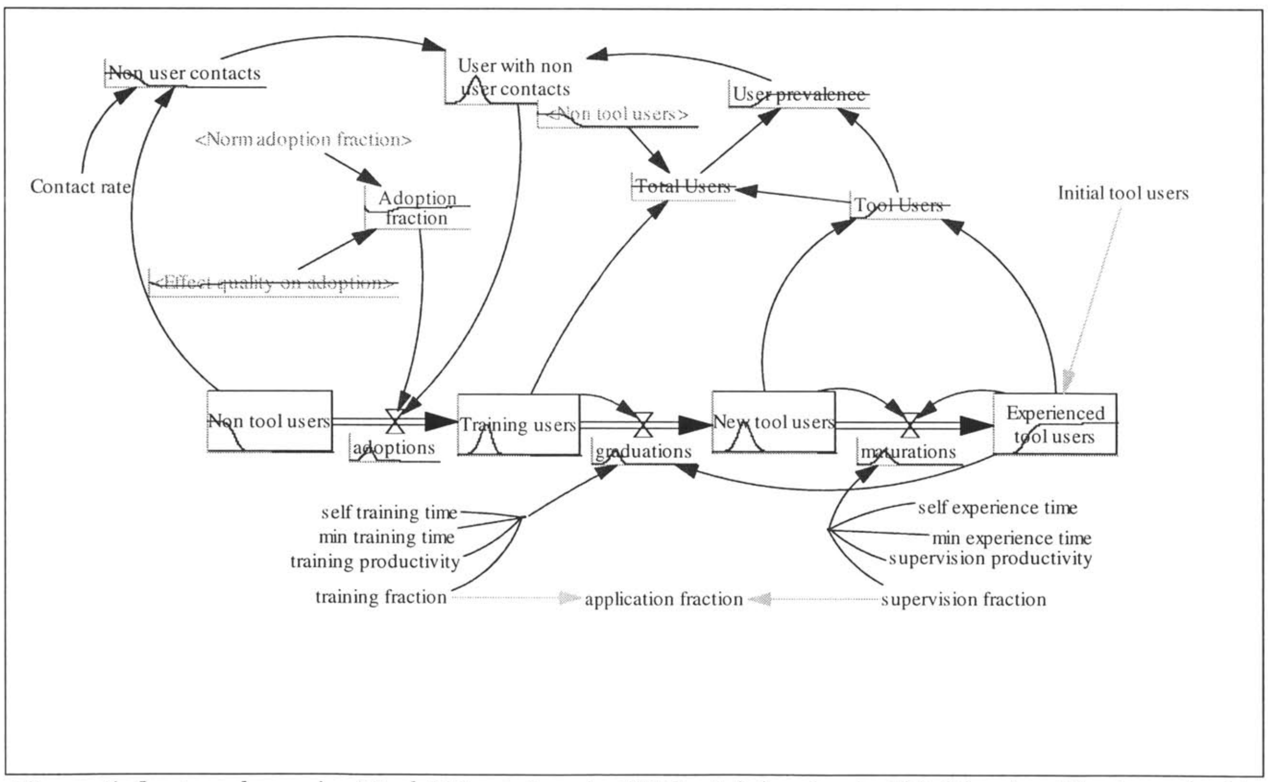


Figure 6: System dynamics Model Depicting the IT Tool Adoption in PD (*Based on Vensim molecules*)

IT has improved the life of a traditional PD engineer in many ways. Similar to many technologies out there, there is definitely an adoption curve for any new IT tool that is introduced into the PD space. We can easily correlate the new PD IT tool adoption by engineers to Michael Moore's new technology adoption by the consumer in the famous book on "Crossing the Chasm".

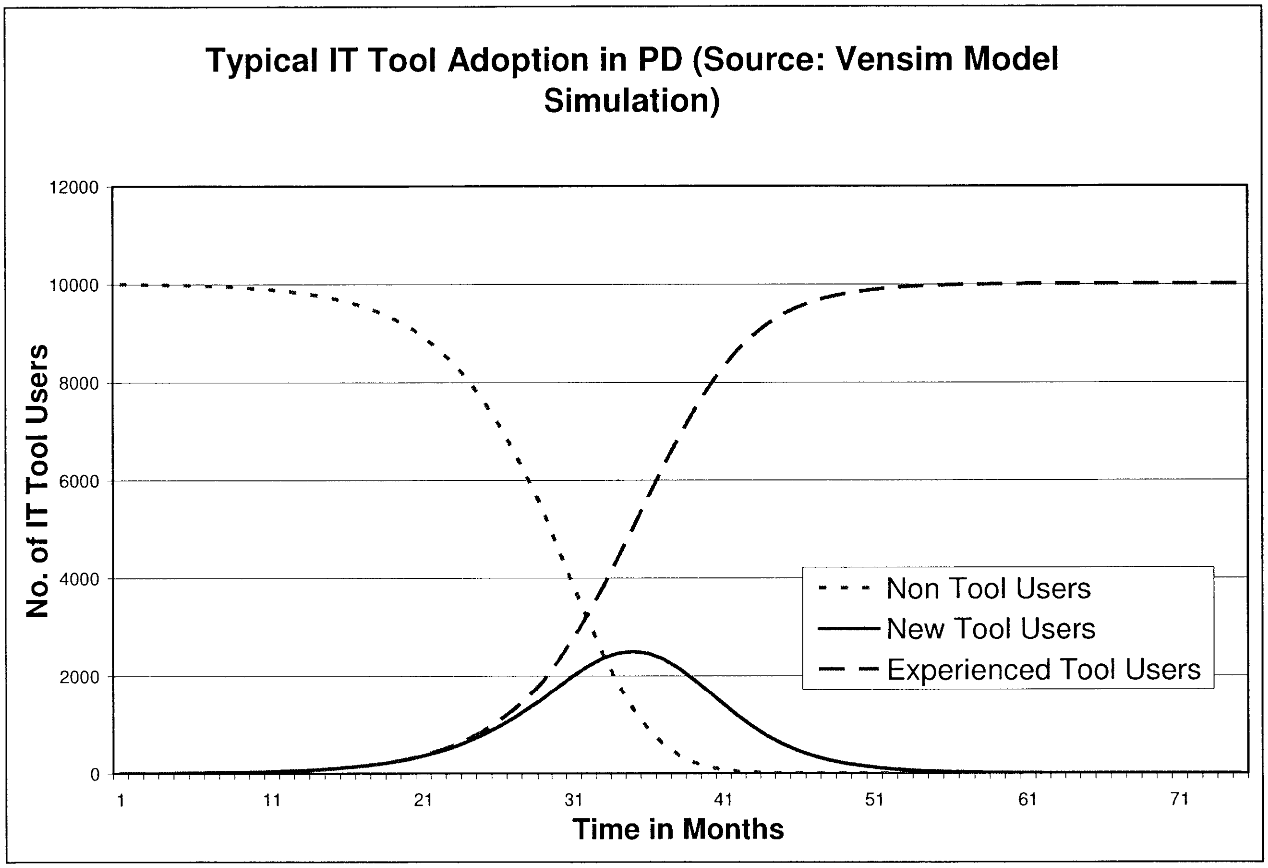


Figure 7: Typical IT Tool Adoption in PD (Source: *Vensim Model Simulation*)

Let us assume that the PD organization may have 10,000 active engineers that have to use IT tools to do their day to day activities. When a new tool is introduced into the PD space there is an adoption that needs to take place, similar to market dynamics. At first the tool is introduced to the core users who will be the 'specifiers' of the tool or the primary implementers. Once they are satisfied with the tool it is released to the general engineering pool for daily use. Most tool releases may need some initial training (1-6 months in duration) to be able to utilize the tool effectively. Some users may also stop using the tool if they don't find it user friendly or may find it very cumbersome to learn. Many such users may influence the premature decommission of the tool. However, if the tool has a high initial adoption rate it may continue to exist for a long time before other dynamics like feature creep (increase in complexity), technology limitations etc start playing their role.

IT Customer

For the purpose of this thesis, the assumption is that the IT customer is a PD engineer who would be the primary benefactor of any tool he/she may use in their day to day operations, whether it is design, planning, verification or build.

The PD engineers come from different areas within PD. Areas like design engineering, attribute engineering (Durability, Safety, NVH – Noise Vibration & Harshness etc), test engineering (Durability testing, Safety testing etc). Design engineers are the ones who design the new tooled end items or new parts. These 'End items' can be just components, sub-systems (Cooling system), systems (Transmissions). Design engineers seek the help of attribute engineers to understand the different system/sub-system specific testing to satisfy design specification or targets pertaining to a particular attribute. And the attribute specific engineers' depend on the test engineers' availability to conduct the attribute specific tests on these new parts/"End items" (Test engineers and technicians physically conduct the tests at labs). Design engineers also release the tested 'End items' to production after the prototype is verified and certified for production by all the attribute specific engineers.

The PD activity also consists of program or project management staff. These people are responsible for the work flow or the PD productivity metrics. They have to oversee that the program is continuing as intended. IT also serves this community with a wide variety of tools.

However, the inputs or information entered into the tools used by the program management directly comes from the PD engineering community; from both upstream (Design) and downstream (Build and verification) activities.

(THIS PAGE IS LEFT INTENTIONALLY BLANK)

CHAPTER 3
SYSTEM ANALYSIS & MENTAL MODELS

Hypotheses

IT departments in big organizations have to work like business units bringing value to the corporation and in doing so would have to work towards delivering the proposed business value to the immediate corporate customers. The organization by itself may consist of several areas like PD, Manufacturing, Purchasing, Sales & Marketing etc. IT as a service organization needs to clearly understand what the organizational behaviors and processes of its customers are to provide the best possible solution.

Understanding the internal behaviors of the PD area is a huge undertaking. The DPS system dynamics tool that was developed in the PD area for the organization under consideration (explained in Chapter 4) has been the most useful system dynamics work thus far to visualize the internal behaviors and structures of the organization. Even though the work is slightly dated, the core systems engineering processes and structures remain unaltered. Based on the DPS model I was able to gain a lot of insight of the PD area as a "System".

Several key hypotheses were made during the process of this study. And I have listed them in the order of their findings.

1. Number of IT Tools: The main problem that has been troubling the IT department and IT managers in major PD firms is that the number of IT tools that are in the space have grown to such an extent that it has become extremely difficult to develop, enhance, manage and maintain them. The number of tools identified in the firm was astonishingly high. The graph below (in Figure 8) shows distribution of the approx. 90 most critical IT tools used in

different stages of PD. Some of the tools are used in multiple stages/phases of PD. The majority of IT tools used are in CAE and Physical verification phase of PD. (Refer to the case study in Chapter 5)

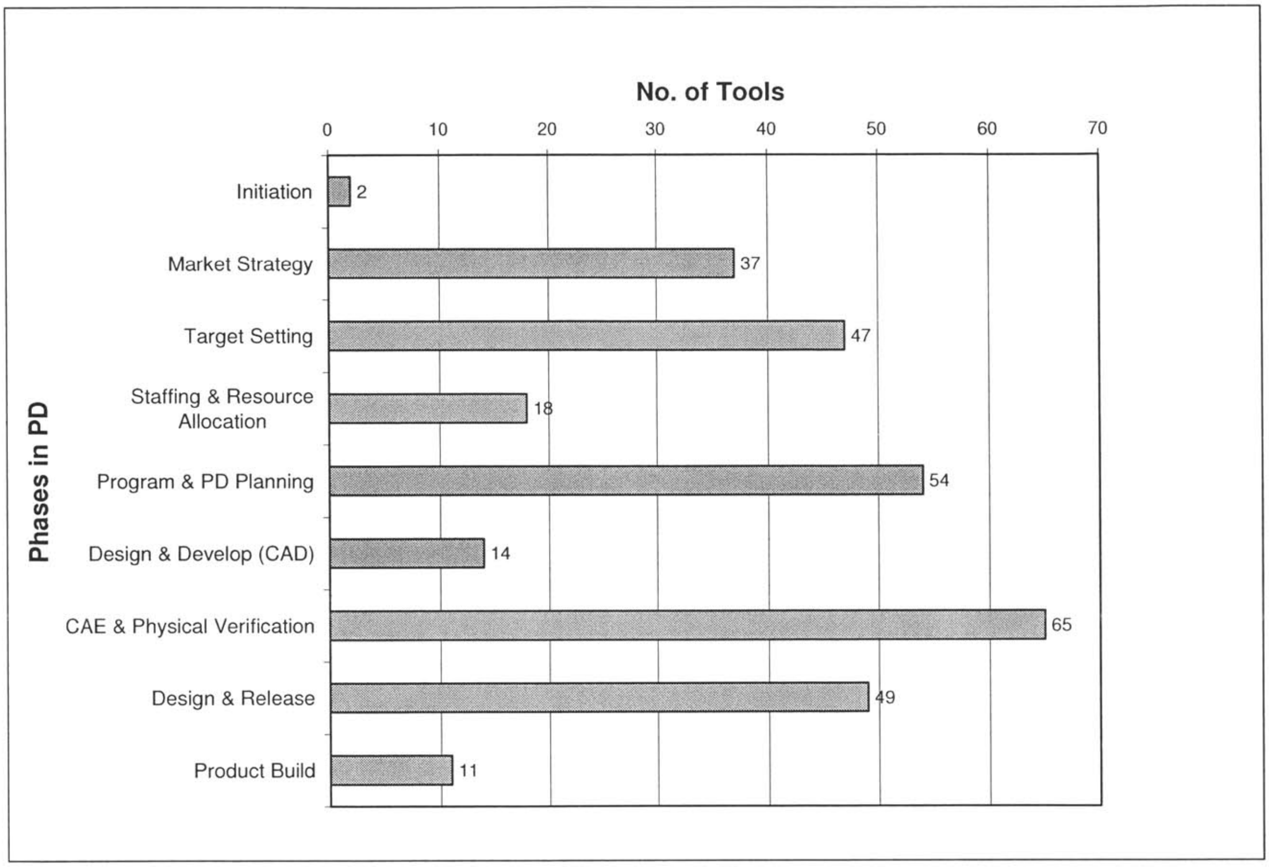


Figure 8: No. of Critical IT Tools Used in Different Phases of PD

2. Legacy Systems: The sheer number of the tools is not just the problem. Some of these tools have been running on operating environments that are at least couple of decades old. IT managers have a daunting task of keeping up with the technology shifts and resource constraints for maintaining these systems. Many IT tools require dedicated resources for maintenance. To add to these problems there are new threats like viruses and security loop

holes in these old operating environments that do not have a fix. In most cases these environments have to be protected with additional firewalls and dedicated networks to operate them.

3. Virus and Intrusion Detection: There are new threat management technologies that collect and decipher output data from a number of different vendors' security products, such as firewalls, intrusion-detection systems and Anti-Virus software. Data and events from security products are typically aggregated and then correlated, enabling enterprises to gain a holistic understanding of a network's security status and take immediate action to address any potential or actual intrusions. However, legacy systems and legacy operating environments cannot utilize these new technologies making the entire enterprise' systems compromise their security for the critical few systems to operate as intended.

4. IT Software Development Methodology: There is a perception within some areas in the PD community that software development methodologies used by corporate IT has created new hurdles. So, it is felt by PD that it always better to create a new tool rather than spend money & time in enhancing or maintaining the old & existing tools. The software development methodology turns out to be 80% process and 20% implementation which are almost similar to product development. This in turn turns off business customers from understanding the value in following the process.

5. Functional Silos: PD operates in functional silos and IT has to align to this. The need to recreate always exists in both places. IT developers are accustomed to the old ways of coding and do not want to move away from their inherent coding techniques. Since these developers do not want to move to new advanced technology and latest platforms, those old platforms have to be maintained just to satisfy the old thinkers.

6. Growth in Data: There is growing problem of data storage and data retention and data protection. The data needs of corporations are doubling every 12 months compared to the processing needs that are only doubling every 18 months (Moore's law – see Figure 9). The primary reasons for data growth in PD area are due to lack of data retention policy. Once data is stored in the corporate systems no effort is made to clean the systems. As engineers move from one position to another, they leave behind volumes of design and test data behind with out any new assigned ownership for the data.

