Hardware Engineering Change Management: An Enterprise Analysis of Factors Contributing to Technical Change

By

Matthew T. Knight

B.S. Physics
United States Naval Academy (1999)

M.A. National Security and Strategic Studies
United States Naval War College (2012)

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

May 2013
[JUNE 2013]

Signature of Author:
System Design and Management Program
10 May 2013

Certified by:
Olivier L. de Weck
Associate Professor of Aeronautics and Astronautics and Engineering Systems
Thesis Supervisor

Accepted by:
Patrick C. Hale
Senior Lecturer, Engineering Systems Division
Director, System Design and Management Program

This page has been intentionally left blank

Hardware Engineering Change Management: An Enterprise Analysis of Factors Contributing to Technical Change

by

Matthew T. Knight

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

Master of Science in Engineering and Management

Abstract

Engineering change management (ECM) is an essential but challenging cross-functional discipline within modern product development firms. ECM is best explained as a discipline because no single process can characterize the complex interactions between stakeholders, processes, information systems, knowledge management practices and cultural factors that enable the control of technical design change. One major challenge to product development projects is gaining actionable *a priori* insight into the risk of technical design change in order to allocate resources to mitigate specific risks. This thesis employs systems thinking skills to identify and analyze corresponding *a priori* factors within a product development firm that designs large complex systems. A case study framework provides qualitative ECM analysis from an enterprise perspective with supporting empirical stakeholder interview data. Furthermore, the research design employs more than 7,000 design defects from three large system development programs to experiment with data-mining models for classifying and predicting technical defects. This research reveals some ECM risk factors and corresponding enterprise policies in the context of process, information, and stakeholder interactions. This study also offers both executable and conceptual quantitative defect models that are appropriate for proactive risk mitigation within specific ECM processes. Ultimately, this holistic analysis provides policy recommendations for the selected enterprise, and identifies factors that have general implications for contemporary industry.

Thesis Supervisor: Olivier L. de Weck

Title: Associate Professor of Aeronautics and Astronautics and Engineering Systems

This page has been intentionally left blank

Acknowledgements

I wish to thank my fellow students, staff, and professors of the System Design and Management (SDM) Program for their support in making my program such a rewarding experience.

I would also like to thank my management at Raytheon Company, who supported my SDM studies and enabled my professional growth. And to my advisor, Dr. Olivier de Weck, who I thank for offering invaluable insight and guidance for writing my thesis.

Finally, and most importantly, I dedicate this thesis to three special people. To my wonderful wife, my success would not have been possible without your love, patience, and constant support throughout this program. To my beautiful daughter, you have brought new meaning to my life. And to my late mother, who I owe so much and so dearly miss.

This page has been intentionally left blank

Table of Contents

List of Figures

List of Tables

List of Nomenclature and Acronyms

Change: A technical design change to a formally controlled product design document.

Change Factor: The average number of changes per design document required to bring a design document into compliance with its design requirements inclusive of the full project period of performance.

Change Notification (CN): Change Notifications or Engineering Change Notice (i.e. ECN); the formal management document that Configuration Control Boards (CCB) use to initiate, adjudicate, and approve formal changes to one or more product design document.

Change Propagation: The phenomena by which technical change in one design document exhibits a causal relationship with a technical change in another design document.

Defect: Technical Design Defect that provides a reason for technical change.

NPI: New Product Introduction.

PDM System: Product Data Management IT application used to manage product configurations within a product development process.

QDM: Quantitative Defect Management.

Release: The process of placing a design document under formal configuration control; all subsequent changes require a Change Notification.

1. Introduction

The study of design iteration is of great interest in contemporary product development. While some iteration is necessary or acceptable within a new product development cycle, significant or frequent unforeseen design challenges can significantly impact project schedule and cost. These design iterations are driven by various factors that act as internal or external influences on a design team. External influences may range from contractual changes in scope from changing customer requirements to smaller redesign activities to account for the obsolescence of parts, or changing suppliers for unique components. Conversely, product development teams naturally have more control over internal influences, such as the sequencing design tasks or organization of constituent project teams to resolve cross-functional challenges that can drive design iteration. Challenging design iterations often emerge at later development stages in the form of costly technical changes that drive rework to the previously released design; this rework often propagates in the form of additional technical change to interdependent component areas that exacerbates program cost and schedule delays.

This research investigates common factors that contribute to hardware design iterations in the context of engineering change management within a modern new product development enterprise. Specifically, this research will focus on contextual social network and process interactions, which contribute to the outline problem. In concert with the identification of these factors, this research proposes concepts for data mining and the synchronization of key processes to help mitigate the impact of unintended hardware design iteration.

The bulk of hardware design work products are usually completed and 'released' within a relatively short period of time following the development. The Defense Acquisition System outlines this process as leading to Critical Design Review (CDR), which is the major program milestone that indicates when a conceived design is ready to move into fabrication, assembly, and testing stage of development (DOD Instruction 5000.2, 2008). As the primary work products of hardware engineering, hardware design documentation (e.g. drawings, models, parts lists) also cue the procurement of supplier parts and subcontracting of corresponding components. Consequently, hardware formal design reviews often experience increasing schedule pressure leading up to a release schedule.

1.1 Motivation

Innovation in complex integrated systems require both a focus on continuous improvement and the ability to manage the implementation of new process interactions and management information systems. Late stage design iterations within complex integrated system development can cause significant impact to the dimensions of program execution known as the Iron Triangle (i.e. Cost, Schedule, and Scope). To control hardware detail design and mitigate associated risk of technical change, IPTs employ various technical management processes to synchronize development activity and support decision analysis. Figure 1 illustrates the focus of this research in the context of major categories and interactions associated with defense industry system technical processes. Figure 2 illustrates a higher-level view of which technical management processes are focus areas for this research. The primary motivation for this research is the application of ECM insight for improved management of these processes and tools on future NPI projects.

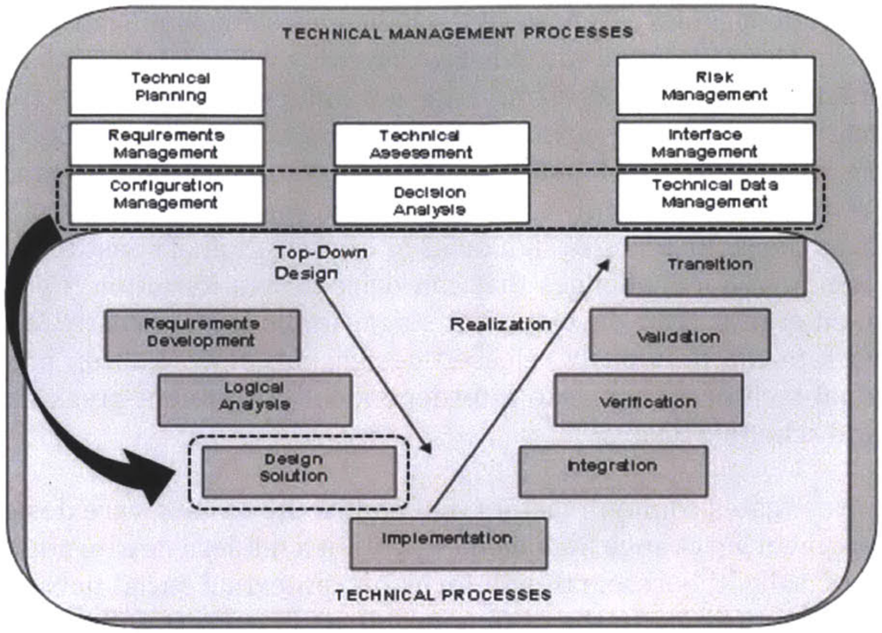


Figure 1. Adapted from 2003 Model for DoD Systems Engineering (DAU, 2003). Research Focus for Technical Management Processes within the System Engineering "V".

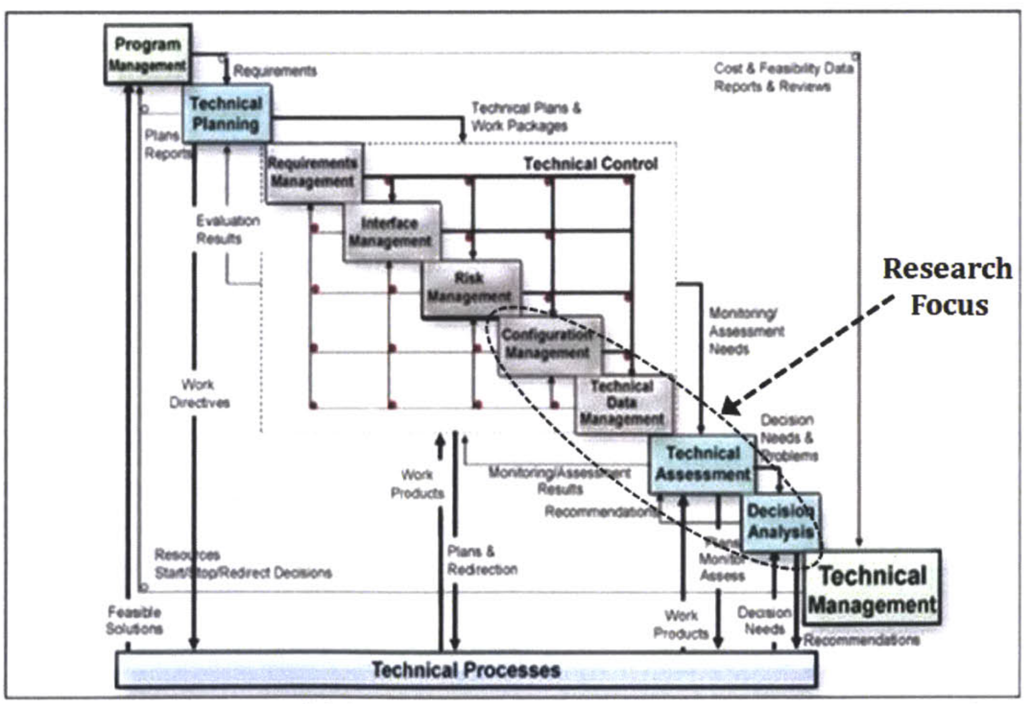


Figure 2. Adapted from 2003 Model for DoD Systems Engineering (DAU, 2003). Research Focus for Technical Management Process Interactions.

This research also relates to my work experience in configuration management, where I participated in numerous CM process initiatives and conducted ECM analysis across numerous NPI projects. After working as a Hardware Configuration Management (CM) line manager, I have detailed insight into technical management sub-processes that effect ECM, and their dynamic interactions across different stages of development.

In the book *True Change*, Janice Klein describes the Outsider-Insider phenomena in the context of organizations seeking innovative solutions that are tailored to organizational needs. Her concept states that external change agents often fail to understand the daily aspects of internal processes they are tasked to improve (Klein, 2004). Klein's research offers that internal employees can often implement change more effectively, either by acting as *Insiders* who look outside of the established business processes for new solutions, or acting as *Outsiders* to the business process while still providing unique perspectives as an internal employee (Klein, 2004).

1.2 Research Objectives and Thesis Questions

The first objective of this thesis is to identify factors that can contribute to technical change activity, with a primary focus on hardware engineering processes within a new product development enterprise. This research focused on aspects of hardware product development processes and the social network layer that contribute to the creation of corresponding design iterations on similar and complex electro-mechanical systems.

The second objective of this thesis is to develop an executable framework for mitigating identified endogenous factors by taking proactive steps that can assist decision-makers at the outset, or during the execution of, a product development program. While this research includes system and software specific defect data, the primary focus is on the hardware engineering processes within highly integrated product development programs.

Thesis Questions

Two primary research questions correspond to the hypothesis, and were investigated through the analysis of an industry product development enterprise and selected program technical change data. Several secondary questions were outlined by an iterative research design, which will be addressed in subsequent chapters. The following primary research questions support the hypothesis:

- *What common qualitative factors contribute to unintended hardware technical change activity in new product development enterprises?*

- *Can these qualitative insights be integrated with data mining models to develop leading tactical measures for helping to mitigate hardware technical change?*

1.3 Hypothesis

This research proposes that an emergent behavior may exist at the enterprise level with regard to engineering change management, which is not identifiable within any single process area. From a quantitative perspective, this research also proposes that current defect containment data can be employed in data mining models to:

- *Process defect data more efficiently* by quickly identifying relationships in historical defect data for engineering design teams and defect analysts, including the creation of visualizations.

- *Predict and target defects* with a predictive defect mitigation framework, which leverages historical defect data to create predictive defect models of configuration items and corresponding leading metrics that design teams can employ in early design stages of development stages to help mitigate future technical change.

- *Employ engineer defect experience* by quickly processing and visualizing existing relationships in PDM data. This may assist engineering managers with staffing policies, cue training, and enable user attributed checklists and cueing within PDM workflows.

1.4 Overview of Remaining Chapters

Chapter 2 (Review of Literature) provides essential background information on key topics that characterize engineering change management including configuration management and related processes, domain interdependencies, and change propagation. This review also provides a baseline understanding of analytical methods employed by this research including enterprise architecture and data mining use of nonparametric models.

Chapter 3 (Research Design) explains the research theory and design, which describes the methodical approach to analyzing the ECM enterprise, develops a holistic vision, and evaluates the utility of data mining techniques.

Chapter 4 (Exploratory Case Study of a Hardware ECM Enterprise) characterizes the selected ECM enterprise in the context of dominant view element interactions and capabilities. This chapter serves as a preliminary study of the enterprise in order to identify key relationships, which will be studied in greater detail in the subsequent descriptive case study.

Chapter 5 (Descriptive Case Study of Enterprise Factors) provides analysis of stakeholder-derived factors that contribute to technical change within the enterprise, and evaluates them in the context of key view element interactions.

Chapter 6 (ECM Enterprise Holistic Vision) develops a strategic-level vision for the selected ECM enterprise given findings from the descriptive case study.

Chapter 7 (Exploration of Data Mining Models in the ECM Enterprise) follows the strategic intent of the developed holistic vision by providing a more detailed investigation into the utility of data mining techniques. This chapter demonstrates how NPI project defect data was tested with selected data mining models, and provides requirements for their real world employment.

Chapter 8 (Conclusions and Implication for Technical Management) provides a succinct summary of findings, implications for technical management, and recommendations for branches and sequels to this research.

2. Literature Review

2.1 Introduction

This thesis considers previous research related to ECM in the context of design iteration across stages of new product introduction (NPI) projects, relevant design processes, quantitative defect management (QDM) techniques, and product data management (PDM) systems. In addition, selected enterprise architecture and data mining techniques are reviewed to provide both holistic and detailed techniques for analyzing NPI projects. Ultimately, this review provides a baseline understanding of previous related research and analytical methods that will enable a thorough investigation of hardware ECM within a modern NPI firm.