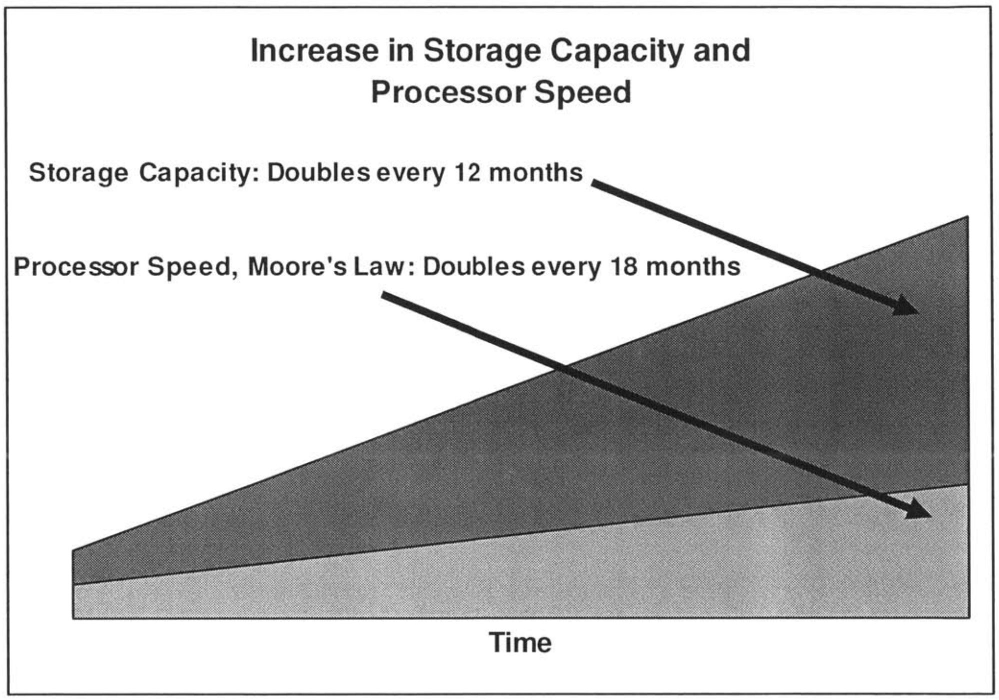


Figure 9: Increase in data storage needs - An industry prediction

7. Technology Drivers: Raw data that is stored and driven out of the real-time systems have no common architecture or back-bone. Data streams or data exchange mechanisms are missing that would enable data availability to enterprise in real-time. Standards are not globally adhered to and each site has its own set of vendor list. Many of the PD objectives like Zero-prototypes, Virtual PD, Real-time Bill of Materials & Analysis and Product

Lifecycle Management - all need standardized infrastructure with seamless data exchange interfaces. If these technology barriers are not addressed, all the efficiencies that IT has promised will be just limited to a silo or specific area. If the IT tool is not expanded beyond the silo it will be another redundant tool consuming resources and costs.

Eliminating redundancies across the above stated hypotheses, I have selected 4 out of the 7 key hypotheses for this study. I have also added some quantitative information derived from the data collection activity in the list below

1. Too many IT tools exist to support PD process (Functional silos) • 90+ critical IT tools support different phases of the PD process • Only an average 37% customer favorable rating for process intensive tools • 200+ Shadow IT tools (Vendor supported IT tools) • PD Testing labs, design studios, build sites all have different ways of executing projects • Tool functional requirements vary by PD attribute teams • IT development teams do not have knowledge sharing or code reuse mentality • Testing labs/Sites have their own set of problems (Union, Capacity fluctuations, Multiple vendors) – <u>Major hindrance for CAE deployment</u>
2. IT development methodology causes major delay • PD Perceives this delay is due to process (80%) and tool development (20%) • A major change request may take up to 3 months to be delivered • PD was used to seeing these changes take place within weeks.
3. Growth in data

• Corporate data is doubling every 12 months • No process exists in PD to follow data retention policies and data protection
4. Technology cannot be fully leveraged due to legacy infrastructure • Real-time systems have no common architecture or back-bone • Standardized data warehouses and data sources are missing to enable data availability to enterprise in near real-time to drive effective decision making • Standards are not present, each site has its own set of vendor list • PLM's (Product Life-cycle Management) vision may be impacted if this is not fixed

Interview Process

The interview process began by identifying the key PD engineers and IT subject matter experts (SMEs) that are familiar with the PD activities and the PD-IT tools. These individuals were contacted for one on one interviews; each interview session lasted somewhere between two to three hours in duration.

A detailed question and answer session followed in a structured fashion with many questions related to PD processes, the culture in PD and the day to day activities within the PD space. Emphasis was placed on questions that would relate to the past PD processes, the IT tools usage by PD engineers and the impact of new PD process on IT tools and enablers. Some key questions are listed below in Table 1:

No.	Sample interview questions
1	Among the different areas of PD such as Design, Build, Testing etc, which area consumes the maximum time and resources of a PD engineer?

2	How did so many IT tools evolve in the PD space?
3	How are tools introduced into the PD area?
4	What kind of tools do engineers prefer to use? For e.g., Process intensive, utility or generic (word, excel), function specific or Non-process intensive tools.
5	Do you think IT is able to deliver what the PD engineers need? Why or why not?
6	Why do some IT tools have higher satisfaction rating over other IT tools?
7	Describe how the new software development process is helping IT tool development.
8	Why is CAE not effectively used in design and verification phases?

Table 1: Sample interview questions

IT SMEs who have seen several years of technology and tool changes were able to provide in-depth information to many of these questions. PD engineers were able to validate most of the empirical evidence derived from the qualitative customer satisfaction survey data.

From each of these sessions key takeaways and patterns were documented and tabulated.

Key interviews takeaways	
1	There is a lack of PD business knowledge within corporate IT (IT)
2	It is perceived that IT lacks the capability to bring new technology into PD sooner
3	Vendor-PD (Shadow IT) partner relationship is much stronger than IT-PD relationship
4	Gap exists in the purchase of the software products and services
5	PD process driven IT tools have least customer satisfaction – So it was advised to avoid putting too much logic into tools
6	Shadow IT exists because PD gets solutions to engineers faster and cheaper
7	Shadow IT lacks the same process rigor that is used in IT
8	PD lacks understanding of IT initiatives and does not want to understand IT problems
9	PD process tools are cumbersome to use because IT was not involved in upfront analysis

	of process. IT was only invited to implement the process. IT implements process using WYSIWYG (What you see is what you get) philosophy.
10	Program managers need to access and know multiple systems to derive information for making decisions.
11	There is no real-time dashboard for program management because of the functional silos and de-coupled nature of the IT systems.
12	PD Design & Release engineer spends more time than desired in process tools to figure out how to get his/her work done.
13	Inconsistent tool usage across teams makes life difficult when engineers move into new assignments (Greater learning curve).
14	PD attribute engineers are domain focused and try to achieve higher efficiencies in their domain without understanding the impacts to other domains. (Upstream and downstream activities)
15	Devised own domain specific processes – Efficient but inconsistent metrics and data collection across different PD engineering teams
16	Information is not fully shared or uploaded to upstream and down stream systems by domain engineers (Attribute and test engineers)
17	CAE cannot be effectively used because certifiers sometimes require physical confirmation. Corporate assets are under utilized if we don't take advantage of physical test sites.

Table 2: Key interview takeaways

Data Collection Methods

Different data collection methods were devised to validate the key hypotheses that were stated above. IT tool customer satisfaction survey data, data regarding IT tool usage, data regarding the choice of testing method used, and other qualitative data (Information from strategy white papers, PD design issues and concerns documentation) was collected.

Survey Data

IT organization that helps PD practice conducts quarterly surveys to gain the satisfaction of their PD customer (PD Engineers, planners, managers etc) for each of the prominent IT tools. Both quantitative and qualitative data were collected from these surveys. The data was from 6 different surveys over a period of 3 years.

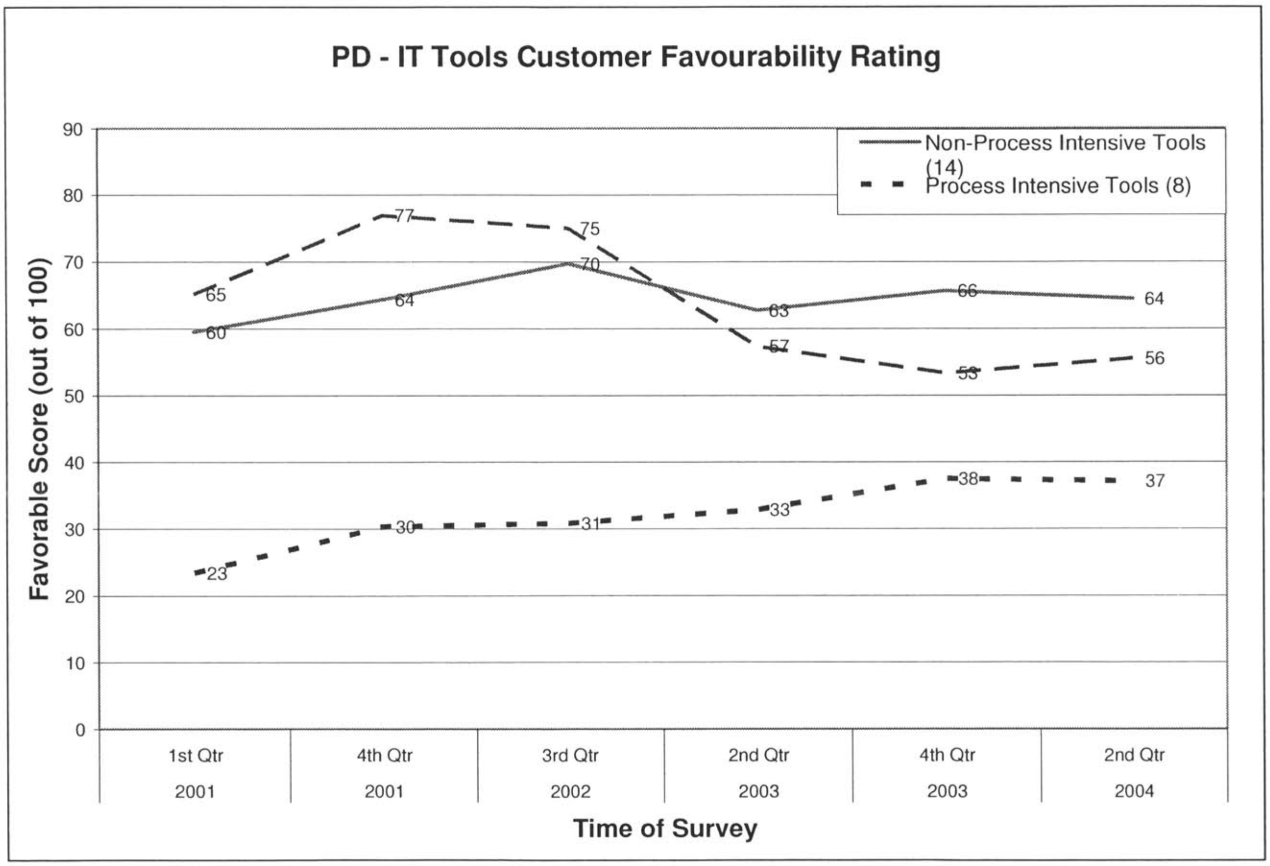


Figure 10: PD - IT Tools Customer Favorability Rating (Source: *PD Tools Survey*)

I have divided the prominent tools into 3 categories.

1. *Non-Process intensive tools* – Tools that are mostly used for deriving reports, tracking issues, comprehensive data searches. In most cases these tools may have less manual input.
2. *Process intensive tools* – Tools that need an engineers input and are mostly used by upper management for status reporting. These tools are directly tied to the PD business processes.
3. *Utility tools* – Tools that help engineers in their day-to-day activities. Tools that help in creating analytical graphics, CAD drawings, CAE analysis etc.

Figure 10 shows the PD customer favorability ratings based on the 3 categories of tools.

It is quite evident from the data in Figure 10 that "Process intensive tools" are least favored by the PD customers. There are several reasons for this phenomenon.

1. PD customer come in different flavors and each have there own specific need for a tool. As a tool grows in functionality, the complexity of the tool may contribute to the unfavorable rating.
2. Tools do not provide an easy functionality (architectural issues) and may impose a lot of process based restrictions to proceed with the task inside the tool. As the time spent per task increases, the satisfaction index of the tool declines.
3. Requirements change due to organizational changes and this dictates tool changes. PD customer has to understand the usage of the tool to effectively utilize it. If help or training is not available it becomes very difficult to be productive on the tool.

Data Repositories

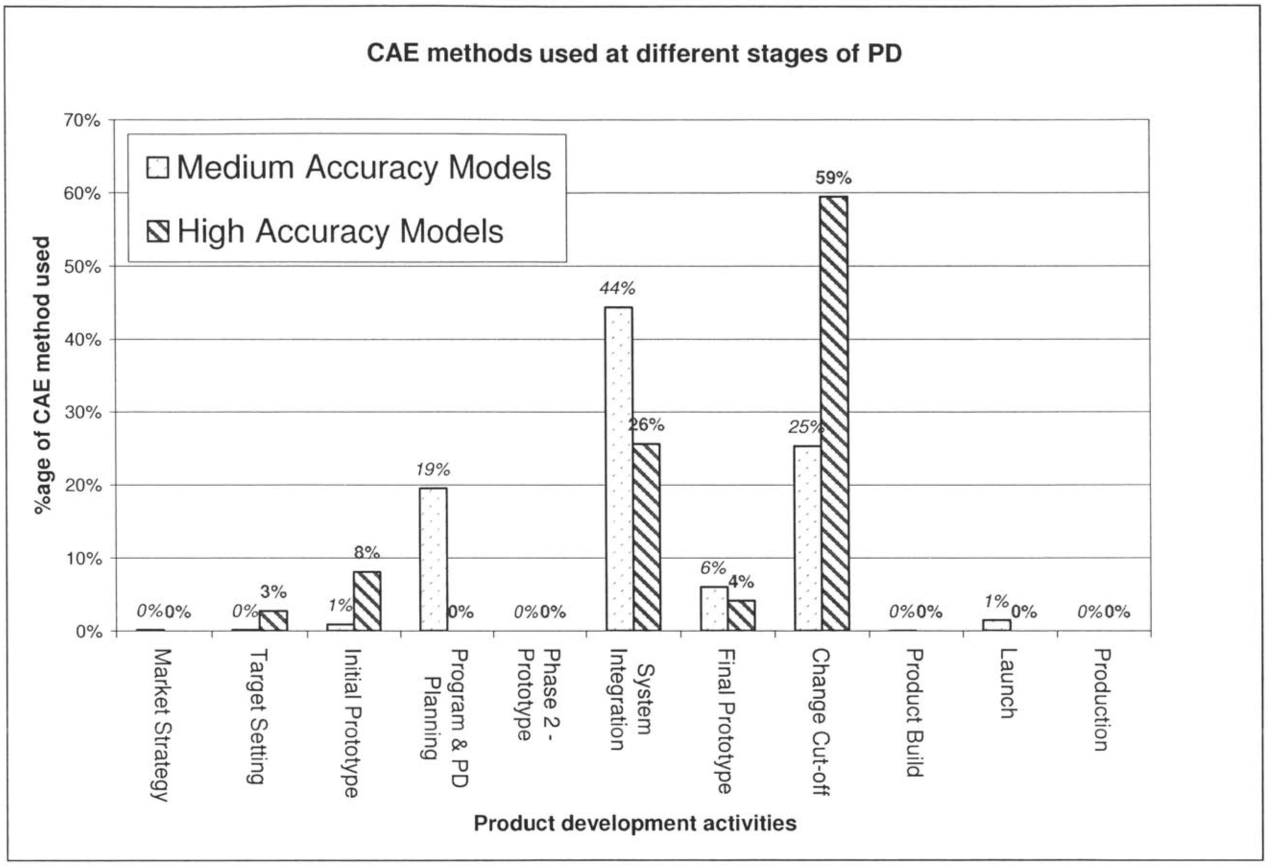


Figure 11: CAE methods used during PD for design validation and verification

Data was collected from data repositories to understand the use of analytical tools and methods by PD engineers during the design and verification phases. Specifically, the use of computer aided engineering (CAE) verification methods when compared to the physical verification testing and methods. It was found out that only 25% of the testing was done through CAE methods and the rest was done through other physical verification methods. The models used during pre-design validation are not represented in the Figure 11. Only data from design verification is represented. During the process of the data collection it was found out that the metrics from pre-design CAE models usage is hard to find. Most of the CAE models are either reused or morphed from previous years' programs for future programs.

Information Sources

It was identified that there were considerable amount of shadow applications (IT tools developed by vendor companies without corporate IT organization's involvement). Information regarding the costs incurred for managing IT tools was also collected. This data is not presented in this study due to confidentiality. However, some of the dynamics are explained later in this chapter. One of the key findings was that PD invests lot of money on reinventing similar functional tools for different functional attribute teams. No focus is given to share or reuse similar tools or functionality.

Understand Identified PD Behaviors through Causal Loop Diagrams

IT Tools Growth:

One of the very important observations that I have made during my interview process is the IT tool growth phenomenon in the PD area. In the Causal Loop Diagram (CLD), the balancing and reinforcing loops are represented with the letters 'B' and 'R' respectively; and the loops are numbered for easy identification. The *italicized* phrases are causes and effects inside the CLD. Arrows are also bolded in some figures to show the directional flow of the causal loop.