2.2 Engineering Change Management

Engineering Change Management (ECM) is a holistic term that refers to the collection of engineering change processes and technical management processes that enable effective management of design changes from a product configuration baseline and through that product's lifecycle. However, a review of ECM literature indicates the dynamics of ECM are generally not well understood (Wright, 1997; Jarrett, 2005). In Wright's (1997) review of more than 15 years of ECM research, he found that most previous ECM research addressed engineering change from a manufacturing perspective, where the discipline dealt with correcting incomplete design actions during production. A notable exception was Reidelbach (1995), who provided more forward looking observations by outlining steps for assessing the impact of proposed engineering changes, and observed that firms with long development cycles were at greater risk for experiencing uncontrolled technical change than firms with very fast clock speed (Wright, 1997). This is particularly relevant for the long lead development of integrated combat systems, which entail complex system and subsystem interactions with large bills of material (BOMs) that mature over several years. Later, Jarret et al provided more detailed review of the change process within a company, and expanded the scope of research into the phenomena of change propagation.

An underlying premise of this research is that effective and efficient ECM requires not only rigid configuration control processes, but more active synchronization of key stakeholders and processes to mitigate the impacts of technical change. This relates to two unresolved research questions posed by Wright's ECM research (1997):

- What are the characteristics of activities and communication channels in an effective ECM control system?

- What are effective and efficient ECM processes, and can these be defined by a type-specific or generic basis?

2.3 Product Design Iteration: Domain Interdependencies

This section reviews the concept of design iteration in the context of methods that analyze quantitative or qualitative interdependencies within NPI projects. While the specific techniques are not employed by this thesis, they do provide fundamental explanation and visualization into the general phenomena. These techniques were also employed by referenced change propagation research, which is evaluated by later ECM enterprise analysis.

Design iterations may result from a variety of new issues that emerge throughout the system engineering process including adjustments to incomplete or misunderstood requirements, simple errors in component design and testing, obsolescence or manufacturability of parts, or complex component interactions between subsystems that produce the need for technical change. While some iteration naturally facilitates experimentation, learning, and improvement of a product, there is a tipping point where the number of rework tasks drives excessive cost and schedule (Eppinger, 2001). Consequently, project management may look to sequence larger design tasks and assess tradeoffs to reduce the risk of design iterations, while anticipating some schedule slack and budget expenditures to resolve some level of expected iteration.

One established method for understanding and planning for product iterations is the Design Structure Matrix (DSM) (Eppinger, 2012). Figure 3 shows how DSM represents dependencies (sequential, parallel or coupled) within a single domain (e.g. tasks or people). In this example, DSM enables the sequencing or repartitioning of project tasks to help manage feedback loops between tasks that can cause design iteration (Ulrich, 2012). Eppinger and Smith also leveraged DSM to develop the Work Transformation Matrix (WTM) to help optimize time and design quality through a "decoupling strategy", and later used the DSM to develop a higher level predictive model for design iterations (Smith, 1992; Smith, 2007).

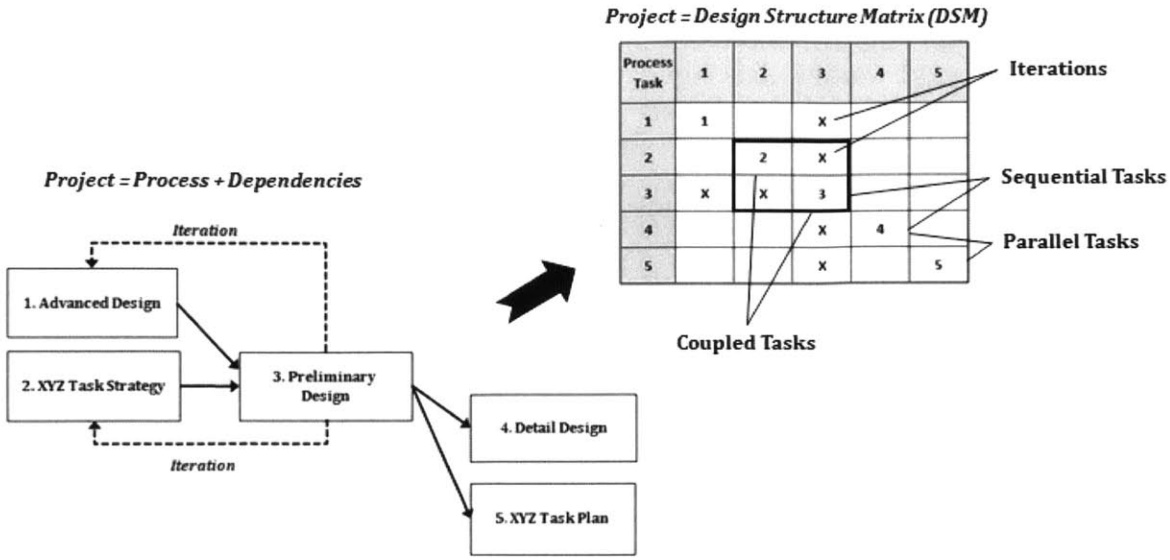


Figure 3. Notional Product Development Project translated into Design Structure Matrix (DSM).

Danilovic and Browning (2007) later developed the *Domain Mapping Matrices (DMM)* as an adaptation of a DSM to model two different domains. This method "enlightening inter-domain representations" that can identify previously unknown relationships. DMM analysis provides several benefits for project decision-makers including traceability of cross-domain constraints, synchronization of cross-domain decision-making, and integration of a new domain into an existing project (Danilovic and Browning, 2007).

More recently, Bartolomei developed the Engineering Systems Multiple-Domain Matrix (ES-MDM) to provide a conceptual framework for modeling large-scale complex systems (Bartolomei et al, 2012). By integrating the primary domains of a socio-technical system into a matrix, the ES-MSM method enables both technical and management practitioners to visually represent complex cross-domain relationships (Bartolomei et al, 2012). This method is a novel way of conceptualizing engineering systems, where the different domains of system complexity can be organized to help practitioners develop a common-mental model of an internal and external enterprise landscape. Comparison of matrices overtime can also shed light on the state of enterprise transformations.

One specific inter-domain aspect of interest is the analysis of coordination-type communication in NPI projects. Research into NPI organizations has indicated that bi-directional information transfer between upstream and downstream processes dominates instances of unidirectional information transfers in either direction (Morelli et al, 1995). This research lends credence to the importance of interdependent communication linkages, and especially information feedback from downstream back to upstream development processes (Morelli et al, 1995). Other research led to the development of the alignment matrix method for identifying misalignment between design interfaces and design team interactions (Sosa et al, 2007). Sosa, Eppinger, and Rowles (2007) established a method by which design interactions were quantified in a simple *design interface matrix*, and then compared to an independently developed *team interaction matrix* to create the *alignment matrix*. This *alignment matrix* described where an organization is exposed to risk of communication failures, and identified how system architectural changes, team organizational changes, or planned lines of communication could mitigate that risk. In addition to identifying that such misalignments were not random during industry research, Sosa, Eppinger, and Rowles (2007) found that specific types of product interface attributes complicated interactions and could be accounted for my program management.

2.4 Engineering Configuration Management

The modern configuration management (CM) discipline began in the 1950s during the development of tactical and strategic ballistic missiles. It was found that design configurations of successful prototype missiles were not adequately documented; this often led to incomplete designs that could not adequately support the mass-production of reliable missiles. Consequently, early standardization of CM practices led to the wide scale use of a military CM standard (MIL-STD-973), which was the basis for the current industry CM standard (G/EIA-649B). G/EIA-649B defines CM as a lifecycle process that "maintains consistency of a product's attributes with its requirements and product configuration information". Figure 4 illustrates the five required processes for executing effective configuration management:

- *Configuration Planning and Direction*: How CM functions are synchronized.

- *Configuration Identification:* What configuration items (CI) are under CM control. Generally, if a component is going into production, it is a CI or a component of a CI. Often, the contract drives the level to which CIs are specifically identified.

- *Configuration Status Accounting (CSA):* How configuration information is controlled. Generally, CSA functions use the PDM system as an IT solution.

- *Change Management (aka Configuration Control):* How changes to CIs are managed. Generally, the program Configuration Control Board (CCB) is responsible for managing these configuration changes.

- *Configuration Verification & Audits:* How physical product is verified in comparison to its approved design configuration.

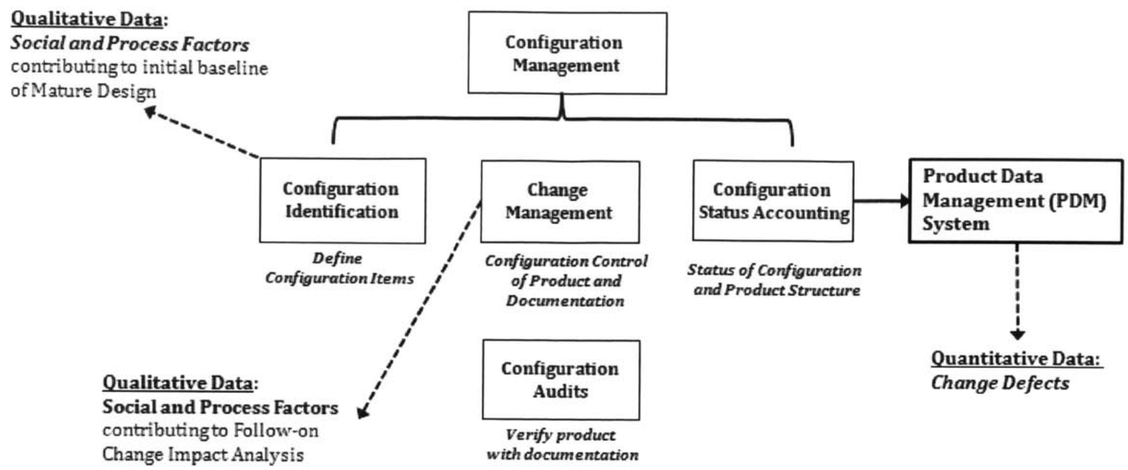


Figure 4. CM processes and interactions with this research.

In the context of modern NPI projects, CM is a key technical management process that provides program managers with adequate linkages to the engineering process and controls over the design. Specifically, CM provides formal control over the design configuration, and its constituent work products, as they develop from an initial baseline to a mature design that can enter integration, testing, and manufacturing stages (Wright, 1997). Every component, subsystem, or higher-level system is documented to describe the 'form, fit, and function' of its design, and how they are aggregated to the highest level of the product. The CM *configuration control* process then facilitates the control of design changes on change notifications (CN), which record why and how a design document is being changed from its baseline configuration (i.e. the first version put under CM control). CNs are controlled by at least one product data management (PDM) system, which is the primary IT tool for tracking how the structure of design documents aggregate to represent components of higher level systems. Tracking how design configurations (i.e. the structure of documents) dynamically change can be particularly complex. A useful analogy is thinking of the baseline design configuration (i.e. the structure of design documents) as the "root" of a tree, after which design changes spawn divergent branches that end at the most current design descriptions, otherwise referred to as "branch tips" (Krishnamurthy, 1995).

Throughout the performance of an NPI project, there are also identifiable clusters of design change activity as reported by Eckert; smaller *ripples* and larger *blossoms* refer to the varying magnitudes of change activity (Eckert et al (2004). Often these patterns emerge from specific events or stages of the Integrated Product Development Process (IPDP). Ripples may occur following the initial release of a configuration baseline, after which design defects are formally resolved and corrected as changes. Also, specific events like the integrated testing of a particular subsystem or obsolescence of key component may drive more significant change activity. Figure 5 illustrates change patterns and evolving configuration baselines in comparison to the common IPDP.

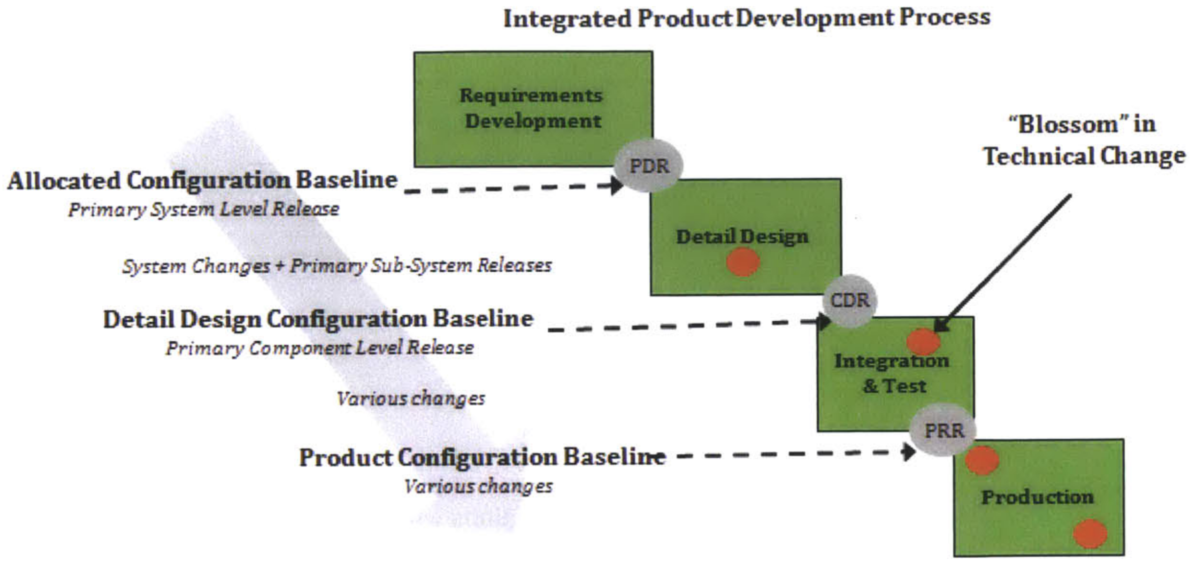


Figure 5. Change Patterns and Configuration Baselines in context of an IPDS.

2.5 Quantitative Defect Management (QDM)

One form of design change analysis is called Quantitative Defect Management (QDM). In the context of design iteration, QDM employs analysis of *defects* in development stages after the stage when their parent design document was originally released. This common industry method aligns processes and tools to capture categorical and continuous defect attributes from approved CNs to enable later statistical analysis and lessons learned.

QDM processes employ existing configuration control processes to explain the documented "reason for change" on each CN in terms of subordinate defects for each design document being changed. This research uses the term 'defect' to broadly describe an error that is causing its parent design document to not satisfy a design requirement. While the term may be construed as describing only a design mistake, in the context of this research, defects include all unintentional design errors (i.e. mistakes) and intentional development actions that only later are discovered to be deviating from defined system requirements (which may themselves be incomplete). Therefore, the term *defect* captures all mistakes that a designer should have known about, and the development actions that were unforeseen. Unique classes of attributes then describe each defect via temporal, qualitative, and quantitative measures. These attributes also indicate the dynamics of when the defect was identified with respect to specific system waterfall stages. Figure 6 illustrates this basic process.

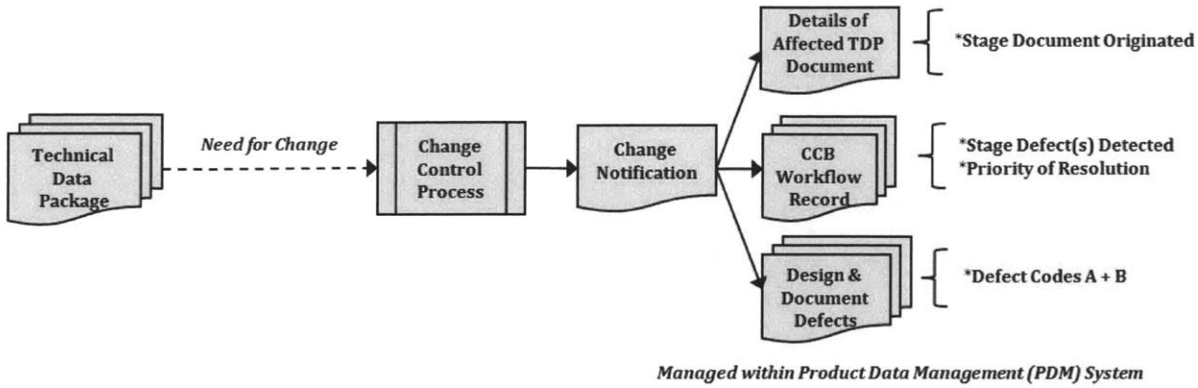


Figure 6. Basic Configuration Control Process employing PDM System and Quantitative Defects Identification.

QDM analysis is usually framed in the context of higher-level stages in the system engineering waterfall process. Defects that are identified later than the stage of original document release are considered *uncontained defects*, which are more strongly correlated to costly rework tasks later in a project.

Figure 7 illustrates a defect containment matrix for a notional software program, which segregates the document origination stage from the stage in which a defect was detected; defects identified below the gray diagonal indicate uncontained defects (Campo, 2007). For example, in the upper left corner of the matrix, we see that a large number of Advanced Design document defects (i.e. 950) were detected in the Preliminary Design Stage, which followed the Requirements Stage where those documents were originally released. Defect containment matrices are useful for *posteriori* tracking of program performance, which may cue project managers to conduct more detailed root cause analysis within those stages take corrective actions.

Stage Detected	Stage Originated						
	Advanced Design	reliminary Desig	Detail Design	IV&V	Manufacturing	Operations	Total
Advanced Design	1000						1000
Preliminary Design	950	1350					2300
Detail Design	725	1250	1850				3825
IV&V	550	720	1050	80			2400
Manufacturing	220	250	320	45	50		885
Operations	0	5	6	5	5	10	31
Total	3445	3575	3226	130	55	10	10441

Figure 7. Notional Defect Containment Matrix within IPDS.

2.6 Change Propagation: The Phenomena of Systemic Technical Change

Assessing the full impact of a proposed design change is the primary challenge for all stakeholders within an ECM enterprise. *Change Propagation (CP)* is a condition where one design change drives further need for change in the technical design of interconnected components (i.e. physical or functional) or the administrative attributes of design documents (i.e. nomenclature, UID codes, part numbers, etc). Consequently, CP is a result of change impact not being fully understood before implementation. Interestingly, military and industry CM standards do not discuss this phenomena, or the processes and data management techniques that are necessary to identify and track such conditions. However, research has observed and analyzed the phenomena across different industries and NPI projects. Jarret el al (2005, p. 9) proposed four reasons for why change propagation can occur:

- "oversight" of known system component connectivity.
- "lack of systems knowledge" with regard to component interconnectivity with other systems components.
- "communication failure" across interdependent design activities on the same component.
- "emergent properties of complex systems" due to non-linear interactions like mechanical vibration.

Research conducted by Monica Giffin (2007) on *Change Propagation in Large Technical Systems*, and by Michael Pasqual (2010) on a *Multilayer Network Modeling of Change Propagation for Engineering Change Management*. Giffin (2007) and Pasqual (2011) provided innovative and invaluable insight via the empirical analysis of relationships between design change documents (similar to CNs) at the endogenous change-network and social-network layers within a single program. Giffin (2007) relied on *parent-child relationships*, where a change document exhibited a causal relationship with subsequent change document; and *sibling relationships*, where two change documents, where linked to prior parent document, as the fundamental CP network connections.

Giffin's (2007) innovative use of DSM to model change propagation corroborated previous research by Jarrett et al (2005), which indicated that few design changes actually led to follow-on change propagation. Interestingly, Giffin's network CP analysis also confirmed previous research by Eckert et al (2004) by identifying that some components exhibited identifiable propagation behavior with how they interacted with other system components; individual subsystem behavior was then expressed with one of three metrics including Change Acceptance Index (CAI), Change Reflection Index (CRI), or Change Propagation Index (CPI). For example, some components acted as "multipliers" of change, by propagating changes to multiple other components within the system (Eckert et al., 2004).

Pasqual later leveraged Giffin's data to analyze the social-network implications of CP and develop a holistic *Multilayer Network Model for Change Propagation* across the social-network layer, change-network layer, and product layer (2011). His study included the development of metrics that identified how individual engineers contributed to overall technical change within the system.

2.7 Enterprise Architecture (EA)

The growing complexity of socio-technical challenges in modern business have led decision-makers to consider more holistic views and systems thinking (Nightingale, 2004). Legacy views of companies as machines have been replaced by the view of companies as organisms, which are more interconnected within larger business ecosystems (Nisbet, 2009). This paradigm recognizes that enterprises exhibit unique higher-level system properties, "soft properties", and behaviors that emerge from the interactions between enterprise elements and stakeholders (Nightingale, 2004). *Enterprise Architecting (EA)* is one such strategic approach for using systems architecting principles to analyze business enterprises as complex systems. EA employs management science and engineering systems principles to provide a suite of visualization and analytical techniques that characterize enterprise landscapes in the context of eight enterprise views elements, enterprise stakeholders, and the business ecosystem that enterprise operates within (Nightingale, 2012). With an understanding of the existing enterprise, decision-makers can leverage the approach to understand potential misalignments and generate transformational concepts and plans for future (Nightingale & Rhodes, 2012). Figure 8 illustrates the ten EA view elements that are used to frame the internal and external influences of an enterprise. Figure 9 provides a notional enterprise example where key element interactions function to achieve a market imperative.

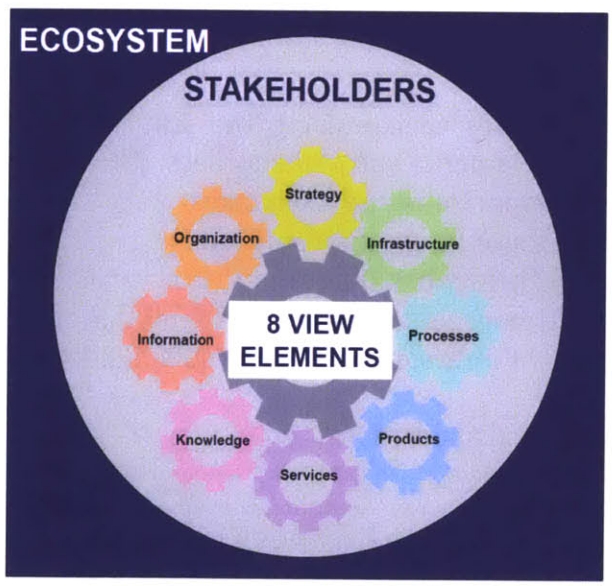


Figure 8. Adapted from MIT ESD38 Lecture 1 (Nightingale & Rhodes, 2012). Ten Enterprise Architecting (EA) Elements: Characterizing the Internal Landscape and External Ecosystem.

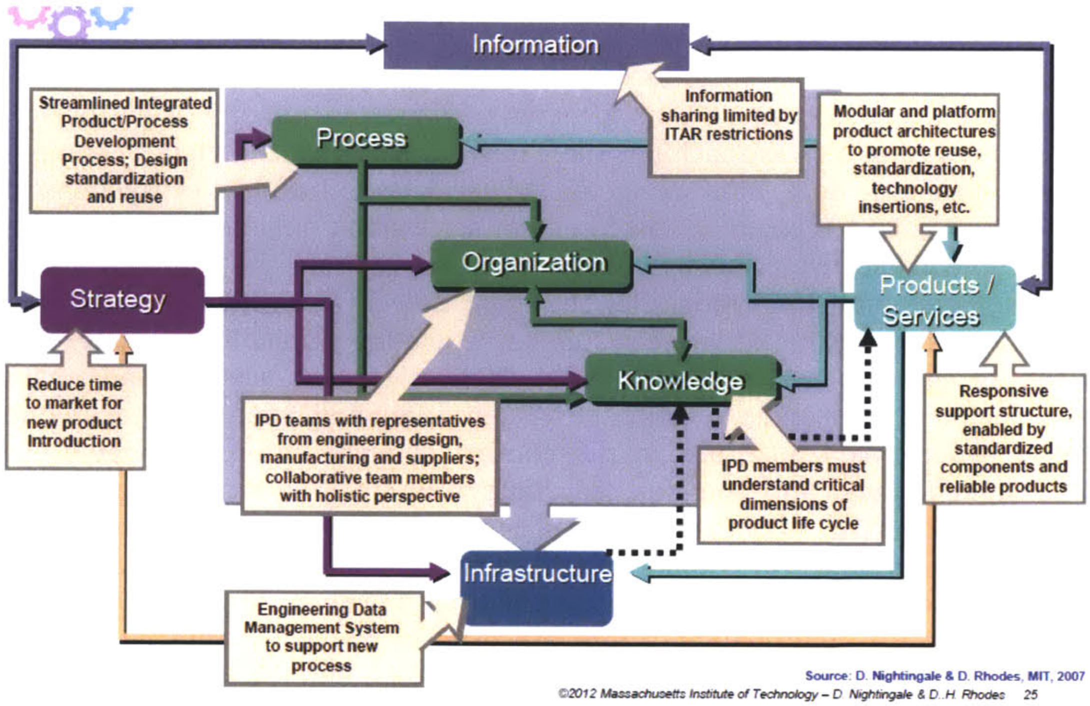


Figure 9. Ten Enterprise Architecting (EA) Elements: Example of Reducing Time to Market Imperative (Nightingale & Rhodes, 2012).

EA employs different techniques to capturing the current state of the enterprise and checking alignment between elements and stakeholders. These may include:

- Stakeholder Value Comparisons and Exchanges
- Stakeholder Salience Venn Diagrams
- Element or Stakeholder Network Diagrams
- Interviews to identify emergent enterprise capabilities and "soft properties"
- X-matrix comparison of strategic objectives, stakeholder values, key processes, and metrics
- Statistical graphics and visualization

2.8 The Process of Data Mining for Business Intelligence

In the current age of 'Big Data', more business are measuring and recording parametric data from key business processes. As more processes and interactions are recorded in databases, there is substantial opportunity for mining those large and varied datasets for business intelligence. One method for leveraging key relationships from the wealth of information in these data sets is the use of models. We know from basic psychology that even the most disciplined professionals can exhibit inconsistent and opinion-based decision-making, or are subject to emotional bias or fatigue. However, with the use of appropriate data and human expertise, data driven models can quickly and consistently identify key relationships that can provide a competitive edge over conventional unassisted decision-making (Bertimas et al, 2012).

This research recognizes that sophisticated PDM systems have enabled engineering firms to aggregate 'data warehouses' with parameterized engineering data from NPI projects. Data mining is a technique that can be used to explore data and build predictive models to support a decision-making task. The process leads to the identification of patterns and relationships between relevant variables, and then models those features in a way than be generalized in comparison to new data. Fundamentally, data mining seeks to accomplish the following (Nisbet, p.34):

- Improving the understanding of data by revealing previously unknown relationships.

- Developing models that predict or forecast events, which support alternative decision-making.

Classification versus Prediction

Classification is the process of grouping data into categories of variables (Nisbet, p.235). Prediction is the process of identifying variables within the input data that are good predictors of a specific target variable (Nisbet, p.50). The difference between the two can be subtle; for instance, if a decision tree is built with a sample dataset to find groupings of data (i.e. classifiers), then this is a case of classification. However, if that same model is then tested against a new dataset to find unknown groupings within that new data, then that model is being used for prediction. However, some texts use the classification nomenclature when using categorical data, and prediction when using numerical data (Shmueli, p.13).

Data Mining Heuristics

When considering models, it is important to consider common heuristics that govern good modeling practices and identify boundaries (i.e. confidence levels) for models and their deployed applications. Just as a book is sometimes judged by its cover, novice model users sometimes misinterpret models as representing truth or believe in the best model; but the value of models should instead be measured by what indicators or features they can accurately represent (Nisbet, 2009). This cautionary heuristic is important for both modelers and subsequent consumers of model analysis, who should understand that application boundaries of a model are just as important as where a model succeeds to convey useful relationships (Nisbet, 2009). This is particularly important when we consider that models (especially non-linear ones) perform poorly when applied outside the bounds of known data from which they were built (Nisbet, p. 774).

Another heuristic is explained by Occam's Razor, which in the context of models, explains that when two methods produce similar results, the simpler of the two is best. This simple rule prescribes two important lessons. The first is best explained by the famous statistician, Leo Breiman, who expanded on Occam's concept by comparing model simplicity versus accuracy (Briemen, 2001). Breiman describes that simplicity provides for interpretability of model function, accuracy, and limitations; but often this interpretability comes at a cost to predictive accuracy (Breimen, 2001). Secondly, Occam's Razor also relates to the generalization of a model. The value of a model is derived from its ability to provide accurate results (i.e. generalize) when applied to various new sources of data. In data mining, each additional explanatory variable used to build the model will also introduce a new dimension for the model to emulate. Consequently, if a model is overfit to the unique features of the dataset it was built upon, then that model is likely to perform poorly (and not generalize) when analyzing new data in the future. This trade-off between the inclusion of variables and model generalization is referred to as the 'curse of dimensionality' (Schmueli, p.145).

Another important heuristic deals with predictive models and understanding when input could be "accepting leaks from the future" (Nisbet, p.741). Modern product data management systems hold tremendous amounts of technical and managerial data, but often do not easily parse what relationships were known at the time (Nisbet, p.743). Consequently, it is not uncommon for retrospective analysis of data to lead to building a model that can only work with knowledge of the future. Conversely, project failures or those that are prematurely cancelled naturally result in lack of documentation, since businesses want to cease unprofitable or then unfunded activities as fast as possible. This is not uncommon in NPI projects, and can introduce "survivor bias" into historical studies of current NPI projects. This heuristic raises the importance of data preparation for understanding key relationships or holes in historical data, which may require consultation with subject matter experts.

Initial Data Selection and Exploration

This process starts with a clear understanding of the question(s) that is driving the data mining effort. In the context of this research, data mining analysis was used to understand the patterns of uncontained defects in similar NPI programs, and determine if that data can lead the creation of basic and useful explanatory or predictive models for engineering processes. This process started with the aforementioned initial selection of variables that were available and related to technical defects, as well as initial processing to support subsequent data exploration. The key tools that enable data exploration are descriptions and visualizations, which may include descriptive statistics, correlations, and various forms of graphical data plots. The focus of this activity is to understand the multi-dimensional sample space that is represented by a multivariate data set, and identifying high-order features that are unique to that sample space (Elder, p.745).

Data Preparation and Reduction

Data exploration supports the identification of interesting data features to use in a model, while at the same time identifying which characteristics are the most important relationships. Understanding these opportunities within the data enables data preparation, variable selection and dimension reduction, which all support more accurate and deployable models. For example, if a model used two or more variables that described the same features within a data set, the modeling algorithm may over-fit the outcome to that specific feature. This example of variable redundancy should be avoided, because it reduces model effectiveness and generalization to new data sets that may not exhibit those same features.