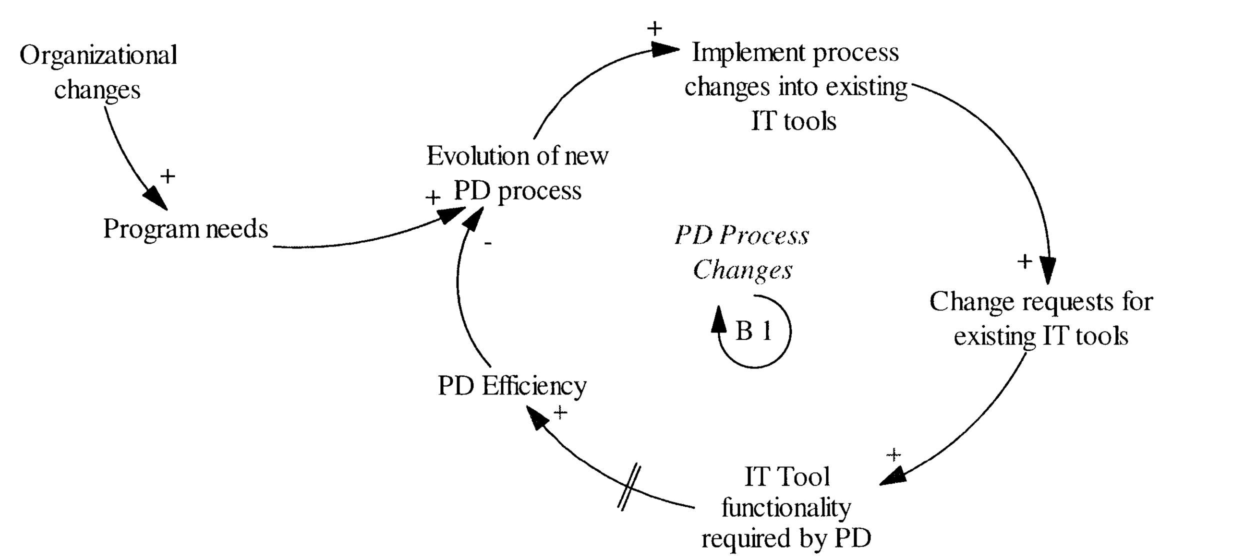


Figure 12: PD Process Change Loop – B1

PD Process Change loop (B1): Generally in PD when there is a need for improvement in timing of a program (*Evolution of new PD process*) or if there is a need to introduce a new policy that may bring in new productivity and quality improvements, it becomes necessary for IT tools to adhere to that change (*Implement process changes into existing IT tools*). Any new process change inside an

IT tool requires an IT change request procedure to take place (*Change requests for existing IT tools*). Once the IT tool change is completed the new process (Improved) functionality will be available (*IT Tool functionality required by PD*). However, introducing the new process functionality does not mean that the impact of the process change is felt immediately. So, there is a certain delay before any *PD Efficiency* is realized. Traversing this loop results in a balanced activity where in the need to further evolve new processes goes down over time.

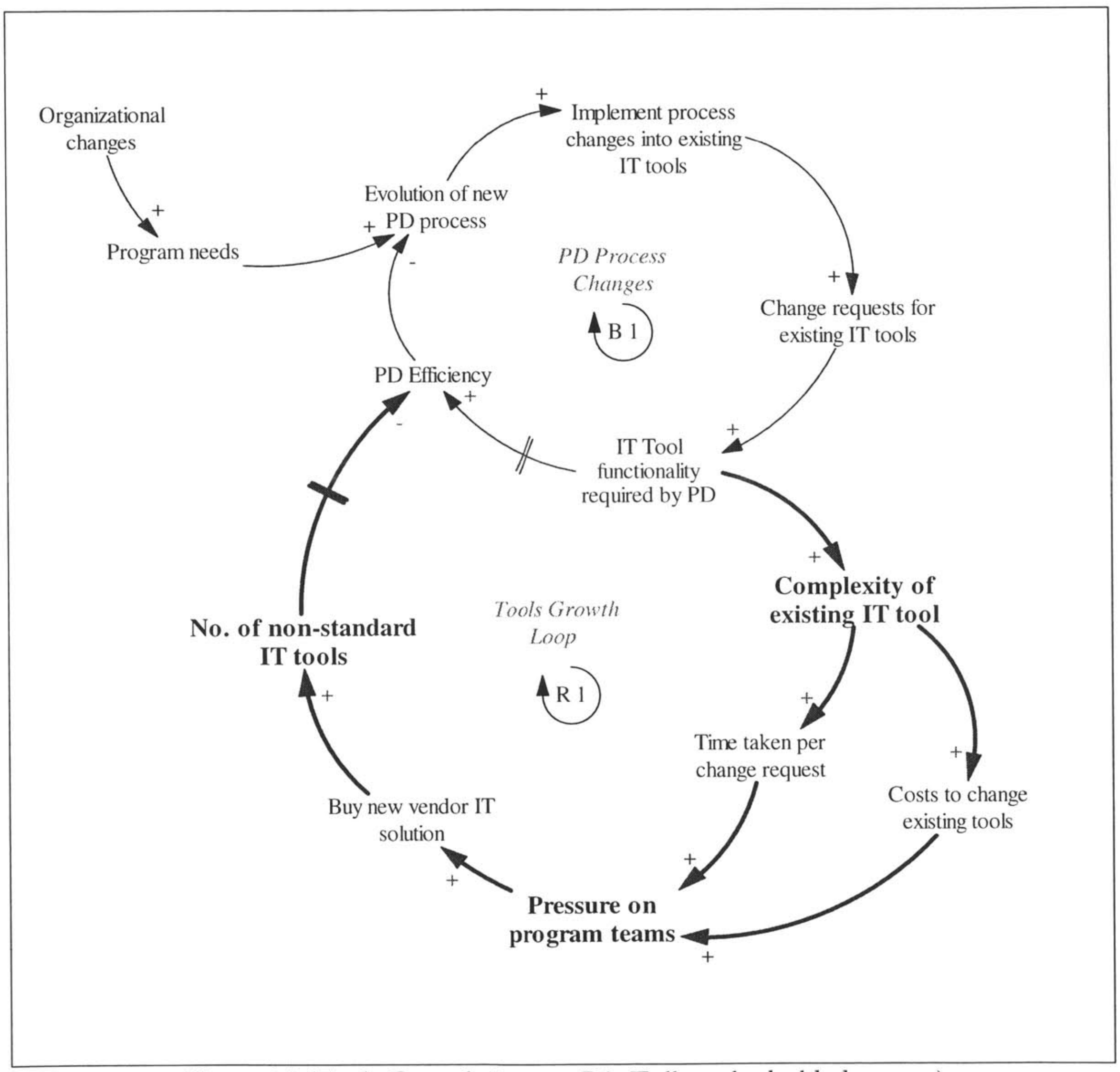


Figure 13: Tools Growth Loop – R1 (Follow the bolded arrows)

Tools Growth Loop (R1): Adding new functionality to an existing IT tool will only add to its complexity (*Complexity of existing IT tool*). Complexity of an IT tool is measured in many ways. Function points count is the standard used to measure complexity of the IT tool in the firm. As the complexity of the IT tool increases, the time taken to implement a change also increases (*Time taken per change request*) and the costs associated with making such complex change also increases (*Costs to change existing tools*). Both timing and cost increases lead to increased pressure on program teams (*Pressure on program teams*). Program managers may either choose to develop solutions quickly by allocating funds and resources if they see a delay in implementation of new IT tool change or indulge in control/extreme measures such as buying a vendor IT solution (*Buy new vendor IT solution*), mostly off the shelf, resulting in growth in new IT tools that do not follow corporate IT standards (*No. of non-standard IT tools*). This leads to perceived process improvement in short term, but in the long run (with a delay) as these new non-standard tools do not adhere standards, result in decreased PD efficiency (*PD Efficiency*).

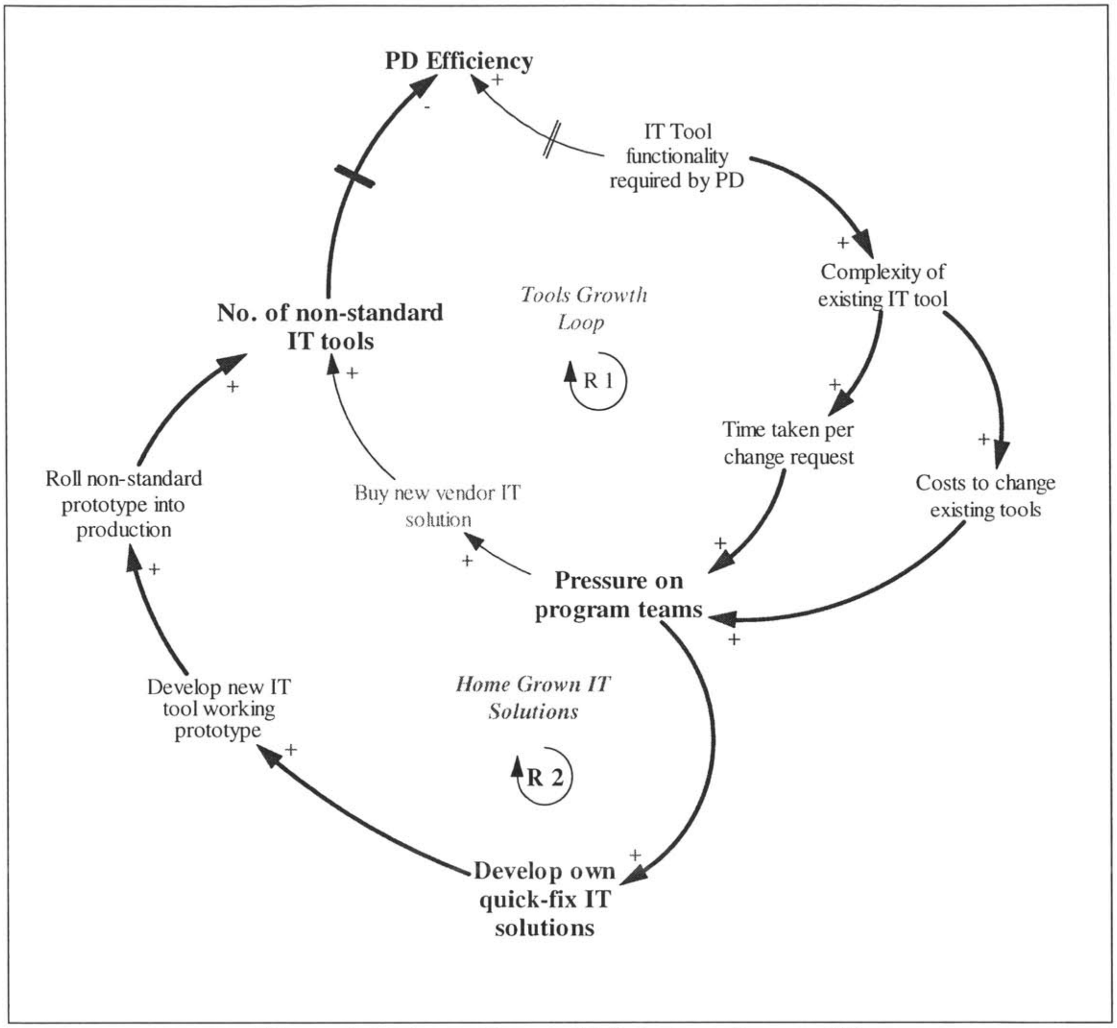


Figure 14: Home Grown IT Solutions Loop – R2

Home Grown IT Solutions loop (R2): As seen in the loop R1, Pressure on program teams also leads to couple more noted behaviors in PD. One such behavior is developing their own IT solutions (*Develop own quick-fix IT solutions*) by quickly using computer skills obtained through reading computer books. In most cases these fixes work with what they are trying to deliver. However, these solutions may not fit well with corporate solutions. New quick-fixes lead to new working prototypes, which might appeal to the corporate PD management, resulting in a new IT tool birth. Such tools may quickly become part of the main stream production tools (*Roll non-standard prototype into production*) without any corporate IT standards. This activity just leads to another reinforcing loop resulting in the proliferation of more non-standard IT tools *(No. of non-standard IT tools)*.

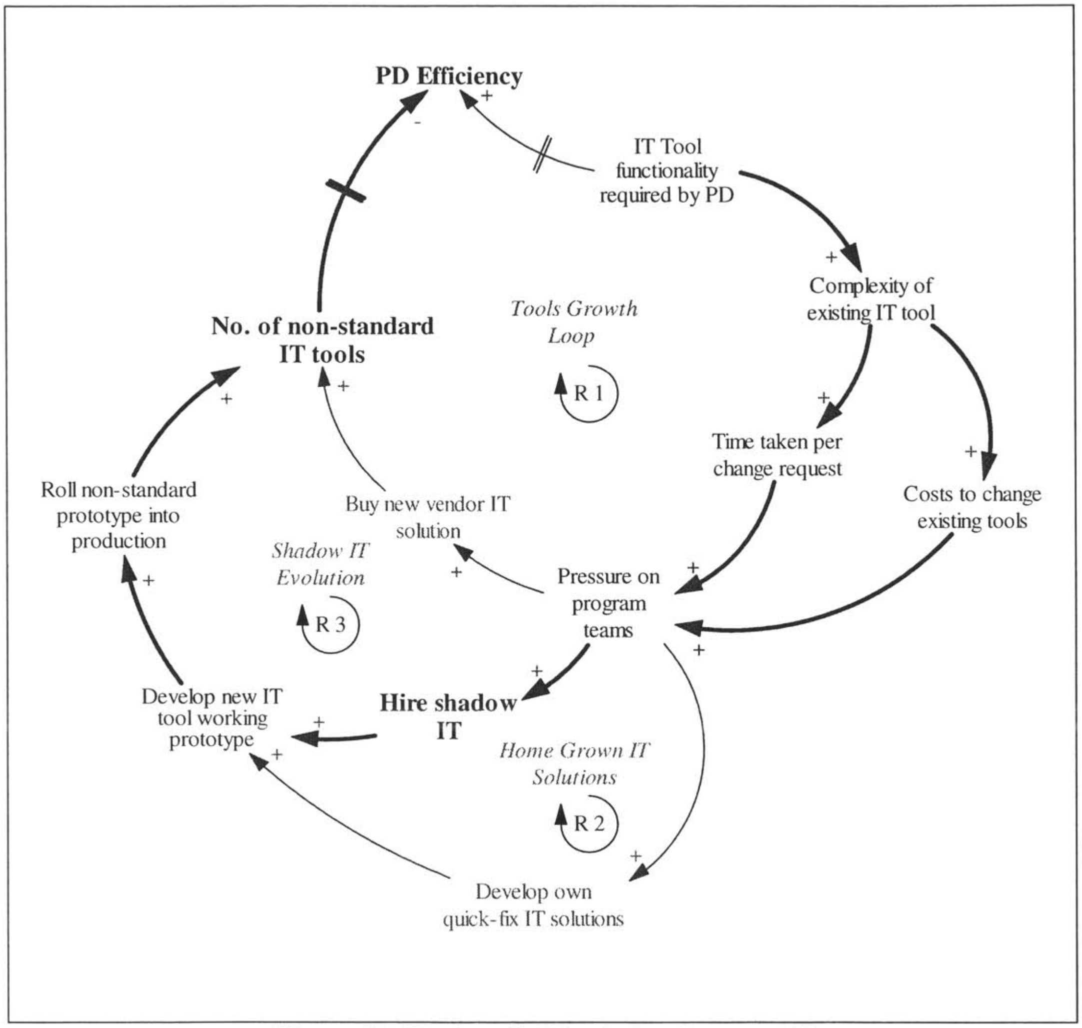


Figure 15: Shadow IT Evolution Loop – R3

Shadow IT Evolution (R3): The third effect (*Hire shadow IT*) caused due to *Pressure on program teams* results in the Shadow IT evolution. Shadow IT is a term used to identify IT activities performed by contract IT staff. Traditionally major IT tools were developed by contract firms with partnership with PD departments. This partnership and IT tool development activity is still prevalent within PD as a source for quick turn-around IT solutions. This behavior produces another reinforcing loop resulting in the proliferation of more non-standard IT tools *(No. of non-standard IT tools)*, as seen in the loop R2 (refer to Figure 14).