When preparing data, it is important to use only those variables that are absolutely necessary to characterize the important relationships. In this way, Occam's Razor articulates a fundamental heuristic of simplicity when architecting models. But when evaluating and designing solutions to complex problems, it can be a challenge to determine in which dimensions simplicity is most valuable. In the context of data mining, Occam's heuristic relates to the principle of parsimony, which generally explains that simplicity is a desired characteristic of models because the N independent variables governs an N-dimensional solution space for a model. Consequently, more effective models reduce the number of input variables to the fewest number that are necessary to explain the variation and features within the data set.

Building, Evaluating, and Deploying Models

The process then considers various techniques to build and validate models, followed by the selection of the best model based on evaluated model performance. Finally, the selected model can be deployed to support the decision-making.

Multi-Layer Perceptron (MLP) Neural Nets

Neural nets were named after the human neuron because they were perceived to learn through algorithmically across various nodes in a similar fashion to interactions between neurons. Modern neural nets use an aggregation processes that sums input variables combined with an activation process that employs a linear or logistic function to reach an output node (Nisbet, p.129). As seen in Figure 10, these functions forward propagate through an architecture of nodes to achieve a desired final target node. One common way of improving the performance of neural nets is employing back-propagation, which changes the weighting of a misclassified outputs and iterates to improve the performance of only those functions that had performed poorly (Nisbet, p.131). Consequently, neural nets can be very flexible and improve the fit of the model to the data; this form of neural net is called a *feed forward neural net with back-propagatio*n (Nisbet, p.131).

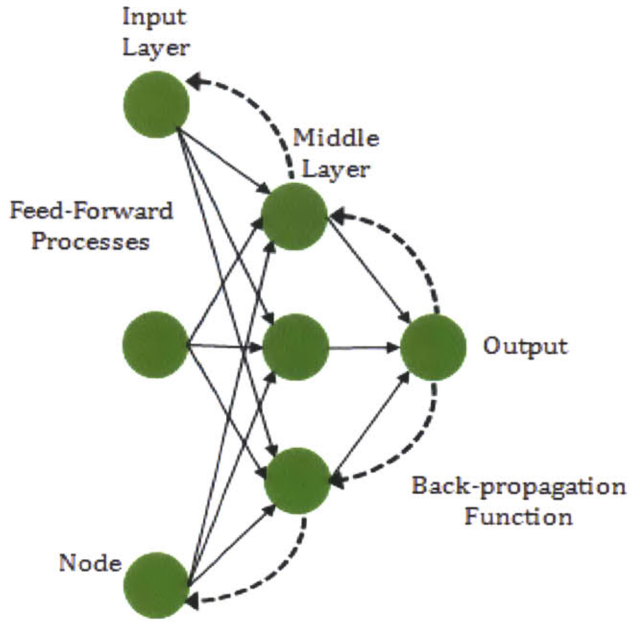


Figure 10. Example of a feed-forward with back-propagation neural net architecture.

Classification & Regression Trees (CART)

Classification & Regression Trees is a fundamental modeling method that produces easily interpreted visualizations of the model construction; they are similar in appearance and function to common binary decision-trees. The algorithm uses recursive partitioning to separate an *N*-dimensional space of the input data into "non-overlapping multidimensional rectangles"; recursive means that one partition (i.e. decision-node) is dependent on previous partitions and continues to produce branches until a terminal 'leaf' is reached (Schmueli, 166).

After the algorithm develops the full tree, it is important to prune away unnecessary tree branches to prevent over-fitting of the tree model to the training data; this is another way to preserve the ability of the model to generalize to new data. Pruning is done by selecting tree decision nodes and cutting away successive terminal 'leafs' to essentially make a decision-node into a terminal 'leaf node (Schmueli, 180). The algorithm then determines measures the effect of the pruning on model accuracy to determine if that branch is necessary to retain. This process is done successively to find the most efficient CART model.

Multivariate Adaptive Regression Splines (MARS)

Multivariate Adaptive Regression Splines are similar to regressions trees, where the algorithm uses smooth basis functions (instead of branches) to fit the model to regions of the input data. The method is also useful for feature selection in the data because it selected basis functions according to their contribution to an accurate output (Nisbet, 82). In this research it was used for classification with multiple response variables.

Ensemble Modeling and Boosted Trees

One of the best modeling techniques is to use groups of models to overcome the shortcomings any one particular model. Known as *ensemble modeling*, this technique requires that models be somehow joined to produce a composite response that is generally better than any one of the constituent models (Nisbet, p.304), provided these model are generally accurate and provide some varied behavior. Figure 18 illustrates how increasing the number of component models within an ensemble can increase model performance.

There are several ensemble methods that can improve the performance of the models, or assist in the estimating of their accuracy. *Boosting* is a method that runs multiple testing iterations on a model, where successive tests are focused (i.e. weighted) on poor performing variables during initial model training. A very successful and flexible implementation of this method is the *Boosted Tree* (Nisbet, p.249). Another method associated with CART models is called *Random Forests,* where many classification trees are run nearly simultaneously with pruning, and then the model with the most reoccurring successful output (i.e. majority wins) are selected for use (Nisbet, p.248). Finally, cross-validation sampling of validation can improve model performance by first dividing the testing data into several parts, followed by each part being independently used to the test model against the remaining parts (Nisbet, p.307). This research employed *Boosted Trees* as the preferred ensemble method.

Figure 11 illustrates ensemble formulas for a notional ensemble of a support vector machine, multi-variate adaptive regression spline, and a neural network.

$$1.\ \text{Ensemble Example} = \frac{1}{3}(\text{CART} + \text{MARS} + \text{NN})$$

$$2.\ \text{Ensemble Variance} = \frac{(\text{CART - Ensemble})^2 + (\text{MARS - Ensemble})^2 + (\text{NN - Ensemble})^2}{3}$$

Given:
CART = Classification and Regressions Tree
MARS = Individual Response of MARSplines Model
NN = Individual Response of Neural Network Model
Ensemble = Individual Response of the Ensemble of Component Models

Figure 11. Equations for (1) Linear Average of Response Ensemble and (2) Ensemble Variance.

2.9 Linking Literature to Research Design

This research has provided a baseline understanding of engineering change management and related concepts, processes, and techniques that characterize contemporary NPI project environments. In addition, this review described fundamental analytical methods that will enable both holistic enterprise analysis and detailed modeling of design defects. In the following chapters, we will review the research design that will methodically employ this knowledge to forward ECM research and propose both enterprise specific and general measures for controlling technical change.

3. Research Design

3.1 Introduction

The literature review provided some insight into previous research, and common standards, in addition to significant discussion into the general subject of engineering change management during new development of complex technical systems. This research design seeks to analyze the efficacy of the proposed hypothetical framework by characterizing the salient elements of the selected enterprise, collecting relevant qualitative and quantitative data, and applying analytical techniques to support the evaluation.

3.2 Theory and Research Design

This research process starts with the definition of a theory, which provides the lens through which the research design is developed. This case study framework will attempt to show important systematic factors at the process and social levels of new product development that complicate the management of hardware technical change; this research will also show alignment opportunities between engineering change management practices with PDM systems that can help mitigate identified factors. This underlying research theory identifies important research focus and boundaries for both planning and allocation of research resources. In addition, this theory provides some measure of generalization of conclusions to other real-world examples, which is ultimately the purpose of pragmatic research rather than focusing on unique outlying cases (Yin, 1993, p.6).

3.3 A Framework using Case Study Method

The first role of research design is to define a sufficient plan for collecting appropriate evidence, which can be analyzed to resolve corresponding research questions. The second role of the research design is to define strategies that provide fidelity to resulting conclusions, and address rival theories that could provide alternative explanations (Yin, 1993, p.45). This research design considered the theory from holistic perspective, which considered that quantitative data from specific programs cannot be appropriately analyzed or applied without first understanding the socio-technical aspects of the selected enterprise. This assumption led to the design of an iterative framework, which integrates a series of two single-case studies to answer different sets of questions, which will provide relevant evidence to the analysis of the proposed hypothesis and thesis questions. The first exploratory set of questions sought to identify relevant program roles and technical change processes within the selected enterprise. The exploratory set of questions was then used to select and adjust a second set of questions, which informed the execution of stakeholder interviews. The case study method was chosen to support the investigation of a contemporary phenomenon in the context of modern design practices, and to logically support resulting theoretical discussion (Yin, 2003).

Figure 12 illustrates how the research design employs qualitative stakeholder interview data and quantitative program technical data to logically support thesis development, answer the defined research objective, and test the efficacy of the hypothetical proposition (de Neufville and Field, 2010).

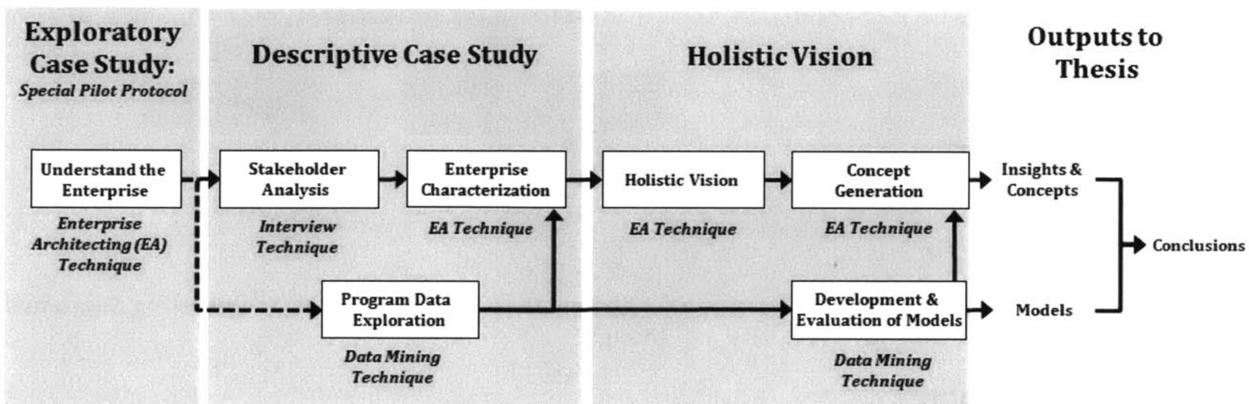


Figure 12. Research Design Diagram: Process, Methods, and Techniques.

Figure 13 illustrates the research design in the context of an enterprise architecting sequencing model (Nightingale, 2012). Consistent with the aforementioned theory, this research investigates the enterprise, develops a holistic vision from identified opportunities, and generates supporting concepts. Full development and validation of these concepts, and a full transformational plan is outside of the scope of this research.

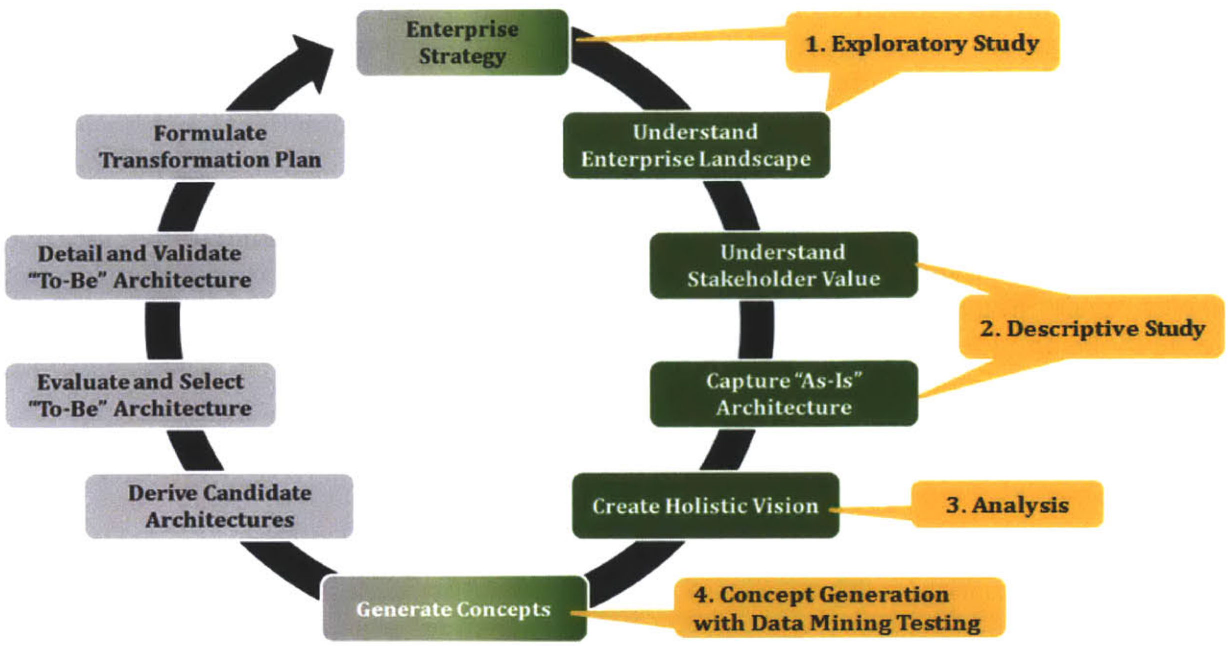


Figure 13. Adapted from MIT ESD38 Lecture 5 (Nightingale, 2012). Enterprise Architecting Sequence Model.

Exploratory Case Study

The first case study was an *exploratory case study,* which was used to understand relevant elements of the selected enterprise in the context of the research theory; these enterprise elements are the focus of this case, and therefore function as the *unit of analysis.* The exploratory case uses what Yin refers to as *special pilot protocol* to resolve uncertainty about how the research should proceed towards the subsequent 'real case study'(Yin, 1993, p.6); this study seeks to identify which specific enterprise aspects are most likely to have a causal impact on hardware technical change. Consequently, the primary *unit of analysis* for the exploratory case are the relevant enterprise elements, attributes, and qualities. The exploratory study functions only as a prelude to the 'real case study', which must gather new data for follow-on analysis.

Supporting Enterprise Architecture Analysis

Enterprise Architecting (EA) was the primary analytical technique for framing and evaluating the ECM enterprise. This technique is useful because it provides holistic analysis of the enterprise, while also enabling one to zoom into specific elements and their attributes. EA techniques also help to identify potential discontinuities between strategy and underlying relationships within the enterprise, which provide contextual understanding of the "as-is" architecture. These more subtle but significant process and organizational relationships are particularly important when identifying the boundaries of ECM stakeholder selection. The ECM enterprise may vary significantly between different firms because the underlying responsibility areas cross several functions and involve both technical and managerial oversight. Consequently, EA analysis not only served to guide this exploratory study, but also identified boundary conditions that support or refute generalization of findings.

Descriptive Case Study

Following the resolution of exploratory questions, a second set of relevant questions were modified to frame a *descriptive case study* that gathered new information. Yin's research indicates this is a critical step to preventing bias via "slippage from the exploratory stage into the actual case study"(Yin, 1993, p. 6). The primary *unit of analysis* for the descriptive case are the representative stakeholders, who can share their perspectives on social and process factors that either mitigate or exacerbate hardware technical change in the context of previously studied enterprise elements.