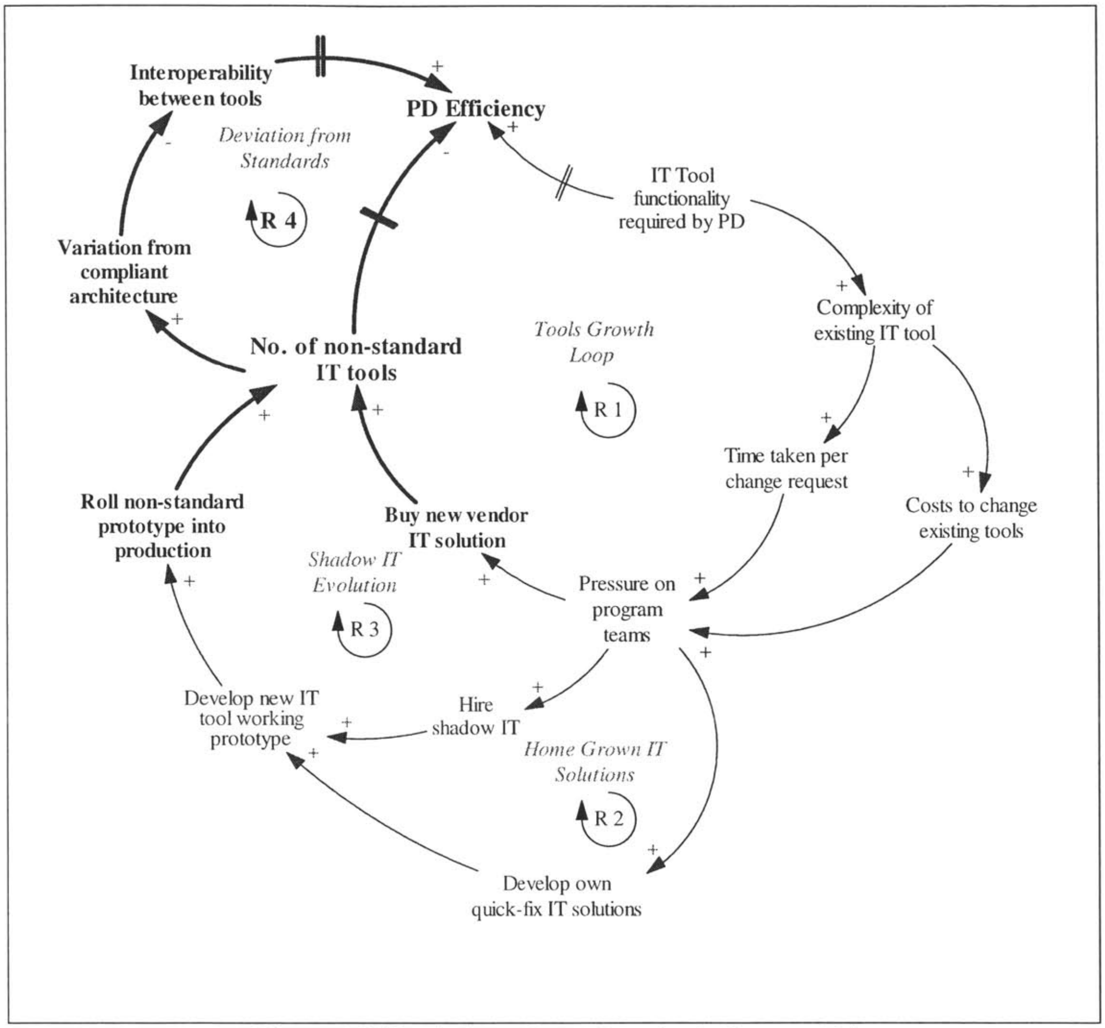


Figure 16: Deviations from Standards Loop – R4

Deviations from Standards (R4): Loop R4 another reinforcing loop is caused due to an increase in the *No. of non-standard IT tools.* An increase in the no. of non-standard IT tools will result in an increased "*Variation from compliant architecture*" and then a reduction in the "*Interoperability between tools*". As the IT tools become more de-coupled and desperate, tools will lose the opportunity to automatically exchange data and information between themselves. PD engineers have to log into numerous tools to get a set of tasks done, resulting in reduced productivity (*PD Efficiency*). To stabilize one process in B1 loop results in 4 reinforcing loops R1-4, only leading to more oscillations and inefficiencies in the PD processes.

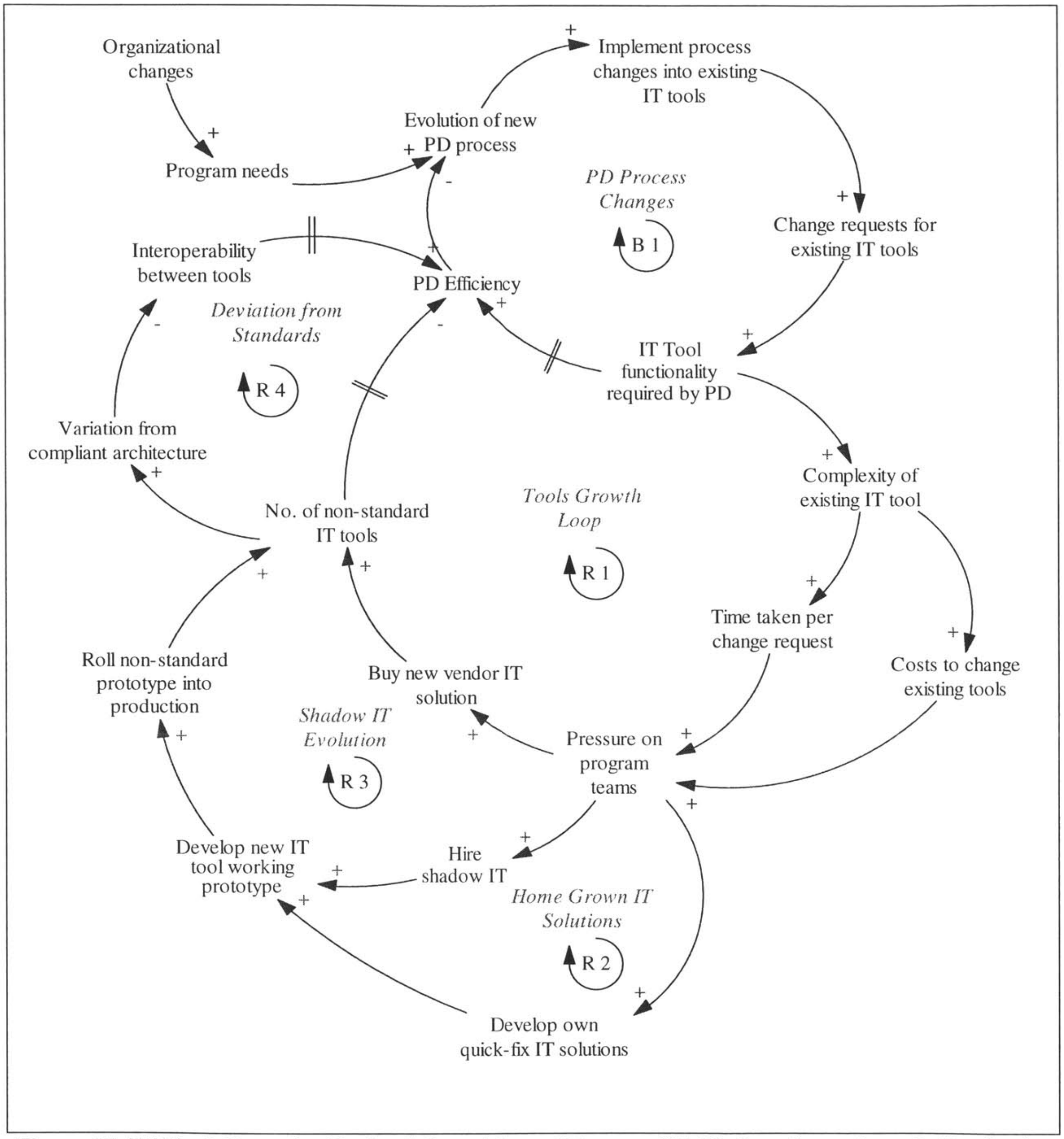


Figure 17: IT Tool Growth - Derived Causal Loop Diagram (CLD) from Interviews & Qualitative Data

IT Tools Growth in Test Sector:

Test engineers, design engineers and CAE tool specialist were interviewed to derive a CLD specific to the Test sector (Please refer the CLD in Figure 21). Design engineers have to validate their design parts or systems. They either use a physical prototype to conduct a variety of physical cataloged tests or use computer aided engineering (CAE) models to perform similar analytical tests.

Physical tests need test engineers and the availability of the test facility/equipment where they can perform these tests. CAE testing needs accurate CAE models for the designed parts/systems that need to be tested. CAE method is faster of the two testing methods. However, traditionally all tests were conducted through physical testing and this culture is hard to die. Since the engineers feel comfortable using physical testing compared to CAE testing, they prefer to indulge in the laborious and time consuming physical verification methods to complete their design verification.

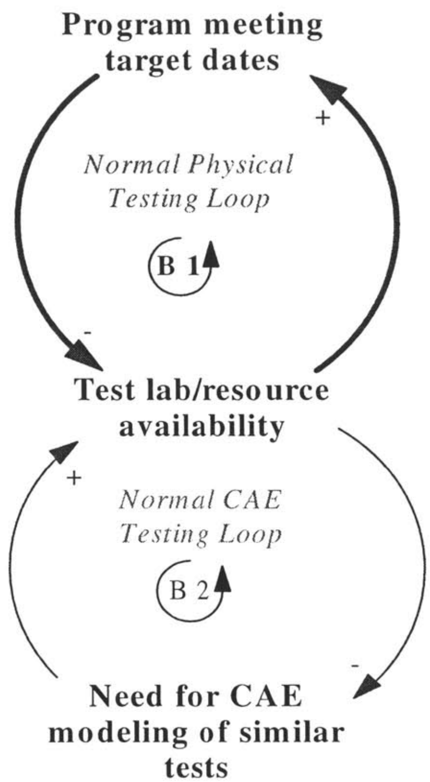


Figure 18: Normal Physical Testing Loop - B1 & Normal CAE Testing Loop - B2

Table 3: Counter Balancing Testing methods

Normal Physical Testing Loop (B1): During the PD testing phase, program teams have to meet the program timing that is normally sketched out in the design verification plans. It is quite essential that the program team meets its set timing targets as other teams may be dependent on the design verification results to proceed with the new designs. However, *Program meeting target dates* is subject to the availability of test engineers who conduct the specific tests on the newly designed parts and the test facilities where these tests are conducted. The *Test lab/resource availability* variable can be fixed. When there are several programs in the verification space, there will be times when there is shortage of testing resources (E.g. Test drivers for vehicle that need road tests) and shortage of test facility (E.g. Labs like conference rooms can be booked for a period of time by a certain program) resulting in a delay to program timing and a balancing action.

Normal CAE Testing Loop (B2): Program teams generally do not want to hold off their work when physical testing facilities are not available (*Test lab/resource availability)* to them. Normally this time is spent in testing new designs using CAE model simulations. However, to perform CAE based testing, engineers need CAE models that adhere to the new design specifications for parts/systems. This leads to the *Need for CAE modeling of similar tests*. As engineers spend more time pursuing new CAE models there may be an increase in *Test lab/resource availability,* which leads to another balancing action.

A very interesting thing to note here is that both loops B1 and B2 are counter balancing loops, or depiction of a catch 22 situation in the test sector. A major culture shift or a policy change may be needed to break this counter balancing effect to take advantage of faster CAE test timing.

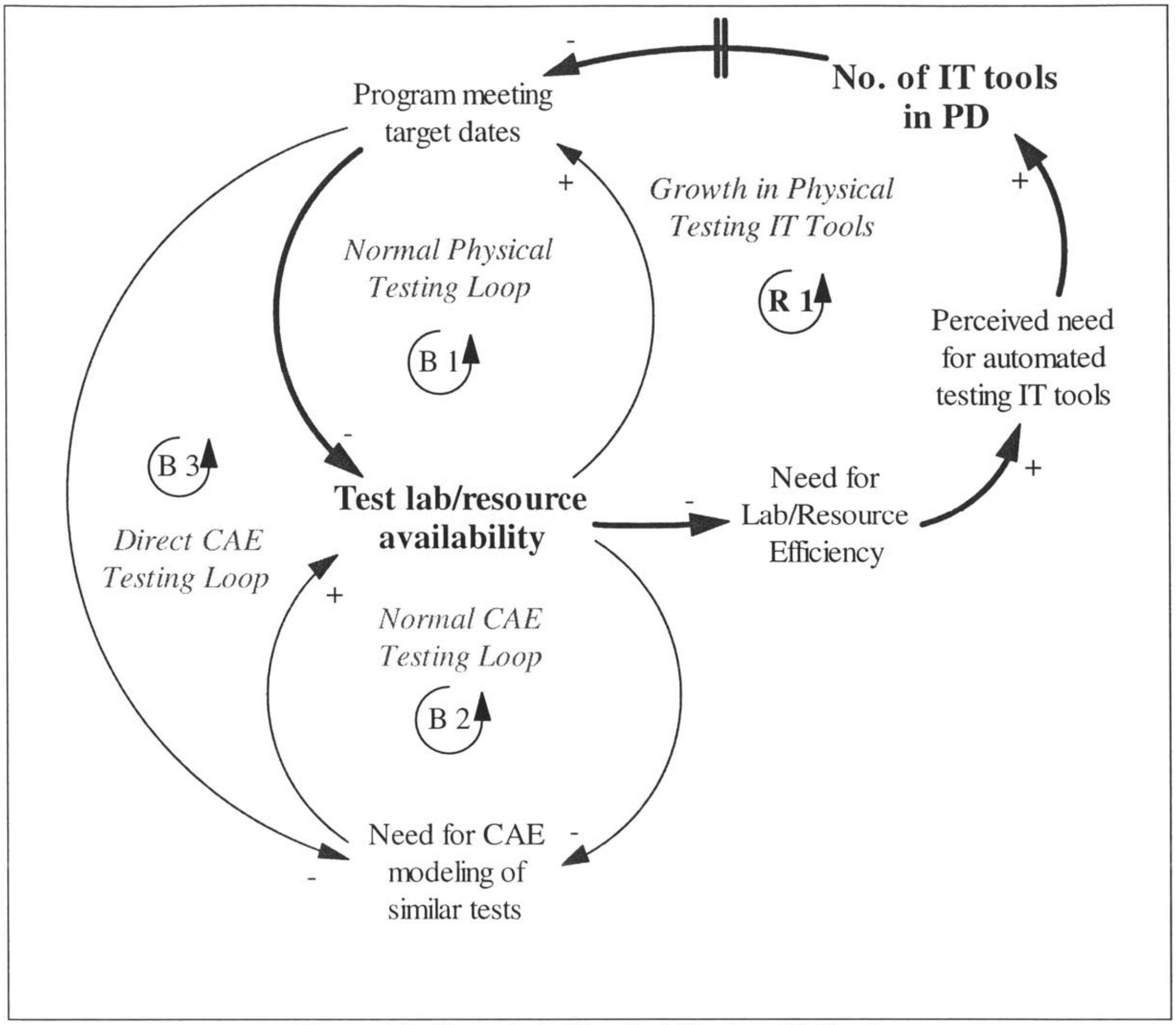


Figure 19: Growth in Physical Testing IT Tools

Forced CAE Testing Loop (B3): If a program manager foresees *Test lab/resource availability* situation ahead of time then he/she may recommend taking a direct path to CAE testing. This loop has the similar dynamics of loop B2 due to *Need for CAE modeling of similar tests.*

Growth in Physical Testing Tools (R1): As the program demands grow, there is a need for managing test lab and resources optimally (*Need for Lab/Resource Efficiency*) so that they are available to the critical program testing needs. One way of doing this is introducing automation in the labs to directly review design verification plans and then optimizing the lab/resource schedules.

However, to introduce such productivity measures means that there will be a need for new automated methods and tools (*Perceived need for automated testing IT tools*). One thing to keep in mind is that each test facility is unique in their operation and they require dedicated IT tools for their functionality. All this means an increase in *No. of IT tools in PD,* resulting in reduced program efficiencies in the long run (with a delay) and an impact to the *Program meeting target dates;* another undesired reinforcing loop.

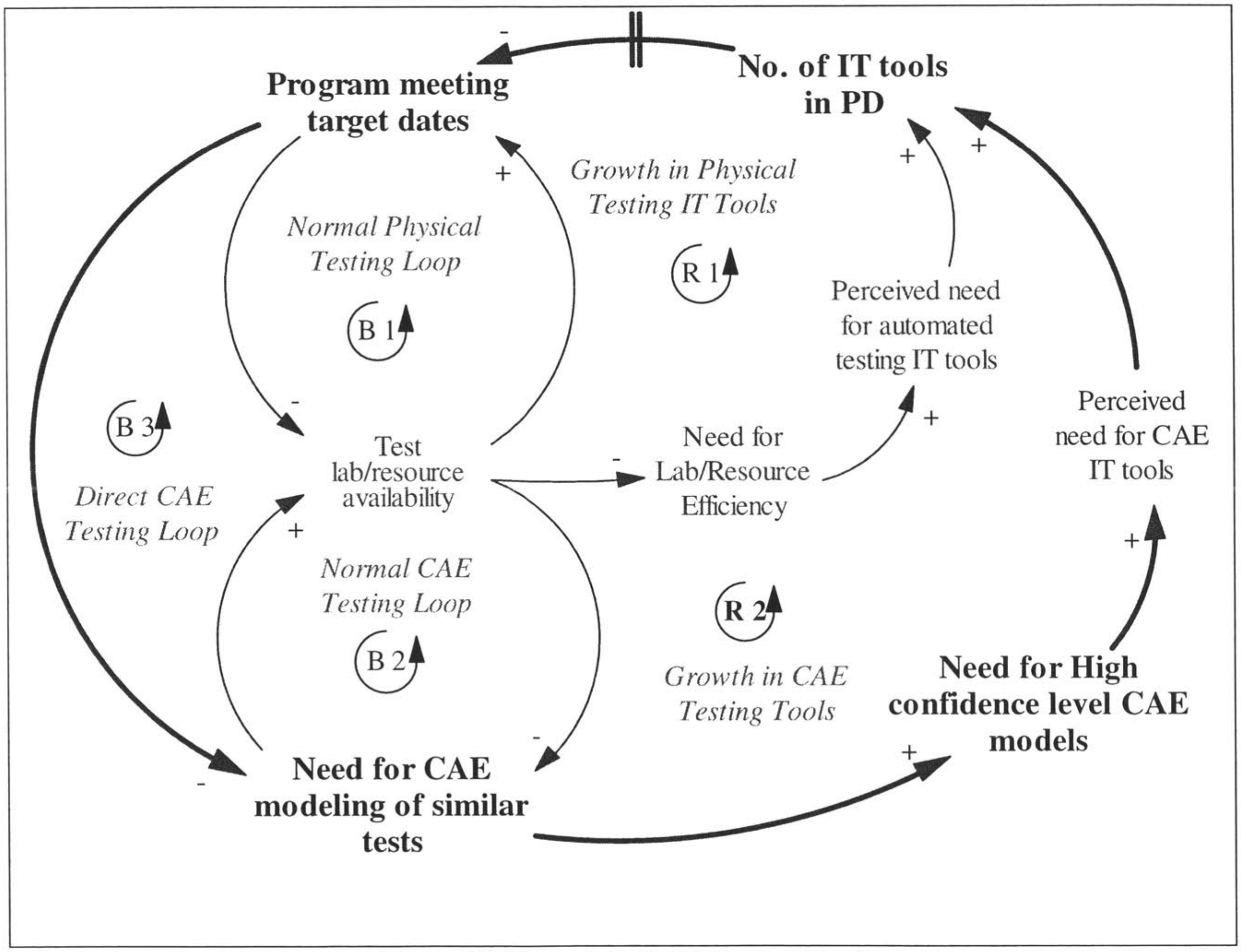


Figure 20: Growth in CAE Testing Tools

Growth in CAE Testing Tools (R2): It is a similar situation on the CAE testing world too. A need emerges for new CAE models (*Need for CAE modeling of similar tests*); models that depict the characteristics of the real-world physical testing. To simulate such real-world dynamics in the CAE model means that the model needs to demonstrate a high degree of accuracy. CAE engineers need to be confident (*Need for High confidence level CAE models*) to execute CAE based tests without the need for a physical test. In many cases this boils down to the need for (*Perceived need for CAE IT tools*) new improved high functionality CAE-IT tools to develop CAE models faster. All this means an increase in *No. of IT tools in PD,* resulting in reduced program efficiencies in the long run (with a delay) and an impact to the *Program meeting target dates;* another undesired reinforcing loop.

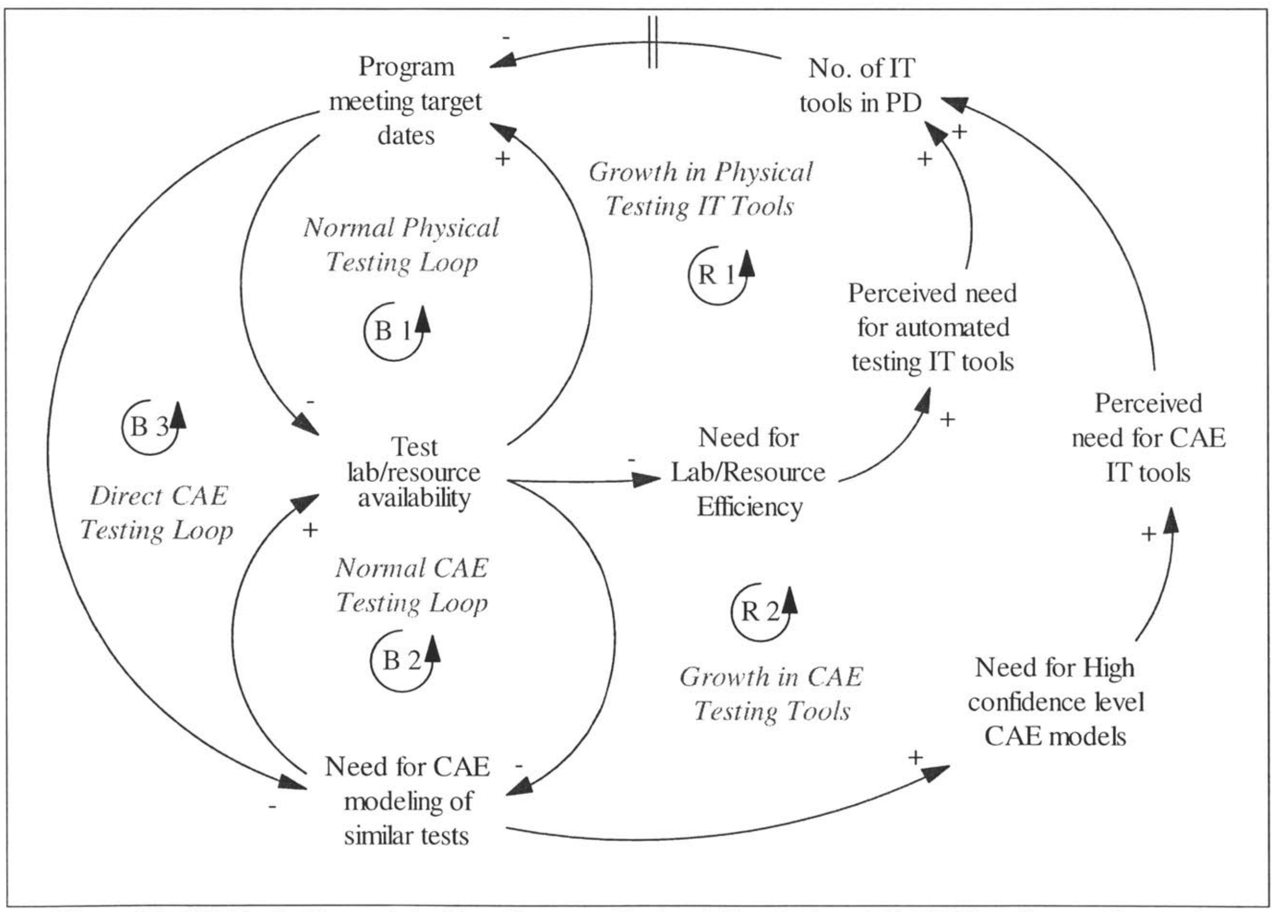


Figure 21: IT Tools Growth in Test Sector - Derived Physical and CAE Test Sector Dynamics

Possible Interventions

1. Reduce IT tools to avoid annual maintenance and support costs
2. Perform IT tool development governance to avoid recreation of new tools just because one program needs it
3. Avoid silo based solutions and treating every solution as a unique solution
4. Find the sweet spot between CAE and Physical testing to control costs and behaviors
5. Reduce physical testing activity to reduce program costs and timing
    a. Control funding for prototypes - Forces engineers to come up with creative solutions on how to use CAE instead of using costly physical testing methods
    b. Promote usage of CAE models as first step
6. Program governance through process changes – reduce availability of physical prototypes early and provide resources for CAE methods

We will see how some of these key interventions can be visualized through case studies in chapter 5.

(THIS PAGE IS LEFT INTENTIONALLY BLANK)

CHAPTER 4
OVERVIEW OF SYSTEM DYNAMICS TOOLS

Brief History

Systems thinking & system dynamics have been effective tools in automotive sector because of the complex nature of the business. Peter Senge in his famous book: "The Fifth Discipline" has demonstrated how systems thinking helped the Engineers and Project managers on a Lincoln program. He quotes a specific example related to the Electrical power in one of the Lincoln vehicle program in his "The Fifth Discipline: Field book". Many program managers and planners have taken on system thinking to understand the core problems associated within systems engineering at Ford Motor Company.

Pugh-Roberts Associates (later on known as PA consulting) have several years of experience in modeling. They have used system dynamics modeling on government contracts to fight legal battles for projects that over-run cost budgets. This experience helped them to move into automotive program management in England.

Jaguar cars program management office started using system dynamics models to understand their quality and productivity problems in managing of the vehicle programs. After getting the system dynamics models developed and validated, the vehicle program planners were able to understand where their main problem areas were in PD. The next generation of vehicle programs that used this system dynamics tool have realized the benefits associated with using such models. Benefits like on-time delivery, accurate forecasting of resource requirements, predicting physical testing load on test facilities etc.

System Dynamics in Product Development

Program management simulators have helped PD firms in a big way. Lately the software development firms are also introducing project management system dynamics models to understand the dynamics of the software project teams and the project performance long before they can initiate a product program. I will be discussing about 2 such program simulators here in this section, one that is used specifically in PD and the other that is used to plan software projects.

Dynamic Program Simulator

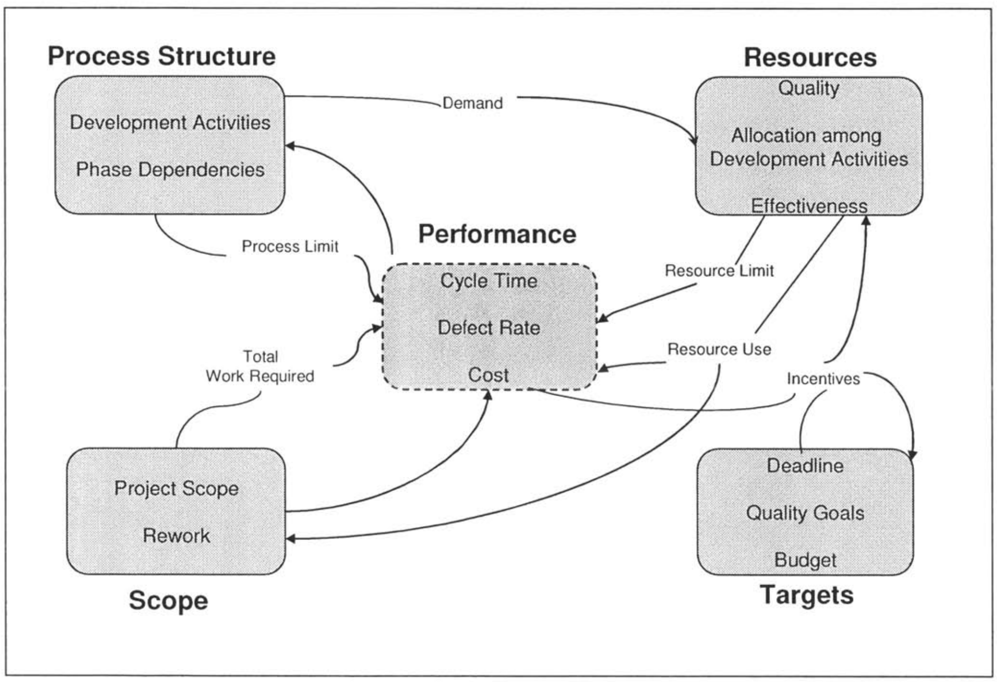


Figure 22: Phased Sub-systems (Source: *Sterman & Ford 1998*)

During my investigation, I found a system dynamics tool in the area of PD and Program management, which is called the Dynamic program simulator (DPS), developed by PA consulting. The DPS is a powerful tool and provides functionality that is equivalent to Project Management Flight simulator - that is, it helps program teams to manage programs better by evaluating in advance the likely consequences to program performance of varying program assumptions, risks and mitigating actions. The DPS incorporates a system dynamics model of the program - simulating resources needed and applied, productivity and its determinants, rework creation, discovery and execution, progress estimates, timing revisions, interdependencies between program activities, and workforce dynamics.

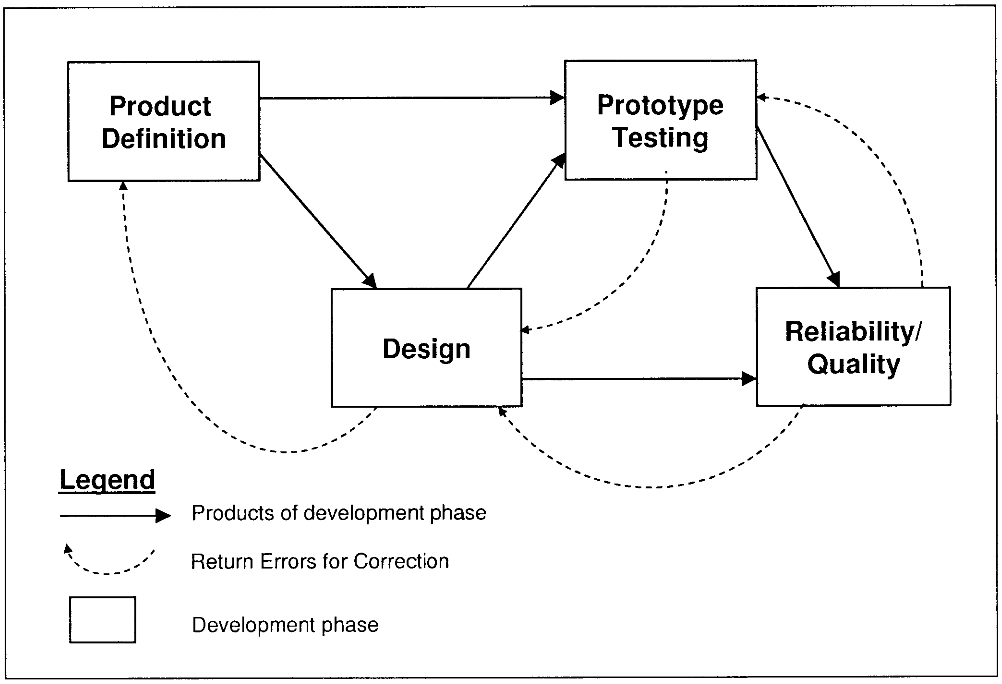


Figure 23: Project Network (Source: *Sterman & Ford 1998*)

DPS is a multiple phase development project tool with internal process concurrence constraints set at each phase. A most generic version of this type of tool was explained (refer to Figure 22) by John D. Sterman and David N. Ford through their phased sub-systems approach. DPS follows the same theme that was explained by John D. Sterman and David N. Ford in their paper titled "Dynamic modeling of product development processes" - 1998, wherein a product development project model should explicitly model all four performance drivers: process structure, resources, targets and scope.

Similar to the Sterman-Ford Project network diagram as shown in Figure 23; DPS separately and explicitly simulates many high-level vehicle program activities, along with related staffing and management decision-making. Engineering phases include the target setting and cascade (Product Definition), optimization and verification (Design & Testing) and Prototypes/Production releases (Prototyping & Release). Appearance design is also represented separately.

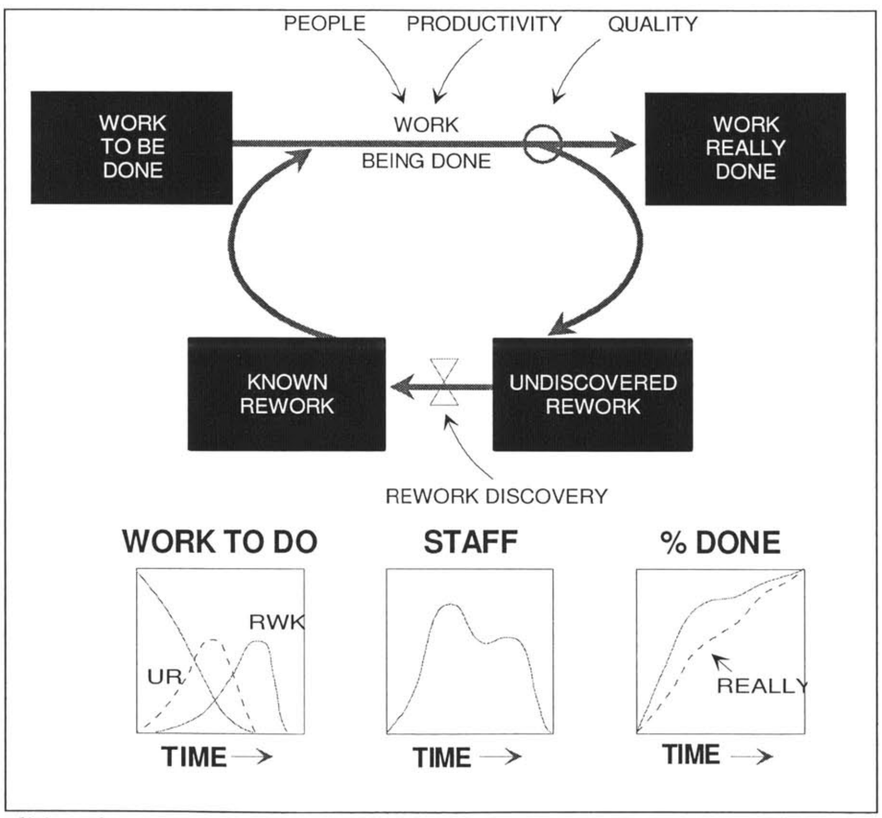


Figure 24: Traditional Project Management Work Accomplishment Stock Flow Structure (Source: *ESD.36J Course Lecture Notes from James Lyneis (Lyneis, 2002)*)

The "Rework Cycle" structure shown in Figure 24 is used to simulate how engineering work actually get done. It reflects the fact that engineering generally involves rework, and that it is frequently unclear at any given time which tasks will need to be redone. All new work scope starts off in the backlog on the left side of the diagram, called "*Work To Be Done*". This backlog is worked off at a rate (Work Being Done) that is determined by the People and/or other resources applied, and by the Productivity at which they work. Productivity is defined here as the output per unit of effort, and so has units of, for example, Production releases per staff-hour.