The descriptive case study also informed the selection of comparable new product development programs, which represent complex integrated system development within the current enterprise. In the context of this research, the key attributes of a design change are the individual technical defects that correspond to a particular engineering design document. Each of these technical design defects have their own attributes, which describe how the defect relates to the associated design document, the cause of the defect, and the time dimension of when the defect was originated and later identified. This research isolates these defect attributes as a secondary *unit of analysis*, which includes isolation and statistical analysis of common trends for the purposes of building data mining classification or prediction models.

Supporting Enterprise Architecting and Data Mining Analysis

Following the exploratory case study and the identification of relevant factors from the stakeholder analysis, enterprise architecting analysis was used to holistically evaluate stakeholder empirical data and characterize the "as-is" enterprise. This characterization enables analysis of important interactions and supports the identification of opportunities at the social-network and change-network layers within the ECM enterprise. To support the analysis of enterprise view elements, this study also employed data mining techniques to explore the dynamics of associated technical defect activity in NPI projects.

Developing a Holistic Vision for the ECM Enterprise

This research design employs the opportunity identified during the descriptive case study to generate a holistic vision for the future hardware ECM enterprise. This vision provided context for subsequent recommendations that addressed stakeholders roles and incentives, sub-process interactions and policy friction.

Exploration of Data Mining Models in the ECM Enterprise

Data mining models were explored to test the underlying theory that such techniques are of value to the specific enterprise and the general employment of ECM. The data mining process was then used to analyze program defect attributes with a focus on supervised classification techniques. This lead to the careful selection of comparable programs, which were representative of common enterprise NPI projects. Even within the same enterprise, different programs may experience unique contractual requirements and pressures that impact detail design change, particularly at the social network layer (i.e. social network relationships with suppliers and customers interactions) and the product layer (i.e. unique design and documentation requirements). Consequently, the comparison of 'apples with apples' required similarity in the context of common project constraints projects. This was achievable by selecting programs with the following dimensions of comparison:

Product Focus on new products under the similar dimensions of commodity type, program schedule, and complexity.

- *NPI through First Article Build:* New product development projects may employ varying amount of design reuse from other contracts. The selected programs were new product development from system design to the production and delivery of the first article, with similar amounts of design reuse within the sensor components.

- *Commodity Type:* While all selected programs were employed with different concepts of operations (CONOPS), they technologically were similar sensor systems in both scale and design.

- *Complexity:* All selected programs had comparably sized technical data packages and numbers of integrated subsystems.

Enterprise Elements Focus on programs that operate within a similar enterprise landscape and under similar constraints.

- *CM Constraints:* The project CM plan will determine how CCB and engineering resources are dynamically applied to control technical documentation in different phases of development. One CM Plan does not fit all programs, and is both customer and product specific. The selected programs had similar CM processes and planning.

- *Team Organization:* A smaller and internal IPT organizations that employs agile techniques may have better leverage to control technical change than larger project that is reliant on coordination with major subcontractors. The selected programs had similar size program teams without major subcontractors.

- *Customer Requirements:* Significant differences in customer requirements, design or testing standards and directly impact CM scope on different contracts. The selected programs were working for a similar customers in the defense industry, who employed the similar contractual standards.

Introduction to Product Development Program Data

Three datasets were analyzed from similar development programs of large sensor systems, which included complete development from system design to product baseline of the first article. Each program was described by between 3000-7000 technical defects that were subordinate to higher level approved *Change Notifications.* Defects represented system, software, and hardware component defects that were captured over 6-7 years development periods of performance. Additionally, there was an associated originator for each technical defect, since each technical defect corresponds to a parent approved CN. Each program defect dataset included more than 100 originating engineers, which enabled the analysis of the experience distribution via initial data exploration. From this distribution, three engineers were chosen for more detailed analysis based on the statistically significant number of defect records they authored (i.e. $n>10m$; where n= # *of records, and* m = # of explanatory variables).

Initial Selection of Explanatory Variables

At the outset, the selection of variables was limited to basic defect attributes and relationships to program development stages. The following predictors were selected from readily available continuous and categorical attributes that were associated with each defect. Data was retrieved from data base reports for each program, with individual defects being described by a small but relevant set of independent variables. Later chapters will discuss the methods used to further reduce these variables for model building.

- *Date of Change (Time Series):* This variable represents the date that the associated CN was formally approved, and was necessary to support unstructured learning and data mining of the defect profile. Because we are not interested in modeling the specific defect instances, the time variable is not included in the predictive defect model or engineer classification models. These dates have significant null time periods between them.

- *Defect Code A (Categorical: Nominal Discrete Qualitative):* This variable represents the general engineering function of the Design Document, which is exhibiting the given defect. Defect Code A can be thought of as a general category for how the document is being used. This Defect Code A is somewhat correlated to the actual document type (e.g. assembly drawing, system spec, software spec, printed wiring board), which was not analyzed in this project.

- *Defect Code B (Categorical: Nominal Discrete Qualitative):* This variable represents the specific error that the document is exhibiting. While not specifically tied to a document type (e.g. Parametric Model, Printed Wiring Board, Specification, etc.), Defect Code B is strongly correlated to the actual document type. Document Type was not analyzed in this project.

- *Stage (or Stage) that Document was Originated (Categorical: Discrete Ordinal):* This variable defines the stage in which the associated TDP Document is first released. These stages align with common defense industry stages for integrated product development. These include Advanced Design, Preliminary Design, Detailed Design; Fabrication, Assembly and Test (FAT), or Operations.

- *Stage (or Stage) When Defect is Detected (Categorical: Discrete Ordinal):* Using the same values as *Stage that Document was Originated*, this variable defines the stage in which the defect was formally identified and adjudicated by the program CCB. Consequently, the *Stage Detected* will always follow the *Stage Originated*.

- *Stage Characteristic of Defect (Categorical: Discrete Ordinal):* This variable indicates if a defect was *Contained (C)* within the stage that the described TDP Document was released. If the defect was identified in a later stage, then the defect was considered *Uncontained (U)*.

- *Priority of Defect (Categorical: Discrete Ordinal):* This variable rated the defect priorities #1-6, where #1 Priority Defects have the highest priority. In this analysis, priority is broadly defined as the significance of resolving the given defect in light of conformance to product requirements, and program cost and schedule.

- *Originator:* This variable represented the individual engineer, who authored and forwarded the design document defect. This engineer may or may not have also been an engineering appointee to the program CCB.

Data Capture and Anonymization

Technical defect data was mined from the complete compilation of selected program change notifications, which were derived from two Product Data Management (PDM) and business intelligence tools. In this particular enterprise, different tools were used to tailor the product development to the needs of specific engineering disciplines. One database contributed to the formal configuration control of system requirements and software items, and a second database contributed to the formal configuration control of hardware components. Specific technical defects were drawn from all approved project CNs.

Program datasets were then reviewed to remove invariant or sensitive data. Sensitive data fields that were appropriate for analysis were replaced with categorical symbology; this preserved the linkage between the individual program, engineer, defect category and the individual defects records without compromising sensitive data. These included:

- Names of individuals

- Names of specific company Integrated Product Development Systems (IPDS) stages

- Names of companies

- Names of programs and subordinate projects

- Names of defect categories
- Dates of defect detection

Data Preparation and Exploration

The selection of appropriate dataset and explanatory variables requires administrative preparation to promote successful data mining techniques. Once attributable information was isolated and removed, the following observations and actions were taken:

- There were no outliers or missing values in the data, which could have skewed the results of specific types of models.
- Transformations were required. Categorical text variables were then transformed to numbers to facilitate statistical analysis and machine learning. For instance, spreadsheet columns with *Categorical Nominal Discrete Qualitative* variables (i.e. Categories 1, 2, 3 with non-sequential meaning) or *Categorical Discrete Ordinal variables (i.e. Categories X, Y, Z* with sequential meaning) were transformed into separate columns with binary variables values.
- Insights were provided by descriptive statistics, including the absence of linear relationships.
- Feature selection enabled variable reduction to essential variables.

Data Mining Model Building Process

Following the selection and preparation of data, the data-mining modeling process was followed. This entailed selection of the appropriate algorithms, model building and experimentation, model evaluation and selection of the most effective model. Proposed models were not operationally deployed in this research.

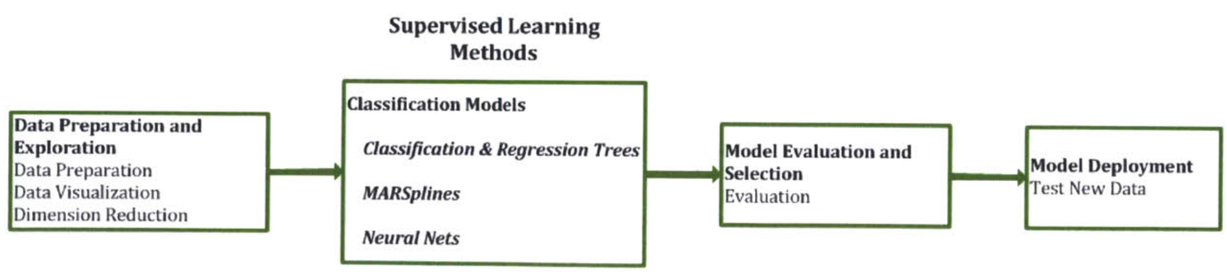


Figure 14. Data Mining Process Flow.

3.4 Logic

The qualitative case study method is universally recognized as a sufficient method based on Yin's research, which indicates that investigators can use case studies to "define topics broadly and not narrowly, cover contextual conditions and not just the phenomena of study, and rely on multiple and not singular sources of evidence" (Yin, 1993). This research integrates a case study method with both qualitative and quantitative techniques to appropriately collect and analyze empirical data in the context of the outlined research theory. Known as *logical positivism*, this practice methodically integrates these activities to test the hypothesis and derive findings and conclusions (Yin, 1993).

Logical Positivism is achieved through a variety of techniques. First, this research partitions different studies by research stages, where corresponding strategic-level and operational-level questions are independently answered through sequential case-studies. Second, a multi-disciplinary approach supplements these qualitative techniques with statistical "analysis of archival (quantitative) records" at a tactical level (Yin, 1993). This approach is supported by Yin's research, which indicates the case study method is effective either alone, or sometimes in combination with other analytical methods, to support the collection of data and logical analysis.

The generalization of research findings to other real-world cases is perhaps the most critical challenge encountered during case study research. While recommendations are tailored to a specific enterprise, this research assessed stakeholder responses for qualitative outliers, and discussed alternative rival theories for observed phenomena (Yin, 1993). Consequently, this design supports *analytical generalization* of identified qualitative factors, and *statistical generalization* for successfully illustrated defect models given similar standardized ECM and QDM processes. In addition, thesis recommendations relate analysis to existing systems engineering metrics, systems architecture heuristics, and previous research findings that relate to Product Data Management (PDM) systems.

3.5 Research Limitations

There are several limitations in this research design, which must be acknowledged for potential impact to findings. First, the single-case study method assumes replication logic to explain that relevant phenomena drawn from the investigation is representative of other similar enterprise cases (Yin, 2003). However, this assumption is reasonable considering the single case represents a sufficiently large and stable matrix enterprise, which replicates standardized processes in a stable enterprise landscape across multiple independent development programs. Understanding similarity in program social networks, strategic managerial policies, and enterprise culture, this research design can logically isolate significant factors from stakeholders who can have a common mental model of the enterprise and that factors that affect programs technical change activity. While statistical sampling was used on three different NPI programs for testing the hypothesis, that method was not intended to identify contributing factors to hardware technical change or highlight opportunities for process improvement. Instead, statistical modeling was used to support hypothesis testing, by trying to build actionable leading indicators. If this research design had required statistical sampling to identify such factors, then a multi-case survey would have been the preferred method to acquire the necessary data (Yin, 2003).

Second, the complete removal of bias is not possible in this research. My experience in the field of hardware change management and interests in complex systems was likely to have some effect on both my perspective on the chosen enterprise and the selection of stakeholder interviews questions. However, purposeful thought was applied to the research design and implementation in order to mitigate bias to elements of this case study and resulting analysis.

Third, many attributes of change notifications and supporting defects were not considered in this research; these various other attributes were left out due to the focus of this research, and difficulty in acquiring that data for several recent and similar programs. Like most modern engineering development enterprises, the selected firm promotes a culture of continuous improvement of processes and synchronizing of these processes to support data collection. In the last few years, the enterprise has established several new enterprise design excellence initiatives and a new enterprise PDM system, which could not contribute sufficient data due to earlier start dates of the selected enterprise programs. In addition, due to process limitations during the execution of the selected programs, hardware engineering change notifications lacked the necessary relationships to support aforementioned change propagation analysis.

Examples of Data Limitations

There are three specific examples of missing data that could have contributed to this research. The first is the causal relationship between defect origination and part re-use on other programs. Consider that a notional program (i.e. Program #2) was using a TDP design document that was originated (and therefore owned) by a previous notional program (i.e. Program #1). If Program #2 wanted to change that re-used document, then the Program #2 CCB would have to coordinate and fund that change through the Program #1 CCB. While the selected enterprise attempts to leverage commonality between similar products, this type of data was excluded for several reasons. First, these types of changes were not considered to be insignificant to overall technical change profiles of selected programs. And second, the acquisition and processing of data associated with part re-use conditions was resource prohibitive. Another example of data that was missing was the cost of correcting defects. This data was not included due to limited resources for data processing, the proprietary nature of such costs, and differing perspectives on how to define these costs. Finally, the commodity type (at a subsystem level) that related to the particular defect was not easily related to defect records.

3.6 Summary

This research design organizes a holistic approach with detailed methods to characterize and analyze ECM as an identifiable enterprise within a selected firm. Specifically, the exploratory case study first characterizes the ECM enterprise, which enables the adaptation of supporting questions and direction of stakeholder interviews within a subsequent descriptive case study. Following enterprise analysis and the development of holistic vision then provides prospective utility for the investigation of more detailed data mining techniques. Ultimately, this research seeks to identify the complex interactions between enterprise elements, and sets the stage for understanding if data mining techniques are both actionable within existing processes and supportive of ECM knowledge management.

4. Exploratory Case Study of a Hardware ECM Enterprise

4.1 Introduction

This exploratory case study characterized the hardware engineering change management (ECM) enterprise and explores which elements and interdependencies are most relevant to hardware technical change activity. This led to a more detailed understanding of potential policy interactions and key stakeholders, which informed the following descriptive enterprise study.

4.2 Exploratory Case Questions

The following questions represent the uncertainty associated with this preliminary study:

- *What is the Hardware ECM enterprise, and which enterprise elements and capabilities are most important to managing technical change?*
- *What key interactions is Hardware ECM most heavily dependent upon?*
- *Which stakeholders have the most leverage over key interdependencies in hardware technical change?*

4.3 External Ecosystem: A Defense Contracting Company

The ECM enterprise operates within a larger defense contracting firm, which is characterized by a "process-driven engineering enterprise" paradigm, with clusters of integrated product teams (IPTs) that are the immediate customer-like organization units that benefit from ECM. These IPTs and Program Managers are conceptually viewed as *Heavyweight Project Matrix Organizations,* where Program Managers have greater comparative control over budget and resource allocation than supporting functional managers as first presented by Hayes et al (1988). While the *Heavy-Weight* vs. *Lightweight* metaphor for matrix organizations only illustrates two extremes, company matrix programs are closer to a middle-weight project matrix organization because functional engineering managers are very influential on program design decisions, management of engineering budgets and EACs, and allocation of engineering manpower.

Project financial management and manufacturing processes are the primary external interfaces between the external ecosystem and the ECM enterprise. To a lesser extent, customer needs may be influenced by the maintenance of fielded products (i.e. as-maintained configurations). However, the key process interdependencies include:

- Earned Value Management System (EVMS) Processes, including engineering Estimates-at-Completion (EACs) and project tracking books.

- Manufacturing Data Package (MDP) document and processes.

- In addition, engineering process initiatives act as infrequent and evolutionary influences upon the ECM enterprise. Key examples of these influences include alignments between:

- Alignment between interdependent IT systems, including Material Resource Planning (MRP) systems and PDM capabilities.

- Incremental updates to the firms Integrated Product Development System (IPDS), which acts as a product development process architecture.

Company Vision and Strategy

The company has exhibited a consistent vision to be an industry leader in the development of innovative defense systems and related technology. The strategy to achieve this vision focuses on innovative solutions and technology in the areas of communication and networking, military sensors, and weapons that span both kinetic and non-kinetic domains. This strategy supports an established engineering-focused culture that values innovation, with a focus on the quality of design solutions and system performance. Consequently, technical change is considered a fundamental task in providing innovative solutions, both on NPI projects and upgrades to previously deployed systems. Although in some regard, ECM is considered a necessary evil within the engineering specific culture. Some experienced engineers (even chief engineers) feel that ECM invokes onerous process requirements and administrative costs at the detriment to resources that could be better applied to innovation. This reflects the cultural dichotomy of ECM and supporting configuration control processes, because the same engineers would not argue the value of consistent traceability of configurations throughout the systems engineering waterfall (i.e. achieving system verification & validation), which is critical to compliance with modern quality management standards and customer value exchange.

Contemporary Configuration Challenges in Defense System Development

The company also operates within a larger defense contracting ecosystem. In the context of influences on the ECM enterprise, the current state of fiscal austerity and regulatory changes have the most influence. Changes made to the DOD's primary acquisition policy (DoD Instruction 5000.02) in December 2008 secured a significant strategic shift in defense acquisition policy that transformed the defense acquisition ecosystem. As a result, major U.S. Defense contractors are being increasingly decoupled from government-funded technology development and managed under closer scrutiny to drive execution of design and development (i.e. post Milestone B) of new NPI programs. With an emphasis on schedule and cost predictability, various measures were enacted by DOD Instruction 5000.2 sought to reduce requirements creep, moderate existing requirements, and improve integrated system testing and evaluation. This policy shift in the defense acquisition ecosystem has created a more competitive environment as customers increase execution standards in light of looming budgetary constraints. Other efficiency efforts included transitioning from legacy sole-source government contracts to competitive proposal practices, while channeling technology development and advanced design contracts to smaller Federally Funded Research and Development Centers (FFRDC) and academic R&D laboratories.

Within the system development process, technical change of hardware design components and subsystems are a major contributor to the number of overall system changes, particularly within highly integrated and complex electro-mechanical products. Recognizing this, external stakeholders including higher-level customer CCBs, government contract auditors, subcontracted partners, and suppliers of key components have some influence on the technical or management aspects of hardware change activity.

4.4 Hardware ECM Enterprise Landscape

This research is characterized by an identifiable "process-driven engineering enterprise" paradigm that operates as a cross-functional entity within a large company. In contrast to the more *Middleweight Project Matrix Organization* at the program level, Hardware ECM emulates the a *Lightweight Project Matrix Organizational* view because the primary view elements are engineering focused (Hayes et al, 1988). While the CCB Chairmen (usually the Program Manager) are the approving authority for released baselines and technical changes, they are more focused on scheduling final reviews of CNs and facilitating higher-level coordination of contractual configuration requirements. Consequently, the knowledge and discretion of key engineering stakeholders provides the greatest leverage within the enterprise.

The danger of no "Burning Platform"

However, ECM is generally not considered a critical capability by itself, but rather an amalgamation of minor processes that provide experiential value to their users. For this reason the engineering culture does not recognize ECM as a "burning platform" that requires improvement. While the enterprise has completed the replacement of IT-based PDM systems, the resulting productivity improvements have had little effect on the underlying effectiveness of ECM sub-processes, which have remained stagnant for more than a decade. Recognizing this lack of urgency is critical to both interview strategy in the later descriptive study stage of this research, and identifying avenues for relaying research findings and recommendations.

4.5 Hardware ECM Enterprise: Dominant View Elements

In the context of EA view elements, hardware ECM is dominated by configuration and design sub-processes, information management via PDM systems, and more abstract knowledge management practices. This section characterizes these dominant view elements, and a brief review of other relevant interactions.

4.5.1 Processes

The enterprise is characterized by specific sub-processes that are subordinate to one or more functional engineering disciplines including configuration management, materials & process engineering, and hardware engineering (mechanical, electrical, and parts engineering functions) processes. The sub-processes where there is the greatest leverage to mitigate the risk of change are:

Detail Design Peer Reviews: The process mandates that subsystems and components must undergo at least one design peer review, which uses various checklists and design tools to ensure compliance to higher level form, fit, and functional requirements and manufacturability.

- *Key Stakeholders:* Cross-functional design teams, process engineering.
- *Structure:* IPT process (i.e. cross-functional engineering), text-based, micro-focus to a design sub-system or component.
- *Behavior:* Standard process, but developmental (i.e. iterative) reviews of designs are not mandated. High repeatability and degree of adherence, but little control over quality or level of defect containment attained.
- *Periodicity:* Prior to Released Baselines.
- *Key Artifacts:* Checklists, Peer Review Record, Action Items, Previous program defect analysis.
- *Measures:* Wrought defect identified, but no measure of defect impact.

Configuration Control: The process mandates that any released design documents can only be changed via a CCB-approved Change Notification (CN), which outlines the reason for change, solution for technical change, and impact of that change to the 'Iron Triangle'.

- *Key Stakeholders:* Configuration Control Board including the program manager and representatives from CM, engineering, and quality.
- *Structure:* IPT process (i.e. cross-functional engineering), text-based, generally micro-focus to a change.

- *Behavior:* Standard process. High repeatability and degree of adherence, but little control on the level or documentation of impact analysis.

- *Periodicity:* After Released Baselines.

- *Key Artifacts:* CN, Configuration Reports, Checklists, Workflow Record.

- *Measures:* Change Factor, Change Impact Cost.

Defect Analysis: The process mandates that defect identified from previous closed projects are analyzed to inform trends, root cause analysis, and lesson learned for implementation on future design efforts. The goal is continuous improvement of defect containment percentages consistent with lean principles.

- *Key Stakeholders:* Process engineering, with design engineering participation.

- *Structure:* IPT process (i.e. cross-functional engineering), test-based with some graphical, generally macro-focus of a project.

- *Behavior:* Standard process. High repeatability and degree of adherence.

- *Periodicity:* Generally after Program Closeout, but analysis is then applied to future design reviews.

- *Key Artifacts:* Technical defect databases (captured from CNs), statistical analysis, and defect reports.

- *Measures:* Project defect containment percentage as measures design maturity.

4.5.2 Information

From a more IT centered view, Weill defined enterprise architecture as "The organizing logic for key business processes and IT capabilities reflecting the integration and standardization requirements of the firm's operating model" (Weill, 2007). This characterization is appropriate for framing the ECM information view element because the primary informational enabler is an advanced PDM system, which standardizes the process and attributes by which NPI project configuration information is stored, controlled, and developed over time.

Fundamentally, the PDM system provides standardized CM functions across all projects within the business unit. This includes configuration control of all design baselines and associated Bill of Materials (BOMs), and provides user-defined reporting functions based on hierarchal "parent-child" or "where-used" relationships within projects. The PDM system also provides a workflow function, which enables CCBs to manage the development of CNs and other technical management directives. However, the enterprise cannot easily identify systematic relationships between change actions without previous knowledge of why change action occurred. This means that change propagation identification, analysis and associated metrics (e.g. CPI, CRI, CAI, etc) are not possible in the current enterprise. Interestingly, the tool is able to establish these types of relational attributes, but the process does not mandate or encourage such action. The Information view element anatomy consists of:

- *Key Stakeholders:* Enterprise IT, CM Coordinators as tactical-managers, all functions as consumers.

- *Structure:* Internally-managed within individual projects. Hierarchal product structures.

- *Behavior:* High degree of openness via a standard process. High repeatability and adherence to process. Alphanumeric "atomic" data, with few visualizations. User access controlled, but generally open to entire user base.

- *Periodicity:* High frequency and driven by product data needs.

- *Key Artifacts:* PDM System Configuration Reports.

- *Technical Data Package (TDP) Reports:* These are PDM output reports of product structures under a top-level part number. In the context of Krishnamurthy's "root" and "branch" analogies (1995), these reports show the complete listing of documents and associations that comprise a particular "branch" of a product's configuration.

- *Document Status Reports:* These are PDM output reports of a particular TDP document configuration history by a specific configuration item.

- *Custom Reports:* CM can create custom spreadsheets to answer unique user requirements. Some requests are implemented as on-demand reports that can be pulled directly from the PDM system.

4.5.3 Knowledge

The ECM enterprise is culturally associated with CM discipline, which seeks to embed baseline ECM knowledge into standard processes that govern access and control to product configuration data (i.e. Information View Element). Consequently, ECM knowledge accrued over time only resides with individual stakeholder experiences. While engineers populate a lessons-learned database with unique design, testing, and manufacturing lessons; there are few if any ECM-specific lessons. The extent to which an engineer can leverage knowledge of change impact is largely experiential, along with knowing how to massage PDM data to mine the relationships that enable the identification of technical change impact. There are no identifiable incentive measures for documenting or sharing systematic ECM-Knowledge. The knowledge view element anatomy consists of:

- *Key Stakeholders:* CM Coordinators, Design Engineering, Process Engineering.

- *Structure:* Internally-derived and product-based knowledge. Functional engineering communities are more experiential than technocratic. Formal lessons are documented in an enterprise database but are product specific design lessons.

- *Behavior:* Moderate degree of openness to knowledge sharing, generally via a standard process.

- High repeatability and moderate of adherence. Few direct incentives for capturing new knowledge.

- *Periodicity:* Performed at Program Closeout.

- *Key Artifacts:* The enterprise lessons-learned database is the primary knowledge asset and repository. ECM-specific knowledge is a function of tacit lessons imbedded in product specific lessons-learned.

- *Measures:* No identifiable ECM-specific knowledge measures. Unlike other product/technology areas, there are no ECM incentive-based measures such as patents or recognition of white papers.

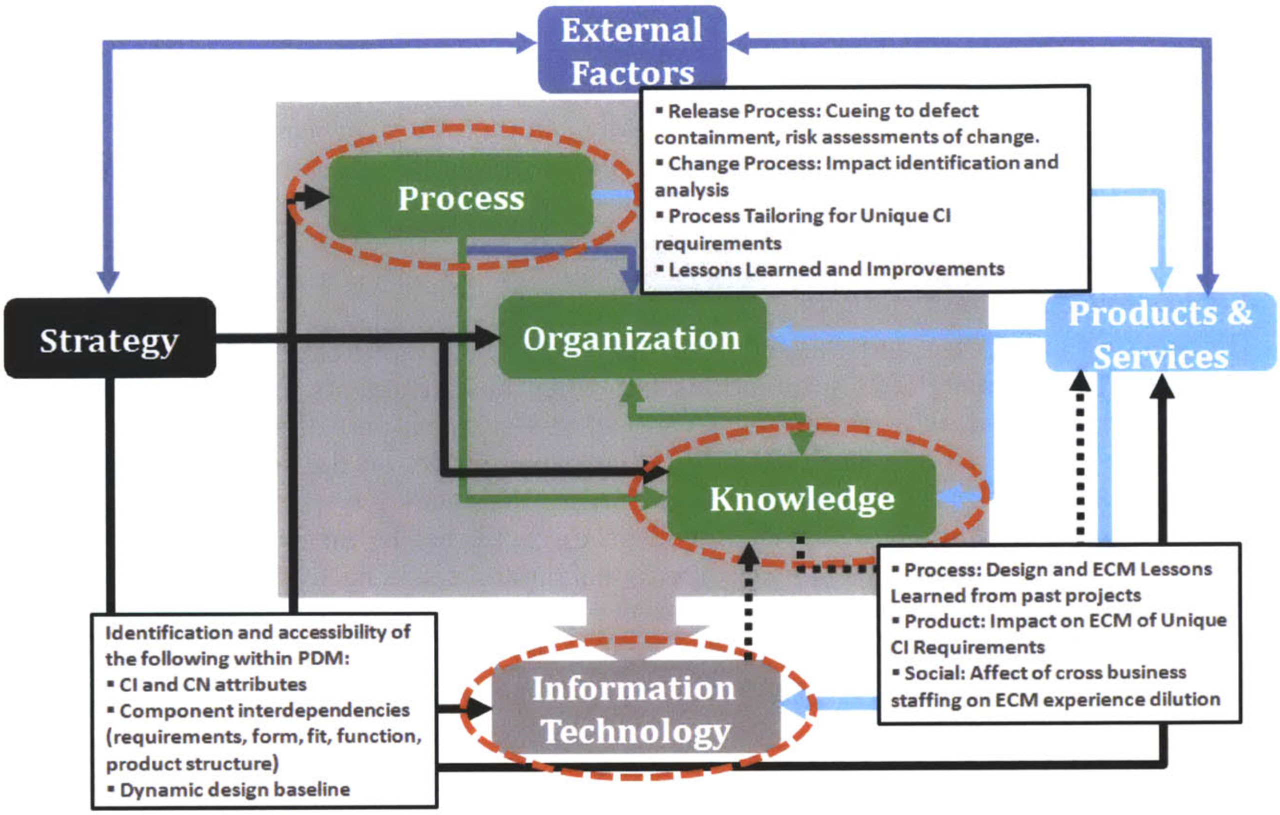


Figure 15. Adapted from MIT ESD.38 Lecture 2 (Nightingale, 2012). Ten ECM Enterprise Elements

4.6 Other View Elements

4.6.1 Organization View Element

While not considered a dominant view element, the distributed organization of stakeholders is an important consideration. At first glance, the CCB seems to be the formal unifying entity for the discussion and resolution of any ECM issues. However, change impact analysis is often more dependent on functional engineering analysis that is already completed prior to the change notification being forwarded to the CCB formal review. Consequently, the primary organization boundaries that drive ECM are defined by the functional departments processes including:

- Configuration Management: Primary stakeholders for configuration control.
- Materials & Process Engineering: Primary stakeholders for Defect Analysis and Design Process.
- Mechanical Engineering and Design: Primary stakeholders for Design Release and Change Impact.

4.6.2 Strategy View Element

While there is no one articulated ECM strategy, the company IPDP and supporting enablers are strategically aligned to achieve mature technical baselines during the design and development stage. Also, as an operational imperative, Process Engineering and Engineering Design organizations have prioritized defect containment analysis across all programs.