Referring to Figure 24, at any point in time, a fraction of the work performed is done for good, and never need to be revisited. This fraction is called the current Quality, which therefore reflects the correctness and completeness of the work. Quality determines how much of the work currently being performed goes into the Work Really Done backlog versus the Undiscovered Rework backlog. For example, if quality is currently at .75, then three quarters of the work being performed goes into Work Really Done and one quarter goes into Undiscovered Rework. Note that Productivity and Quality vary significantly over the course of the program in response to the conditions under which program work is performed.

"Undiscovered Rework" becomes "Known Rework" by a variety of rework discovery mechanisms, which may include, for example, the execution of downstream engineering work, the building of physical prototypes, and/or component or vehicle testing. Once in the "Known Rework" backlog, this rework is part of the visible work that remains to be done on the program, to which resources can then be allocated.

Some of the known factors that will be affecting the work "Productivity" and "Quality" and that are part of the model include:

- Quality of upstream work in current phase - e.g. poor vehicle level targets propagate quality problems down to component level targets
- Availability/quality of prior phase work- e.g. availability/quality of targets affects optimization and Prototype/Production release work
- Staff skill/experience level - to a certain point, more experience benefits Productivity/Quality
- Appearance design progress - engineering work depends on it
- Schedule pressure - it may increase Productivity, but generally at a cost to Quality
- Quality pressure - acts oppositely, helping Quality but hurting short term Productivity
- Out-of-sequence work - working out of the ideal sequence hurts Productivity/Quality
- Overtime - it does add equivalent heads, but eventually hurts if sustained long term
- Program management - senior management diverted by fighting fires hurts performance
- Morale - Productivity/Quality are helped by good morale, hurt by poor morale
- Changes in organizational size - getting used to new reporting relationships and logistics can consume time (hurting productivity), and impact Quality adversely
- Contract labor - though use of specialist labor may be effective, broad dependence on contractors may cause problems due to lack of corporate familiarity/motivation

The tool allows product planners and strategists to understand and analyze the "What if?" conditions. Conditions like -

- What if the timing changes in development program?
- What if the resources are hard to get for this program?
- What if the design from a related concurrent program is late?

- What if key suppliers get into difficulties meeting the program timeline?
- And what if a significant design change is necessary for the program?
- How do you rapidly evaluate the impact on your program?
- Even more importantly, how do you determine - in advance of committing to action - which mitigation strategy will be most effective?

DPS requires that the user have a certain degree of knowledge in order to realize full benefits from its use. For this purpose a Program Management Cookbook system interface is made part of this tool, user is driven through a set of very structured screens to input his program/project criteria, also the knowledge that is required to effectively use the DPS under most common situations is briefly illustrated through these screens.

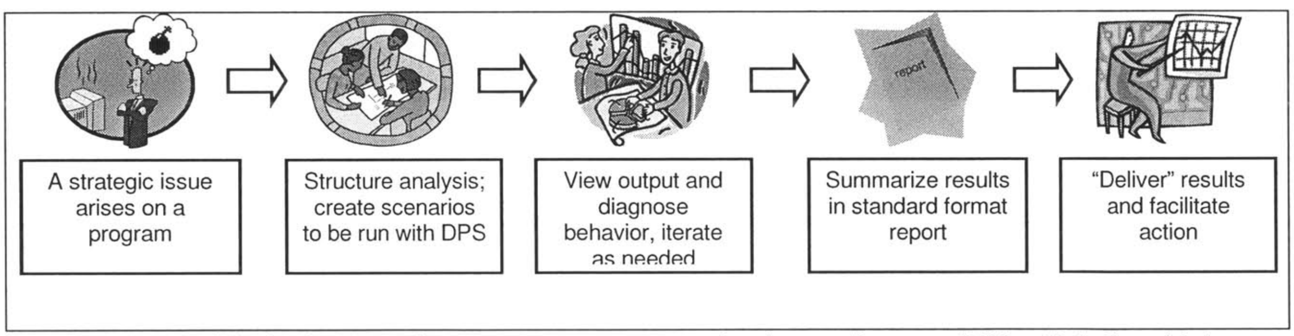


Figure 25: Typical usage of program management simulators (Source: *DPS – PA Consulting*)

The scenarios are created within 3 main sections: Program Assumptions, Risks, & Mitigating Actions.

Program Assumptions: In order to define a vehicle program, the team needs to define basic goals of the program.

Program Type	Program Description	Duration (Months)	Resources (Staff Months)
Type 1	Updated Trim – A small change to the vehicle with a complete carry over of the engine and transmission components.	21	1000
Type 2	Minor Vehicle Refresh – A minor change to the vehicle exterior or interior with carry over of the power-train.	28	1600
Type 3	Moderate Vehicle Refresh – A significant update to the vehicle interior and exterior with minor changes to the engine and transmission.	38	2400
Type 4	Major Vehicle Refresh – New vehicle exterior with a carry over platform and/or a new power-train variant and emissions controls.	43	4000
Type 5	New Vehicle Platform – This program type incorporates a brand new vehicle platform and a new engine and transmission combination.	50	6500

Table 4: Reference Program Definition (Source: *Program Types - Edward Esker, 2000 SDM Thesis*)

Program Assumptions include:

- Program timing – Based on the level of the program or the program type (refer to Table 4)
- Work Scope – Work content in different phases. For example; new tooled end items that are released from design to supplier for build. New tooled end items are classified as parts that need new tooling for prototypes or production releases.

- Engineering Staffing – Planned engineering budgets and resource availability to the program. In many cases a program may not be able to get all the required staff and may even be capped by HR due to company hiring policies.

- Investment – Budget allocated to the program

- Other known initiatives or strategies – For example CAE tools will be used extensively in the upfront design and analysis.

Risks: are defined as those known or unplanned events and conditions which adversely impact program performance. Risks commonly impacting programs tend to arise from five primary sources – aggressive program assumptions, unfamiliar technologies, organizational changes, resource constraints, and late changes to the program.

Mitigating actions: are more like key leverage variables in the DPS tool and they fall into four primary categories:

1. Disciplined program management

2. Leveraging suppliers

3. Staffing policies and

4. Timing changes

These policies are to be in place for a program to take advantage of these key triggers variables. A good program manager may have to key this on their program charter before the program is kicked-off.

Software Development Model

Similar to the DPS tool, Prof. Tariq Abdel-Hamid has developed a model that simulates the behaviors associated with Software (SW) development practice. This comprehensive system dynamics model was developed to understand the key problems faced by software development houses in the late 1980s and early 1990s. This model was developed with 4 sub-systems similar to the phased systems as discussed in earlier sections. The 4 sub-systems are:

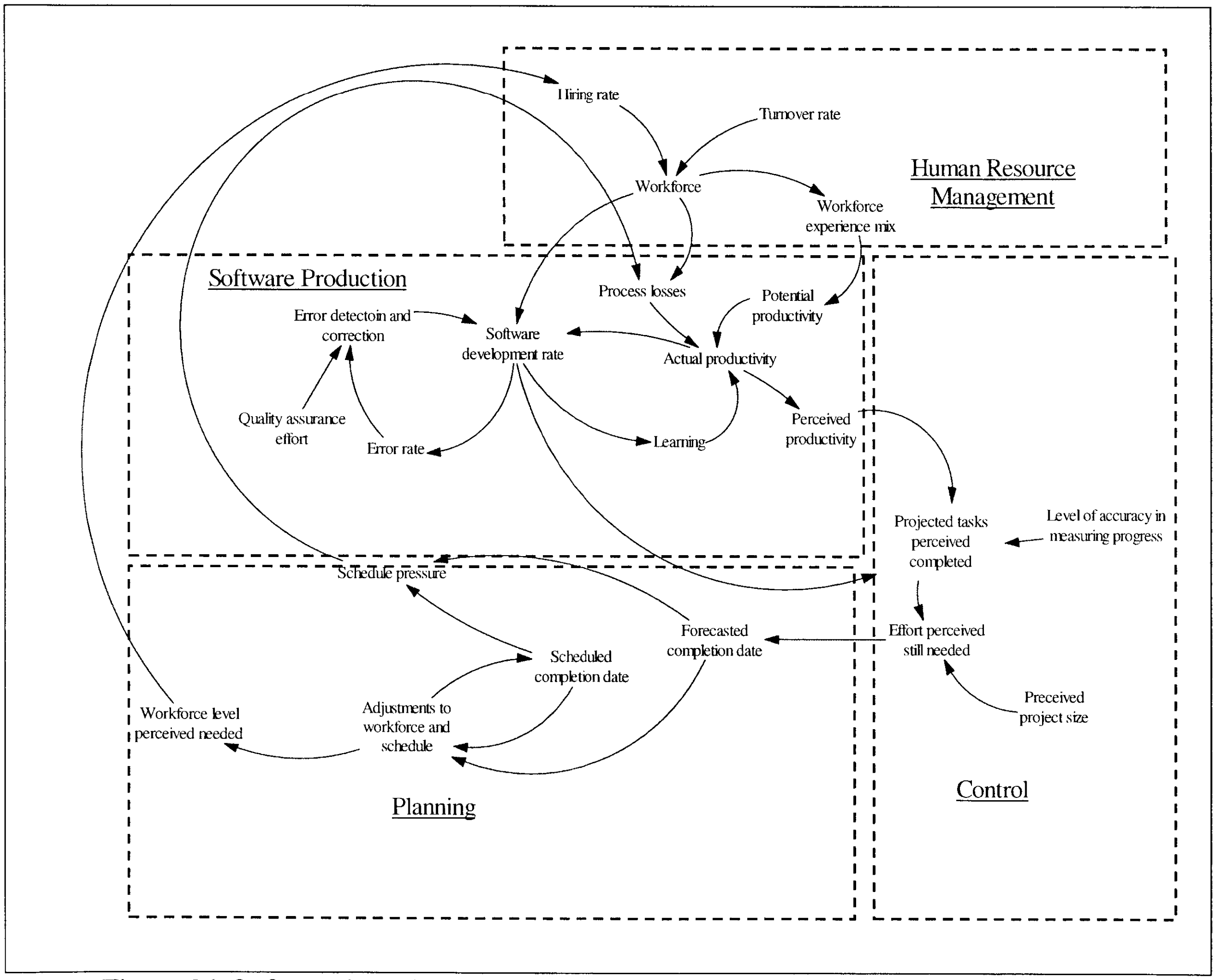


Figure 26: Software Development Model Structure (Source: *Abdel-Hamid & Madnick*)

1. **Human resource management**: Consists of the typical rookie-pro employee types and their behavioral dynamics within a project/program.
2. **Software production**: Consists of the software development life cycle dynamics. This sub-system has different phases of SW development like design, build, quality assurance (QA) and testing. The most interesting phases of SW development that are addressed in the model are the QA and testing.
3. **Control**: This sub-system is primarily focused on project management, quality measurement, project timing and other decision maker's actions.
4. **Planning**: This sub-system is used in upfront project definition like timing, scoping, resource allocation etc. And also used through out the project life-cycle for taking actions like increasing program staff, adjusting timing etc.

Several simulation runs were performed on an enhanced SW development system dynamics model, developed by John Tvedt (John Tvedt's PhD Thesis dissertation can be found in http://www.eas.asu.edu/~sdm/tvedt/dissertation.). This model is an extended version of the Abdel-Hamid & Madnick model that has effects of inspections (Stage gate process) on SW development.

Model results showed that inspections improved the overall project performance in terms of resource requirements, project quality and project completion. These simulated results are widely available in the public domain for review. The main takeaways from this analysis are tabulated in Table 5 below.

	Without inspections	With inspections	Difference
Project Cumulative Man-days (Resource needed)	3937	3222	-715
Total tasks under inspection	154	3.4	-150.6
Tasks completed	912	1056	144

Table 5: Simulated results: With & without inspections during software development

(THIS PAGE IS LEFT INTENTIONALLY BLANK)

CHAPTER 5
CASE STUDY: SIMULATING THE SYSTEM DYNAMIC MODELS

Approach

In chapter 3 we have visualized some of the key PD behaviors through causal loops and their effects on PD efficiency. In chapter 4 we have seen how the system dynamics tools have been developed in industry to visualize and forecast project dynamics, even before the projects or programs are implemented. In this chapter we will be able to visualize some of project dynamics through a scenario building process and evaluating some "What if?" conditions through two case studies.

Let us consider a hypothetical situation where PD business has to fix the problems identified in the study conducted in the chapter 3 (refer to causal loop diagrams in chapter 3). The decision makers are tasked to decide between the two initiatives and pick the one that makes the best business sense. Let us consider the two "What if?" conditions:

1. What if we reduce the number of IT tools in the PD space?

2. What if we devise a process change to use CAE based testing methods (or CAE prototypes) early in the design process for faster execution of product verification?

Now let us consider how these two business initiatives (through two case studies) will impact the PD process efficiency in terms of cost, timing and quality.

Structure Analysis:

The system dynamics model that was developed for the current PDS system was used for this analysis. DPS tool was used to run the Current State of the system for a Level 5 program (refer to Table 4). The *Current State* simulation results were compared against the optimized runs (simulations) for both these cases. It was observed in the system dynamics model that by reducing the number of IT tools in the PD space improves the PD efficiency through an improvement in "Accelerate the rework discovery" (refer to Figure 27). It was also observed that to improve and promote the use of CAE methods the organization must first reduce the organizational dependence on physical prototypes for early design verification, this can be done by using past CAE models, IT driven knowledge databases with historic design and validation results, or just by using digital or physical mockups. Four key variables were identified in the model that supports these 2 hypotheses as listed below:

1. What if we reduce the number of IT tools in the PD space?

- Key leverage variables to observe –
    - *"Accelerate Rework Discovery"*

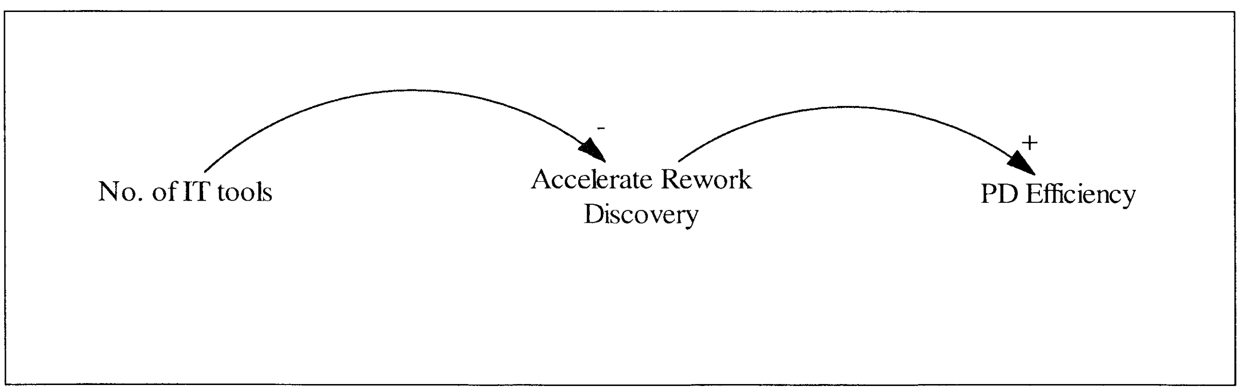


Figure 27: Effect of Accelerate Rework Discovery on PD Efficiency

2. What if we devise a process change to use CAE based testing methods (or CAE prototypes) early in the design process for faster execution of product verification?

- Key leverage variables to observe –

  ➢ *Reduce Physical Prototype Dependence*

  ➢ *Early CAE analysis*

  ➢ *Prototypes - Early Design Phase*

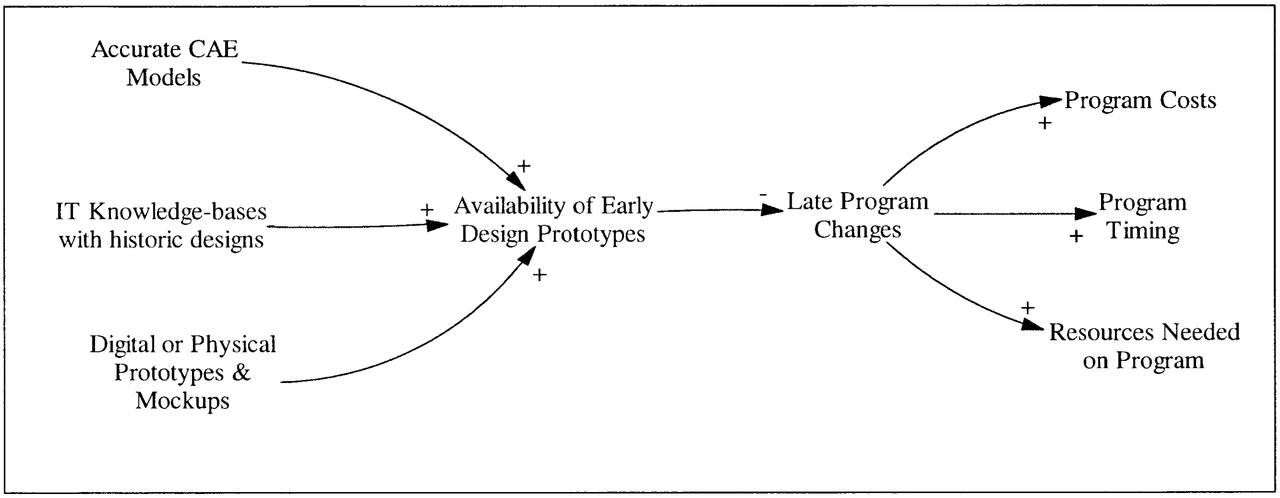


Figure 28: Effect of Availability of Early Design Prototypes on Program Metrics

For both the business initiatives common variables were identified in the system dynamics model. These common or generic input parameters are shown in the Table 6, and are the standard set of variables supplied for all the simulation runs. These variables basically let us specify the level of the program, key milestone dates (timing), scope (how many new designed parts are needed), funding allocated to the program (No. of prototypes needed at different phases of program), resource allocated (staffing) etc. The *Current State* generic variable values are carried over into the 2 business initiatives except for the key leverage variables values.

Explanation of the key leverage variables:

Accelerate Rework Discovery: Accelerating the rework discovery reduces the amount of time it takes to discover problems. Due to disconnected processes and IT tools, engineers may not have all the desired information at one place for a given issue (Discovered rework). This may lead to undesired effects in program timing and program costs etc. (This variable is a organizational capability multiplier)

Reduce Physical Prototype Dependence: Provide the capabilities or methods for engineers to reduce the dependence on physical prototype. This can be achieved through a robust analytical testing infrastructure, through improved IT knowledge databases of past product design information, through mimicking the physical testing processes and through CAE tools and methods and through increasing the capability of the CAE models (This variable is an organizational capability multiplier).

Early CAE analysis: Provide improved CAE tools to test new designs earlier, thereby greatly reducing dependence on traditional hardware build/test processes. CAE will support and improves processes for all three engineering phases modeled. A point to note here is that this variable helps in improvement in normal quality of engineering work and decrease in rework discovery times. However, it may result in reduction normal productivity since some extra effort is needed to develop these CAE tools (This variable is process dependent multiplier).

Prototypes - Early Design Phase: Early availability of design prototypes or mockups help reduce the effects of design iterations (Physical mockups or CAE morphed models). If engineers are provided early design intent prototypes, can improve their design productivity and reduce rework early in the design process (This variable is effort/usage variable).

Current State of the System under Analysis

To simulate the *Current State* of a system we need the following inputs as tabulated below:

Input parameter	Value	Units
Generic Variables		
Program Level	5	Number (1-5)
Program Timing		
Plan program end date (Job 1)	1/1/2002	Date
Milestones (Each design phase)	No Change	Months before Job 1
Engineering Start/End Timing	No Change	Months before Job 1
Prototype Start/End Timing	No Change	Months before Job 1
Test Start/End Timing	No Change	Months before Job 1
Work scope		
No. of New parts designed	950	Number
Percent reuse	70	%age
Prototype Needs / Phase		
*Early Design Phase**	2	Number (Effort)
Design Phase	54	Number (Prototypes)
Confirmation Phase	78	Number (Prototypes)
Launch Phase	141	Number (Prototypes)
Staffing		
Pre-staffing	50	Lookup
Staffing Max	400	Number
***Leverage Variables**		
Early CAE analysis	50	%age
Prototypes - Early Design Phase	2	Number (CAE Effort/Prototypes)
Reduce Physical Prototype Dependence	25	%age
Accelerate Rework Discovery		
Targets Setting	50	%age
Verification	50	%age
Production/Launch	25	%age

Table 6: Current state simulation - Required input parameters

The *Current State* is a true depiction of a real program that went through different phases of the PD process using the current PDS. The results of the *Current State* simulation will be considered as a base case for the next 2 cases. The most important variables to note in the Table 6 are:

1. The program level, which is set to 5, at which the program was executed

2. Program end date, which is set to 1/1/2002, is the proposed end date of the program

3. Percent reuse, which is set to 70%, of the parts and sub-systems in this particular program

4. Early design prototype, which is set to 2,

5. Staff Max, which is capped at 400.

The early design variable (a key leverage variable) is a numeric value for the CAE effort or No. of physical prototypes put in the early stages of design. The higher the number the more effort goes into this area. This can also be treated as the number of CAE models available, or number of non-working mock prototypes, or clay models etc.

Case Study 1: Accelerate rework discovery

The first business initiative is targeted to improve the PD engineering efficiency by helping design engineers to perform better on their job. Let's call this initiative as "BC-1". PD engineering efficiency can be derived from 3 common metrics – Cost, Time & Quality. One of the areas where engineers spend most of their time is in fire-fighting during design verification and rework discovery in each phase. BC-1 initiative affects the entire PD chain (refer to Figure 1 on Page 14) and specifically those areas that design engineers are exposed to, like Targets setting, Verification and Prototype build/Production launch.

One of the key leverage variables that are targeted by this initiative is the Accelerated rework discovery (ARD). ARD affects the Normal engineering rework discovery delay, which in turns effects the Engineering rework discovery delay (ERDD). If ERDD is reduced considerably, may help engineers to quickly detect problems before it is too late. ARD has 3 parts based on the 3 design phases and each phase can be assigned an optimal value.

Accelerating the rework discovery means a reduction of the amount of time it takes to discover problems. This multiplier is applied in addition to whatever assumptions are made for rework discovery. However, the multiplier can not exceed the capabilities of the existing state of the system. For instance the engineers can only use the CAE models developed for a certain part or a sub-system on which durability and stress analysis can be done. They may not have a model to do crash analysis. To perform crash analysis may require new capabilities in terms of new CAE IT tools and models.

100% of Early CAE analysis here means that the multiplier can reach the maximum available capability during the upfront analysis. An optimization was performed by keeping all variables same as shown in Table 6 except the key leverage variables listed in Table 7.

Input parameter	New Value	Old Value	Units
Leverage Variables			
Early CAE analysis	**100**	50	%age (Changed)
*Prototypes - Early Design Phase**	2	2	Number (Effort)
Reduce Physical Prototype Dependence	**100**	25	%age (Changed)
Accelerate Rework Discovery			
Targets Setting	**77**	50	%age (Changed)
Verification	**75**	50	%age (Changed)
Production/Launch	**100**	25	%age (Changed)

Table 7: Case Study 1 – Optimized key leverage variable values (*See bolded values*)

Results of simulation for case study 1

Based on the optimized simulation run, the most noticeable benefits like timing and costs that can be derived from BUSI-1 are tabulated in Table 8, Table 9, the resource allocation trend and program quality characteristics can be seen in Figure 29 and Figure 30 respectively.

Listed below are the findings of the simulated run for BC-1:

- Improvements in engineering program timing: Just 8 weeks late (better than *Current State* by 11 weeks)
- Improvement in rework discovery over *Current State* by 43%
- 22% Lower late revisions than *Current State*
- 16.5% less program hours than *Current State*
- A cost saving of at least $67 Million (On just one program)
- Also an improvement in engineering quality at all 3 phases of design & development

Summary			
	Current State	Optimized for Accelerated Rework Discovery	Difference
Planned job 1 date	1-Jan-02	1-Jan-02	0
Job 1 weeks late	19.6	8.2	-11.4
Planned FER date	31-Aug-00	31-Aug-00	0
Engineering Release weeks late	10.9	2.7	-8.2
Planned Confirmation build start	1-Oct-00	1-Oct-00	0
Confirmation Build weeks late	9.8	4.9	-4.9
Undiscovered rework at J1	9.9	5.6	-43.50%
Late revisions Build to J1+90	1831.7	1420.9	-22.40%
Total program hours (000)	2,622	2,190	-16.50%

Table 8: Summary of program timing - Case Study 1

Costs			
(000's)	Current State	Optimized for Accelerated Rework Discovery	Difference
Engineering Labor	131,115	109,479	(21,636)
Prototype tooling	19,695	22,412	2,717
Prototype vehicles/testing	27,500	27,500	0
Production tooling	324,081	297,265	(26,816)
Launch	88,847	67,359	(21,488)
Total fixed cost	594,438	527,215	(67,223)

Table 9: Summary of Costs - Case Study 1

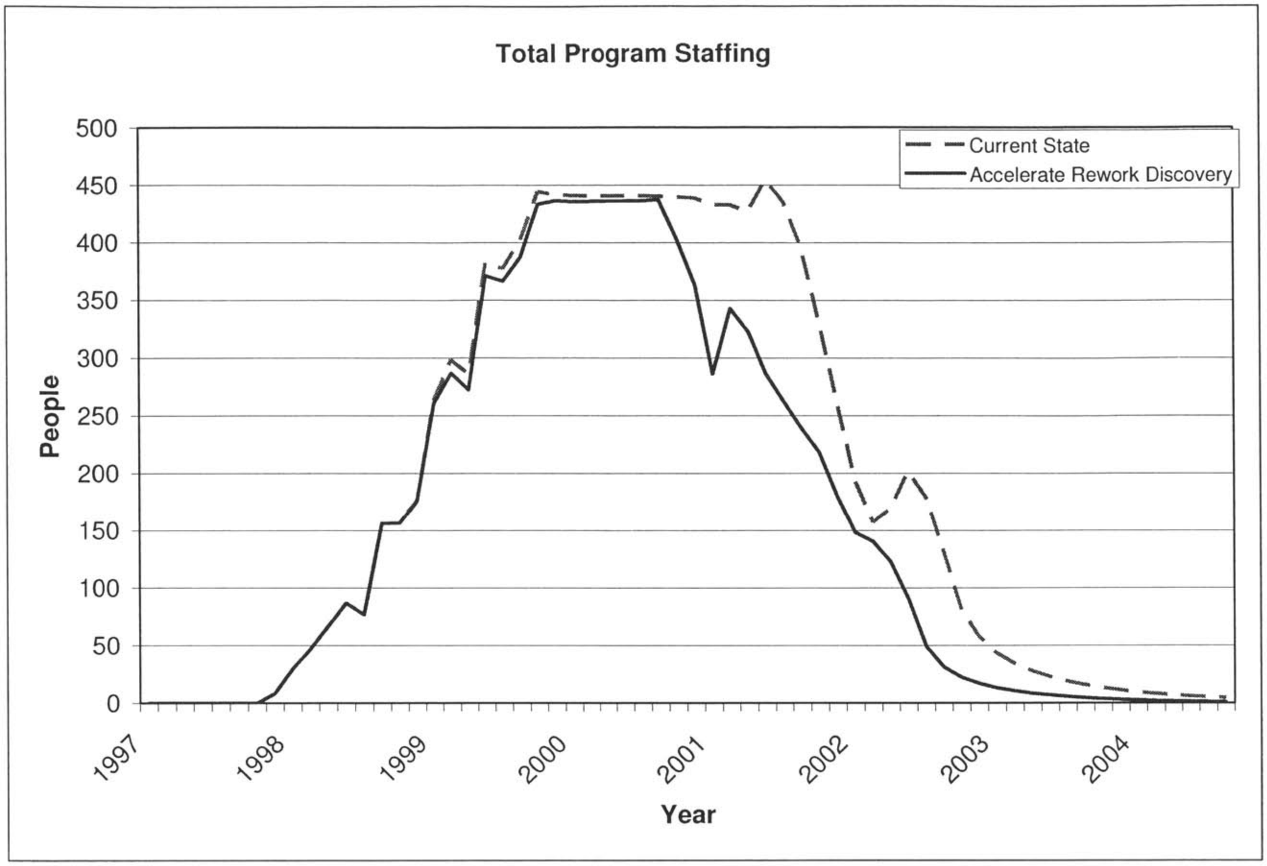


Figure 29: Total program staff - Case Study 1

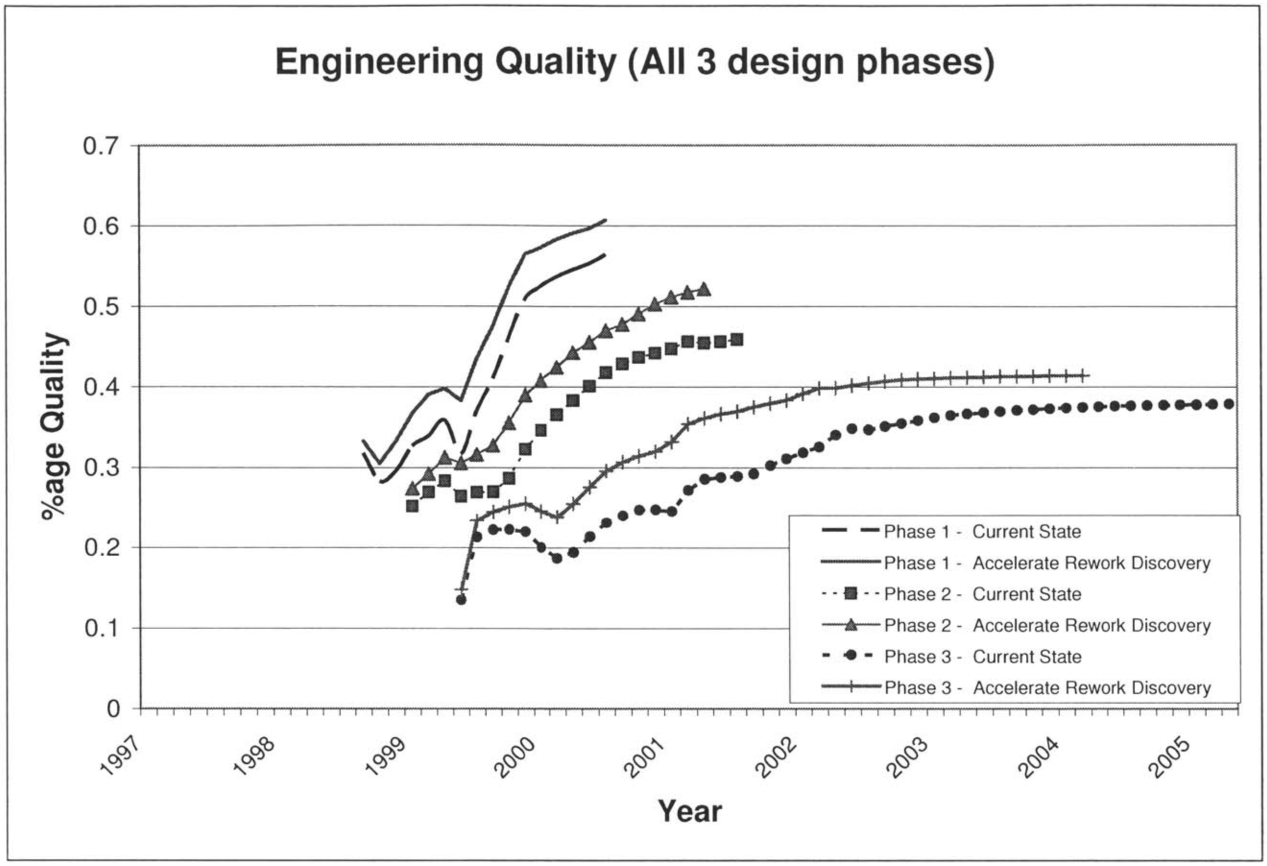


Figure 30: Engineering Quality for all 3 Design Phases - Case Study 1

Case Study 2: Increased availability of early design prototypes

The second business initiative is to take advantage of early design prototypes during the pre-design and design phases of PD. Let's call this initiative as "BC-2". CAE is a very robust means to validate the optimal design targets before committing to the metal parts. This helps all the program metrics (cost, timing and quality) because new designs are not released for tooling. This also minimizes the need for expensive and time consuming physical builds and physical verification testing. PD can realize dramatic improvement in the program timing by avoiding a lot of early churn.