4.6.3 Defense System Products

The existing enterprise and the scope of this research is more closely aligned with NPI projects. The primary ECM interactions are when design defects are found during the *Integration, Verification & Validation (IV&V)* of sub-systems and during *Low-Rate Initial Production (LRIP).* To a lesser extent, ECM has some impact on the servicing of deployed systems (i.e. as-maintained configurations). Also, the recursive nature of upgrades and refurbishments with many of the deployed products cannot be discounted. For instance, most complex sensor and weapons systems encounter some hardware design upgrades throughout their lifecycle. While most are minor, some products undergo significant electro-mechanical upgrades as old configurations are periodically sent back to the company to undergo nearly complete refurbishments to newer design configurations.

4.7 Enterprise Stakeholders

Figure 16 illustrates the primary internal and external stakeholders within the ECM enterprise. These stakeholders exhibit the greatest leverage over identification of early risk of technical change or later during the analysis of specific design change impact. Internal stakeholder groupings reflect alignment with the three hardware engineering functions and corresponding departments.

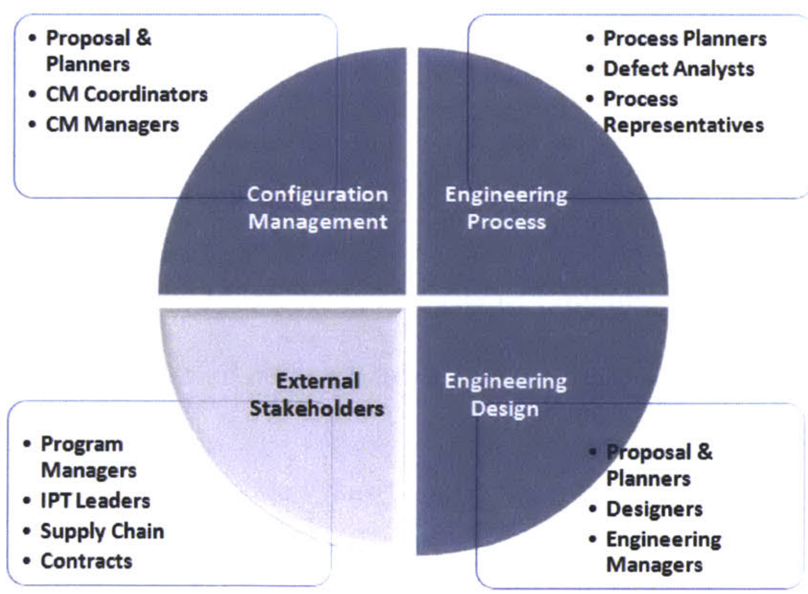


Figure 16. Basic ECM Enterprise Stakeholder Categories.

4.8 Enterprise Capabilities

The external landscape (i.e. IPTs and Business Units) is very robust, by aligning functions to deal with varying development challenges of unique designs. This external landscape is an industry leader in understanding unique subsystem interdependencies with complex electro-mechanical sub-systems, identifying problem states, running root cause analysis, and solving complex "tactical" design challenges across a broad variety of commodity types (e.g. various RADAR exciter-transceivers, SONAR arrays, electro-mechanical assemblies, C3I systems).

However, this engineering robustness does not necessarily translate to tactical measures for understanding of systematic design change action. Consequently, the lower level ECM enterprise exhibits sustainability of configuration control processes as the primary enterprise capability. Interestingly, the CM process is considered sufficient because it is compliant with contract and industry standards, which have no substantive requirement to understand propagated change. Consequently, "process improvement" initiatives focus on near-term manpower productivity and standardized IT solutions, rather than establishing an awareness and controls on propagated change.

4.9 Summary

The exploratory case provided a basic definition of the ECM enterprise and highlighted key interactions between view elements. Considering the unique enterprise capabilities, cultural norms, behaviors and attributes of dominant sub-processes; the knowledge of key stakeholders is the primary leverage point for effective ECM. Consequently, engineering participants to those sub-processes and CM coordinators can provide the broadest perspective contributing hard and soft factors for effective ECM.

5. Descriptive Case Study of Enterprise Factors Contributing to Hardware Technical Change

5.1 Introduction

The descriptive case study leverages stakeholder interviews to gather qualitative empirical data that can identify factors that contribute to hardware technical change. Specific interview questions were selected for each stakeholder based on their interaction with specific view elements, and their ability to comment on relevant social interactions and design release processes. Stakeholder empirical data was then collated in the context of the enterprise elements and key processes, and analyzed for common concepts that were seen as contributing factors to technical change.

5.2 Descriptive Case Questions

The following research questions outline the purpose of the descriptive case study:

- *Do engineering stakeholders observe common qualitative factors which contribute to uncontained defects within engineering change management process or social interactions, which contribute to uncontained defects?*
- *Where could a predictive defect model provide the greatest benefit to line operations and program management during new product development?*
- *What aspects of existing process or social interactions detract from the utility of the concept?*

5.3 Profile of Stakeholders Interviews

Interviews were conducted with experienced ECM stakeholders from program technical staffs, hardware functional engineering and the CM discipline at various levels to provide insight into significant factors to technical change activity. Table 1 provides stakeholder profiles to provide context to aforementioned key interactions. Subsequent interview discussions highlighted soft factors and interactions that would be difficult to identify with questionnaires or surveys; deeper discussion also enabled iterative questioning to dig deeper into substantive issues.

# of Stakeholder Interviews	Function	Perspective	Experience Level	Focus of Interview Questions
1	IPT HW Lead	Manager	15+ years	*Social and Change Network Interactions with ECM *Change Propagation
1	HW Engineering	Manager	10+ years	*Social and Change Network Interactions with ECM *Change Propagation
1	HW Engineering Process	Technical	10+ years	*Social and Change Network Interactions with ECM *Defect Containment, Change Propagation *Design Review Process
1	HW Engineering	Technical	10+ years	*Social and Change Network Interactions with ECM *Defect Containment, Change Propagation *Design Review Process
1	HW Engineering	Technical	5+ years	*Social and Change Network Interactions with ECM *Defect Containment, Change Propagation
2	HW Configuration Management	Technical	15+ years	*Social and Change Network Interactions with ECM *Change Propagation
1	HW & SW Configuration Management	Technical	15+ years	*Social and Change Network Interactions with ECM *Change Propagation *Difference between SW and HW CM
1	HW Configuration Management	Manager	15+ years	*Social and Change Network Interactions with ECM *Change Propagation
9 Stakeholders	Cross-Functional: 4x CM, 4x Engineering, 1x IPT	Different Perspectives: 3x Managers + 6x Technical	Distribution of ECM Experience	Tailored to Stakeholder Process Interactions

Table 1. ECM Enterprise Stakeholder Interview Profile.

5.4 Intersection of Information and Organization Elements

Exogenous Supplier and Customer Interactions

There can be significant differences in how internal and external stakeholders define, collect, and record instances of technical defects. Another example is use of modern computer aided design tools across the social network layer. Simply using CAD tools with defined technical requirements does not preclude the occurrence of interference between two components. A subcontracted assembly may be designed and fabricated with different software, configuration management processes, or quality control standards that can introduce misunderstanding when reviewed or inspected by the prime contractor. Previous defense industry research used surveys to report that prime contractors were more likely than suppliers to integrate product data across the product lifecycle, and use more capable PDM systems that functioned with interdependent applications and workflows (Hines, 2005). Consequently, a system integrator who subcontracts major portions of design, fabrication, assembly, and test with a new subcontractor may risk additional technical change that otherwise wouldn't have occurred if internally sourced. One interviewed engineer had worked for several years with iterative design and testing of a complex electro-mechanical assembly that was being developed by a subcontractor, who conducted configuration control with less rigorous process controls; the resulting configuration issues led to significant effort spent on problem resolution and resulting CNs. In a similar discussion, an another experienced hardware engineering IPT lead also indicated that customers and partnering firms must have a common understanding of both contractual design limitations and configuration management requirements. Discussions indicate that misalignment in design capability, rigidly controlled system interfaces, and configuration reporting capabilities can contribute to additional design and administrative technical change.

Separate interviews with a more senior engineer and manager explained that significant design change activity usually follows the release of major design baselines as previously referenced. Both interviewees mentioned that customer incentive awards to meet contractual milestones associated with program events (e.g. CDR, TTR, PDR) may actually encourage programs to release design baselines with known errors. This may happen because the resulting technical change costs are thought to be less than the net gain from incentives and customer satisfaction. This research was unable to locate processes for assessing ROI in these situations; there appeared to be no discrete measures to assess the systemic impacts to cost and schedule.

Endogenous Interactions between Integrated Product Teams (IPT)

Interviews indicated that engineer inexperience or lack of alignment with unique program requirements can also introduce design defects. Unique design requirements often correspond to a commodity type, or contractual documentation format of a specific customer configuration management system. One manager indicated that some project managers feel they are not getting "the right people" for the technical design challenges they are facing with complex development tasks. Some experience dilution that had caused unforeseen technical changes could be traced to the introduction of available engineers, who had comparatively less experience with a unique commodity type. The discussion is interesting because the engineering firm has a strategic imperative to leverage the relatively new enterprise PDM system to drive more agile work share and cross-company collaboration.

Analysis of Experience Dilution

This phenomena may indicate that experience dilution can be a driving factor for technical change with unique documentation requirements. Brooks' Law explains that introducing relatively inexperienced engineers late in a development project can cause experience dilution, which reduces the effective work accomplished on projects (i.e. introduces rework and lowers productivity), and is particularly problematic if there is already a backlog of rework tasks (i.e. unresolved design defects). Appendix A illustrates Brook's Law with a causal loop diagram for a notional NPI project; the corresponding system dynamics model illustrates the notional project would achieve a marginal reduction in project schedule (finishing 2 weeks early), but at a significant cost of required manpower in order to pay for the new staff and resolve rework tasks (Knight, 2011). While interview findings were not in the context of adding staff during the later stages of projects, the issue of experience dilution would introduce a similar effect to productivity and rework (i.e. technical change) as a result of incomplete change impact analysis.

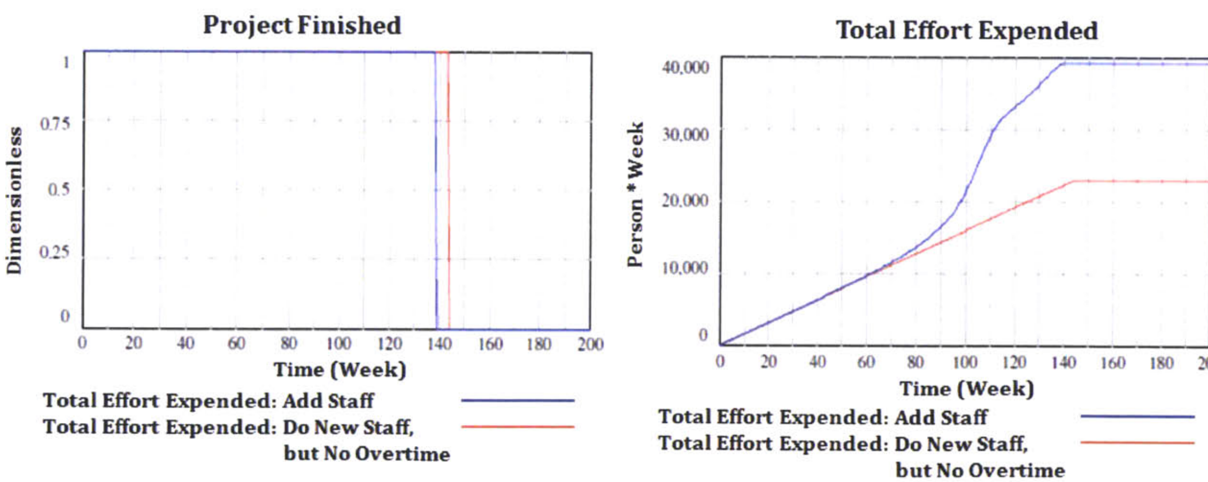


Figure 17. Net Effects of Adding New and Inexperienced Staff (Knight, 2011).

There is some disagreement to the severity and applicability of Brooks' Law with modern development practices. Steve McConnell explains that "controlled projects are less susceptible to Brooks' Law than chaotic projects" (1999). He offers that better project tracking enables managers to better gauge when and how to add staff. Also, modern documentation and design processes "make tasks more partitionable", which enables new staff members to more quickly close the relative gap in experience (McConnell, 1999). McConnell concludes that mature design firms generally know when additional staff would be counterproductive. However, enterprise analysis and general discussion across several interviews indicates that ECM experience is not only based on processes, but also product specific knowledge and information interactions that may reduce the rate of learning.

5.5 Process, Knowledge, and Information View Elements: Change Propagation

Discussions with CM Coordinators, Design Engineers, and Senior CM Managers indicated the existing enterprise does not collect identifiable empirical evidence on propagation relationships. None of the CM personnel were very familiar with the CP phenomena, and out of three design engineer interviews, only one engineer indicated that he or she had regular experience with identifying propagated change between subsystems. While considered less common, technical CP was more problematic to foresee and correct once identified. CP was related to either some level of misunderstanding between two technical perspectives, which may involve remote work-share, cross-functional, or cross-IPT scenarios.

Also, all interviewees acknowledged cases of administrative propagation, where improperly implemented CNs required a second CN to correct the original design intent. Conceptually, this rework is similar to technical change propagation, but was not viewed as a significant problem with design work. There are two causes of this CN rework:

- "Hanging" CNs: When successive CNs are approved for one design document, but not implemented into the design documents themselves, the CN's are said to be "Hanging CNs" against their parent design document. While this is a common industry practice, the interpretability of the current design diminishes as numerous Hanging CNs aggregate. Hanging CNs are used to save on drafting costs under the assumption that implementing multiple changes at once is more cost effective than individual CN implementation into design documentation. The interpretability problem also varies depending on the complexity and number of aggregated CNs.

- Simple errors that are only later discovered after they have been implemented into the design.

Process Hurdles for Change Propagation

The key component to identifying systematic propagation between change activities is informing parent-child relationships between CNs (or related directives). Generally speaking, the current CCB process does not require board members to discuss or document the understanding that a change action under review was driven by an error or incomplete impact analysis on a previous change action. Also, CNs are generally used as bins for proposed changes that were identified or documented at the same approximate time. While some IPTs create separate CNs for subsystem specific changes, this is the exception rather than the rule. Consequently, CNs can include completely unrelated changes from multiple documents because they were identified at the same development time. While not critical to implementing change propagation relationships, partitioning of proposed changes into subsystem specific CNs may benefit propagation analysis.

There are four primary challenges to adjusting processes for change propagation:

- *The Culture:* Social factors and the negative cultural view of technical changes may be the primary challenges to this implementation. Few engineers want to recount a causal relationship from a previous change for potential fear of culpability or jeopardizing interpersonal relationships.

- *Perceived Challenge of Cross-IPT Coordination of Relationships:* Some may argue that sub-IPTs that provide CNs to a program level CCB, may not be able to identify causal relationships that cross from one sub-IPT to another sub-IPT technical change. However, the program-level CCB (and it's engineering representative) probably already discussed these cross-relationships when synchronizing engineering functions. Also, sub-IPTs are often keenly aware of higher level requirements and exogenous interdependencies that affect their design solutions.