One of the key leverage variables that are targeted by this initiate is the *Prototypes - Early Design Phase* (P-EDP). P-EDP affects the Initial production backlog, which in turns affects the early design intent prototypes (Physical mockups or CAE morphed models). If engineers are provided early design intent prototypes, they can improve their design productivity and reduce rework early in the design process.

An optimization was performed by keeping all variables same as shown in Table 6 except the key leverage variables listed in Table 10.

Input parameter	New Value	Old Value	Units
Leverage Variables			
Early CAE analysis	**100**	50	%age (Changed)
*Prototypes - Early Design Phase**	**20**	2	Number (Changed)
Reduce Physical Prototype Dependence	**100**	25	%age (Changed)
Accelerate Rework Discovery			
Targets Setting	50	50	%age
Verification	50	50	%age
Production/Launch	25	25	%age

Table 10: Case Study 2 – Optimized key leverage variable values (*See bolded values*)

Results of simulation for case study 2

Based on the optimized simulation run, the most noticeable benefits like timing and costs that can be derived from BC-2 are tabulated in Table 11, Table 12, and the resource allocation trend and program quality characteristics can be seen in Figure 31 and Figure 32 respectively

- Improvements in engineering program timing: Only 3 weeks late (better than *Current State* by 16 weeks)

- Improvement in rework discovery over *Current State* by 48%
- 31.8% Lower late revisions than *Current State*
- 24.6% less program hours than *Current State*
- A cost saving of at least $83 Million (On just one program)
- A steep reduction of engineering staff results in the tail end of the PD process when compared to the *Current State*
- Also an improvement in engineering quality at all 3 phases of design & development

Summary			
	Current State	**Early CAE & Prototype Analysis**	**Difference**
Planned job 1 date	1-Jan-02	1-Jan-02	0
J 1 weeks late	19.6	3.3	-16.3
Planned FER date	31-Aug-00	31-Aug-00	0
Engineering Release weeks late	10.9	-0.5	-11.4
Planned Confirmation build start	1-Oct-00	1-Oct-00	0
Confirmation Build weeks late	9.8	1.6	-8.2
Undiscovered rework at J1	9.9	5.1	-48.00%
Late revisions Build to J1+90	1831.7	1248.8	-31.80%
Total program hours (000)	2,622	1,978	-24.60%

Table 11: Summary of program timing - Case Study 2

Costs			
(000's)	**Current State**	**Early CAE & Prototype Analysis**	**Difference**
Engineering Labor	131,115	98,922	(32,193)
Prototype tooling	19,695	22,544	2,849
Prototype vehicles/testing	27,500	27,500	0
Production tooling	324,081	293,974	(30,107)
Launch	88,847	65,149	(23,698)
Total fixed cost	594,438	511,290	(83,148)

Table 12: Summary of Costs - Case Study 2

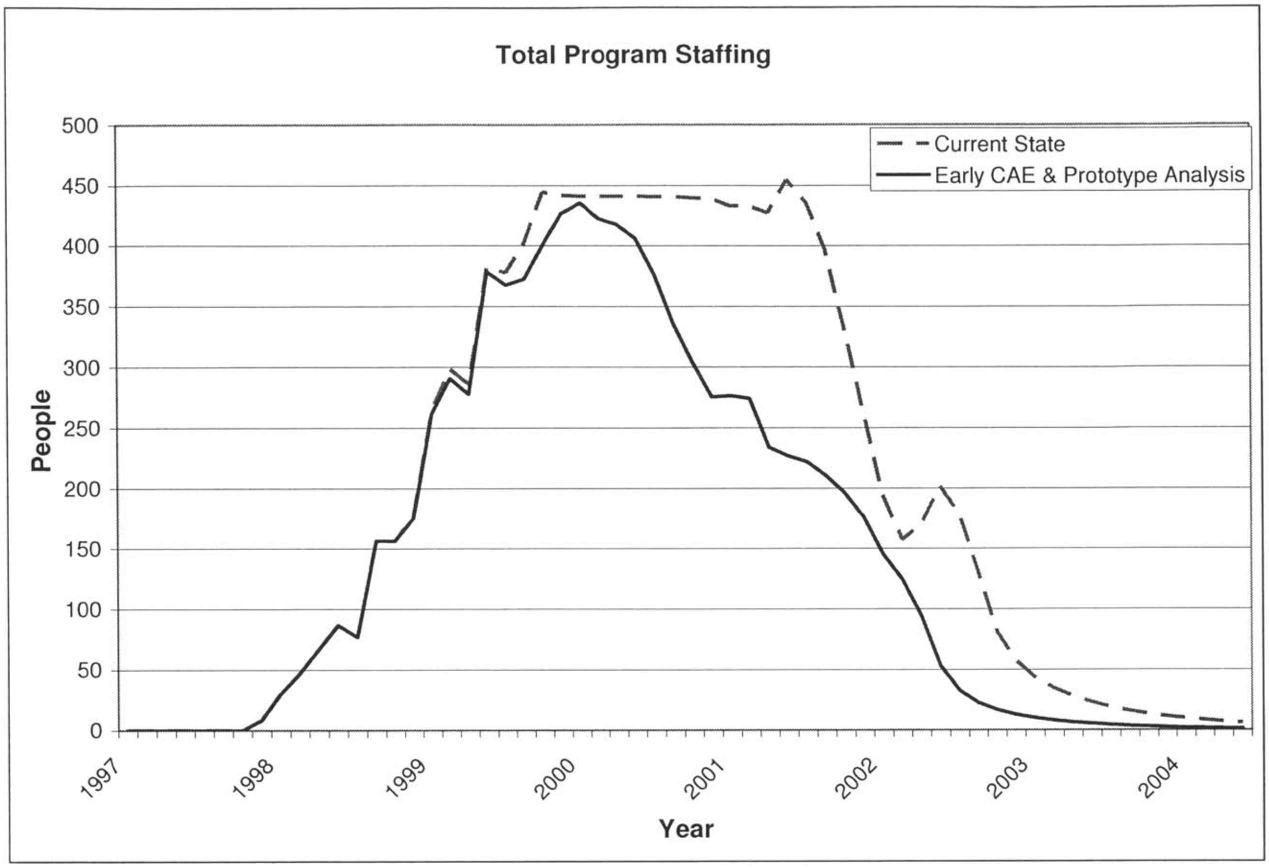


Figure 31: Total program staff - Case Study 2

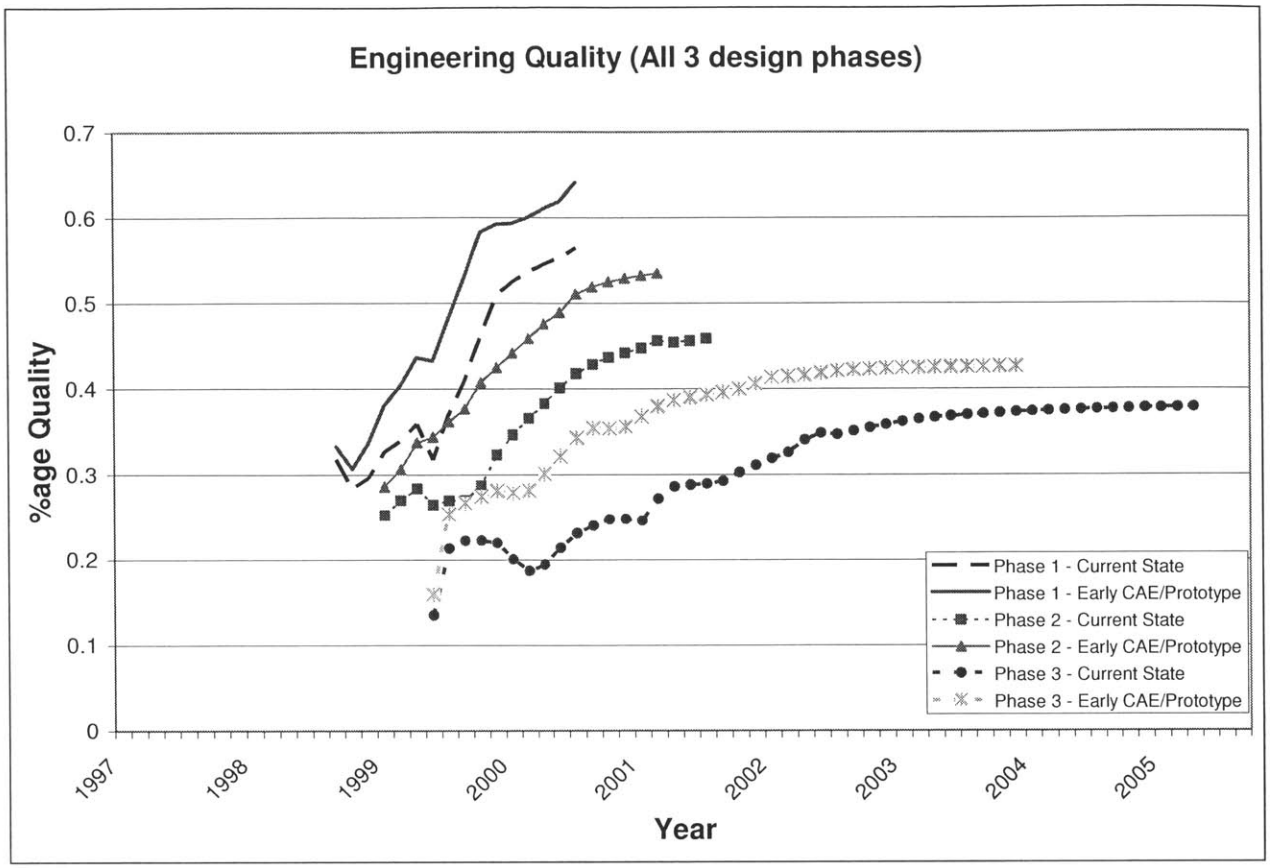


Figure 32: Engineering Quality - Case Study 2

Possible Interventions

If we compare both the cases it makes more sense to introduce the policies around increase in early design prototypes or BC-2. I have also run a simulation combining the key leverage variables from both the cases BC-1 and BC-2. The simulation resulted in very minimal improvements to the engineering metrics. The next step is to identify what is the real cost of implementing anyone of these 2 initiatives. Also, risks of implementing and not implementing these initiatives in a timely manner should be understood in terms of market dynamics, financial condition of the company and engineering competencies.

(THIS PAGE IS LEFT INTENTIONALLY BLANK)

CHAPTER 6
RECOMENDATIONS

Systems thinking and system dynamics have helped PD in many ways. Authors like Peter Senge and John Sterman have contributed immensely to this area; also combined with thousands of industry practitioners have contributed hundreds of thousands of literary pieces for the advancement of this field. Systems thinking have helped many complex organizations to understand the core problems and system dynamics has helped to validate those findings through a measurable set of data and models.

Many key leverage points were discussed in the past chapters both in the PD and IT side. However, all organizations cannot fully leverage these levers because of the inherent financial and operational structures. But, an effort has to be made to understand which levers best suit the organization and the value they will deliver.

For the PD organization to fully take advantage of the future of information systems, man power need to be trained to the new ways of doing business. Eliminate past processes, those that were created when technology was at infancy and the processes made sense in the past. Value stream mapping a key concept in Lean thinking will help organizations to understand where the process delays are and these process delays can be embedded in system dynamics models to understand the real problems associated with the legacy processes.

IT tools are just a replication of those legacy processes. Digitization of a process is merely analogous to have a scan or photo-copy of a paper document into a computer. Effort has to be made by business leaders and subject matter experts to understand the capabilities of the information systems before they invest huge sums of money into the implementation of those systems. In many cases combining efforts of multiple departments may reduce the money and effort spent on implementing new tools and methods.

Governance when creating new IT tools is seen as a major problem. Cutting the red-tape is favored in most organizations to get tasks done quicker, and these behaviors are awarded. Of course, this type of behavior is needed where processes are legacy and will create "MUDA" (Waste in the process). However, inculcating this behavior and getting used to those activities may be fatal for the organization in a long run if they are not controlled. It proves out to be a double-edged sword.

As seen in the IT software development model, governance of software development activities through either inspections and through gate-reviews has yielded better results in quality and productivity of software development firms. But, as the complexity of the software or the product grows these activities may see to be more of a bottleneck then prove effective. So, the recommendation would be to find the optimal point or sweet spot for governance.

The Dynamic program simulator (DPS) tool as discussed in this study is a stand-alone tool used for managing a single program at a time. However, in the real-world this tool is not ideal, since most organizations have multiple programs in the pipe line and they may be dependent upon the same resources (People, facilities, suppliers etc). To accommodate these new requirements of multiple program management, DPS has naturally evolved in a tool called Dynamic portfolio manager. However, to manage a tool that needs inputs from multiple programs needs a lot of man power or process automation. This is a problem in a huge organization like Ford Motor Company.

While this study was under way it was found out that the PD & IT management has taken an initiative to put controls on IT tool purchases (IT governance) to minimize the effects of proliferation of redundant IT tools.

(THIS PAGE IS LEFT INTENTIONALLY BLANK)

CHAPTER 7
CONCLUSIONS

Many new concepts were learnt during the process of this exercise. A well executed product development activity will lead to a better product and a better experience. System dynamics simulators and models will definitely help visual complex systems quickly and help understand the dynamics of any complex system quickly with out the need for extensive research. However, caution has to be observed to fully trust these models since models are representation of a point in time activity and the parameters that went in to these models. In any case, system dynamics if used as a tool will help any decision maker to quickly come to an understanding of the system.

During the course of this study I was fortunate to participate in the development of an integrated product development system. This study helped me immensely to relate to the advantages and disadvantages of the currently developed integrated product development system. It was also observed that subject matter experts tend to make improvements inside their domain without understanding their effects to the system on a whole. In many cases it was also observed that improvement in timing in certain areas has very minimal to no effect on the total program timing.

This study not only gave me a holistic understanding of the PD process but also a new perspective of how to plan and manage the software development processes. This study can be extended into other areas of an organization like finance, purchases, marketing, supply chain etc. And there is a huge value to educate and communicate the systems thinking and the system dynamics concepts to any learning organization. In conclusion, and from a holistic systems thinker's mind, system dynamics concepts and tools are forever here to stay.

(THIS PAGE IS LEFT INTENTIONALLY BLANK)

BIBLIOGRAPHY

1. Doe, John B. 1996 "Conceptual Planning: A Guide to a Better Planet - 3d ed." Reading, MA: Smith Jones.

2. Doebelin, Ernest O. 1998, "System Dynamics: Modeling, Analysis, Simulation, Design" (eBook) New York Marcel Dekker, Inc.

3. Ford, David N. and Sterman, John D. 1998. "Dynamic Modeling of Product Development Processes", System Dynamics Review, 31-68.

4. Hundal, Mahendra S. 2002, "Mechanical Engineering Publication" New York Marcel Dekker, Inc.

5. Lyneis, James, 2002. Course Lecture Notes and discussion in ESD.36J, System Project Management course, MIT, Cambridge, MA.

6. Mechanical Life Cycle Handbook: Good Environmental Design and Manufacturing (eBook)

7. Repenning, N. (2001). Understanding Fire Fighting in New Product Development, Journal of Product Innovation Management, 18, 5: 285-300

8. Repenning, Nelson, 1996. "Reducing Cycle Time and Development Time at Ford Electronics", Part II: Improving Product Development, MIT Case-study.

9. Repenning, N. (2000). A Dynamic Model of Resource Allocation in Multi-Project Research and Development Systems, System Dynamics Review, 16, 3: 173-212.

10. Senge, Peter M. 1990. "The Fifth Discipline: the Art and Practice of the Learning Organization",Currency Doubleday, New York, ISBN 0-385-26095-4.

11. Sterman, John D. 2000. "Business Dynamics: Systems Thinking and Modeling for a Complex World", McGraw-Hill, Boston, ISBN 0-07-231135-5.

12. Sterman, John D. 1992. "System Dynamics Modeling for Project Management", System Dynamics Group.

13. Smith, Chris. 1997 "Theory and the Art of Communications Design." State of the University Press.

14. Thomke, Stefan Experimentation Matters: Unlocking the Potential of New Technologies for Innovation HBSP June 2003.

15. Tvedt, John D, 1996, "An Extensible Model for Evaluating the Impact of Process Improvements on Software Development Cycle Time", Ph.D. Dissertation, Arizona State University.

16. Ulrich, Karl T. and Eppinger, Steven D. 2000. "Product Design and Development", McGraw-Hill, Boston, ISBN 0-07-229647-X.

17. Utterback, James M. 1994. "Mastering the Dynamics of Innovation", Harvard Business School Press, Boston, ISBN 0-87584-740-4.

18. Womack, Jones, Roos. 1990 "The Machine that change the world"

19. http://www.netlibrary.com/Reader/

REFERENCES

i The Machine that change the world; by Womack, Jones, Roos; 1990

ii The Industry Handbook - The Automobile Industry; http:// www.investopedia.com

iii Reuters Business Insight – August 2004