- *Departure from "CN Binning" would add unnecessary administrative costs:* This is unlikely because the average CN already contains 2-3 design documents, which leads to a higher frequency of CN generation. Partitioning the few CNs that contain four or more design documents would not significantly impact cost, and would likely improve both CCB discussion and the quality of relational data that is documented on those separate CNs.

- *Near Term Priority:* Some may argue that retrospective analysis is inconsequential to solving the present problem, which is an opinion that belies the fact that understanding systematic relationships informs present decision-making. As shown by previous research, systematic change relationships not only exist in the development of complex defense systems, but they have a causal relationship with expensive technical changes to subsystem interfaces and can provide useful leading metrics (Giffin, 2007).

Relation to Previous Change Propagation Research

The use of Hardware CNs to "bin" only contemporaneous but technically unrelated change action means that *sibling* relationships reported by Giffin (2007) may not always reflect a CP relationship. Consequently, the reduction of "CN Binning" is likely to contribute to quality and applicability of subsequent CP network analysis and metrics.

5.6 Information and Process View Element

While stakeholders felt that all necessary data was in the PDM system, they acknowledged that processing that data for required relationships was often a slow and arduous process. After indicating that the current PDM system lacked any efficient visualization capabilities (e.g. dashboards, histograms, etc), one experienced stakeholder used the terms "frustrating" and "unwieldy" to explain the system's inability to easily identify multidimensional relationships. This is not surprising because the complexity of ongoing

PDM system integration and development has focused on meeting many unique requirements that legacy PDM systems had evolved to provide to their business areas and customers.

While not related specifically to ECM knowledge or configuration information, the review of related business processes identified a similar lack of multidimensional data visualization. For instance, management uses a simple range bar to quickly review and make decisions about ECM productivity metrics. As a process requirement for most project proposals, the simple plot illustrates a linear range of program metrics, the average metric, and a selected historical program metric for comparison. The linear scale does not enable decision-makers to decipher stochastic relationships in the productivity data, which could lead to the 'flaw of averages'. While the distribution of metrics may be narrowed from the global distribution, they cannot indicate the mean value is the most probable value. Without a comparative probability distribution (e.g. data spark line or cumulative distribution function) of program data, it is more difficult to understand the probabilistic values and outliers. These particular references can shed light on the risk involved in proposal metrics or productivity improvement goals.

Relation to Previous Research

This issue of understanding complexity appears to align with reporting from Jarrett et al (2005), who found that both individual engineers and design teams were often lacking the tools or common experience to understand "the complex network of linkages". Research reported by Giffin (2007) provided network diagrams of propagated change between multiple subsystems, which may be useful to understand how a specific configuration evolved from specific design change actions. By relating color or distance between network nodes to specific change attributes, a PDM user may have a understanding of multi-dimensional dynamics within the system. Such attributes may include:

- Time or Project Event
- Subsystem Commodity Type
- Design Document Type
- Exogenous or Endogenous Change
- Defect Codes
- Defect Containment Attributes

5.7 Knowledge View Element: Application of Lessons Learned

Both enterprise processes and the engineering culture employ the lean principle of continuous improvement and Six Sigma practices. These activities include a mature and standard process to document lessons learned in an enterprise database. Lessons are captured from almost every project, and include detailed comments from various organizational functions across stages of development. This is not to say that every function provided lessons for every stage; while all functions are aware and encouraged to contribute to the database, there is not necessarily a hard wired trigger for documenting these lessons (i.e. they generally voluntary). However, hardware engineering functions have been the most consistent at providing regular and detailed inputs since the introduction of the database several years ago. While the larger firm employs high-level skill gap measures when hiring and internally sourcing manpower, there are no apparent skill gap measures employed for ECM knowledge.

Interestingly, it was found that lessons were less likely to apply to NPI projects. Considering Krishnamurthy's analogy of a hierarchal configuration tree that represented evolving and divergent configurations, discussions indicated that lessons usually only applied to specific configuration "branches" (Krishnamurthy, 1995). Generally, specific lessons were useful to redesigns (e.g. engineering obsolescence) of the same product configuration that spawned the lesson, but not applicable to later unique configuration "branches" that are more likely on NPI projects. Consequently, hardware engineering lessons learned are less likely to apply to NPI projects, which may impact effective diffusion of organization knowledge within similar product families.

Relation to Previous Research

It is worth noting the relationship between this finding and previous research in product families and ECM. First, this finding is related to research into complex product families, which indicates that product family configurations tend to diverge over time due to unique product requirements that drive part re-identification over time (Boas, 2008). Second, this observation relates to previous research reported Jarret et al. (2005), which indicated that "capturing experience and rationales...[is] so much of understanding possible change propagation". During the study of a engine design firm, it was noted that engineers felt that accessing previous related experiences enabled inexperienced engineers to "ask the appropriate questions" (Jarrett et al., 2005). This research appears to corroborate the general value of experiential knowledge, but indicates there is likely less value in the specific application to contemporary configuration challenges.

5.8 Process View Element: Software versus Hardware Technical Change

Interviews with CM coordinators and managers from across different NPI projects and business units indicated that hardware CN rejections occur on less than 5% of all change. Program defect profiles confirmed that less than 5 of more 2000 defects (from mostly hardware items) were rejected. Given the more rigorous hardware CN development process, higher rejection rates are very rare due to engineering review and oversight of proposed formal changes. Conversely, an interview with an experienced CM Coordinator, who works on both software and hardware changes across various programs, indicated that it is not uncommon for software CN rejections to occur on 20-25% of all related change.

Relation to Previous Research

The stark contrast between these general figures highlights that engineer specific analysis reported by Pasqual (2010) may not provide sufficient utility to hardware ECM practices. The Engineer Change Propagation Index (Engineer-CPI) measures the change that is propagated by individual engineers, and the Proposal Acceptance Rate (PAR) measures the approval rate for engineer proposed changes. While Pasqual found that the social network layer can directly contribute to programmatic change propagation, this research indicates those relationships may not generalize to normal hardware ECM activities. First, sparse instances of rejected CCB changes would not be significant enough to garner the interest of IPT Leads or accurately measure engineer performance. Second, individual engineer metrics wouldn't consider the natural fluctuations presented by engineer turnover between different programs, or the fact that many engineers may contribute to CN development, but only one will formally submit as the CN originator. Still other management decisions may complicate the value of individual engineer CP metrics to performance assessment. For instance, many projects will defer instead of reject those changes for later consideration or customer funding streams due to schedule pressure, earned value constraints, or customer direction.

5.9 Enterprise Stakeholder: Who is more relevant and why?

Access to ECM knowledge and collaborative discussion are the salient requirements for effective management of technical change. Figure 18 illustrates stakeholder saliency and highlights the intersection of legitimacy, power and criticality within the ECM social network (Nightingale, 2012). Here, design engineering, IPT leads, CM coordinators, and process engineering emerge as the most influential stakeholders.

Three attributes of saliency:

- *Power:* Program managers, customers and suppliers are powerful external stakeholders, but are either dormant or less suited to provide positive support due to demanding contractual relationships. However, more dominant Engineering Managers are in the best position to influence enterprise transformation because they represent culture, coercive authority, and understand the basic needs (i.e. Utilitarian) of stakeholders at the execution level.

- *Legitimacy:* CM managers have an evolutionary role in the improvement of supporting processes. However, their influence is derived from processes and systems that are not very dynamic.

- *Criticality:* Engineering and CM own the primary sub-processes that drive the ECM enterprise. When key configuration control and engineering action is required, the process people ensure that required stakeholders are present and sub-processes are properly sequenced.

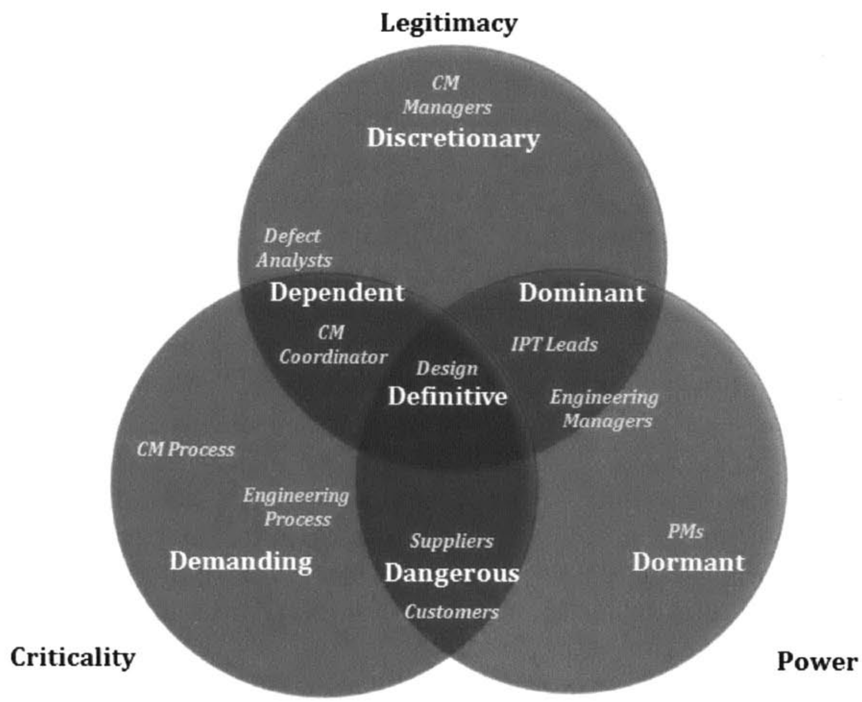


Figure 18. ECM Stakeholder Salience Venn Diagram.

5.10 Key Interactions

Analysis of intersections between key sub-processes with stakeholder saliency indentifies subtle yet important opportunities for enterprise alignment. While design engineers and IPT leads have the greatest ability to leverage such knowledge, informational barriers and the lack of systematic ECM analysis are primary limiting factors for these stakeholders. Ideally, CM coordinators and process engineers would clearly illustrate these relationships, which are necessary for the aggregation and diffusion of knowledge across design teams. However, several interactions act as friction to the effective transfer of organizational knowledge. These friction points include a less efficient PDM system, an administrative (versus analytical) CM focus, and the lack of change propagation (i.e. relational) enablers within the CM process. Finally, and perhaps most important, there is no vision or management concept that addresses these interactions in a holistic way. Considering the nature of these cross-element interactions and the need for a holistic vision, engineering management has the most leverage to sponsor policies than can synchronize the following enterprise misalignment:

- *No holistic ECM vision:* Focus on legacy standards and experiential learning without proper enablers reduces effective and efficient knowledge management.

- *Informational barriers:* Inefficient information exchange and lack of CP capability, which reinforces experiential dynamic.

- *Organizational Misalignment:* Administrative workflow focus of configuration SMEs (CM) does not adequately support engineering end-users, which reinforces experiential dynamic.

5.11 Soft Properties

The identification of soft properties is important to understanding the human dimension of an enterprise system because they often function as "leading indicators" of emergent enterprise properties (Nightingale, 2012). This enterprise exhibits conviction to established ECM sub-process interactions and experiential versus technocratic ECM knowledge.

5.12 Driven by Interactional Perspective or a Zeitgeist of the IPDP Paradigm?

The impact of hardware ECM to earned value management is clearly an important aspect of capturing new business and executing competitive contracts. Design and process engineering functions are aware of the need to identify and assess the impact of technical change, as well as efficiently accessing and processing product configuration data. But how much do external pressures upon design engineering to quickly innovate within an IPDP model actually drive the ECM interactions? Is there a more balanced view of external and internal actors that primarily influence the enterprise behavior? While admittedly influenced by the more interactive EA method, noteworthy findings indicate that internal interactions (i.e. across view elements) between various underlying legacy processes are the primary influences on the enterprise.

5.13 Summary

This descriptive study identifies opportunity for greater alignment between key ECM sub-processes and the technical information that stakeholders use to understand and mitigate systematic technical change. By integrating empirical knowledge of view element inter-relationships from stakeholder interviews, this research has identified key systematic factors that lead to a more complete characterization of the enterprise's internal and external landscapes.

6. ECM Enterprise Holistic Vision

6.1 Introduction

The descriptive case study identified several key sub-process interactions, soft properties, and cultural biases that illustrate opportunity for alignment across this Hardware ECM enterprise. The current enterprise is more focused on IPT or individual experiential knowledge, which is neither consistent nor scalable for expected future vision of greater cross-company collaboration and work-share. To address these needs, a holistic vision should articulate a more sustainable and technology-based strategy for organizational understanding and employment of systematic ECM relationships.

6.2 Holistic Vision Statement

Consistent with our goals to provide world-class innovative products and technology solutions, we must treat engineering change management (ECM) as a competitive engineering capability for capturing new business and achieving the highest standards of innovation. With an enterprise systems approach that synchronizes people, processes, and technology solutions, ECM should compliment integrated decision-making to mitigate technical change while preserving the innovative process.

Expected Delay in Value Delivery

Value delivery proposed by this holistic vision will require synchronizing policies with an expected mid-to-long term return on investment. While some short gains will come by introducing some tactical enablers, a greater capability can only be realized by aligning existing processes to collect data relationships on various projects from *Advanced Design* through the *Low-Rate initial Production (LRIP)* stages of development.

6.3 Proposed Stakeholder Value Exchange

Stakeholder value exchange is an EA method for comparing individual stakeholder importance within the enterprise to the perceived value those stakeholders receive from the enterprise. While value exchange focuses on the stakeholder view element, it takes into account value exchange as an emergent property of all dominant view element interactions that were identified in the exploratory and descriptive case studies. Figure 19 illustrates the current stakeholder value comparison and the changes that would be required to align stakeholders with key view element interactions. This vision would improve value delivery to design engineering and IPT leads by shifting CM resources to build systematic change processes and enablers.

Specifically, CM would align more closely with defect analysis and support to engineering decision-making by introducing changes to CCB process and PDM systems that enable systematic change surveillance. Design engineering and IPT Leads would introduce data-driven systematic change surveillance, metrics, and visualizations into their decision-making processes. And, process engineering would act as a moderator to ensure CM initiatives are synchronized with design needs. This assessment is driven by the following analysis of interactions:

Process-Information-Stakeholder

- *CM Alignment Opportunity:* The current CM focus should could shift from administration of the CCB process to more valuable statistical change analysis, and improvements to the current PDM system that would enable systematic change metrics and visualizations for the IPT and engineering users. As a growth opportunity, CM could leverage short term increases in productivity to provide resources, and the existing enterprise PDM "Burning Platform" to execute transformations.

- *Design Alignment Opportunity:* The design process and culture is for good reason experiential versus technocratic. However, management should consider the value of implementing systematic change models, visualizations and leading metrics into existing processes.

Knowledge-Stakeholder

A design and CM shift to more user-enabled surveillance of systematic relationships may require improved knowledge management practices. To enable this, enterprise leaders should consider aligning stakeholder incentives and IT solutions to encourage participation in knowledge documentation and collaboration across organizational boundaries.

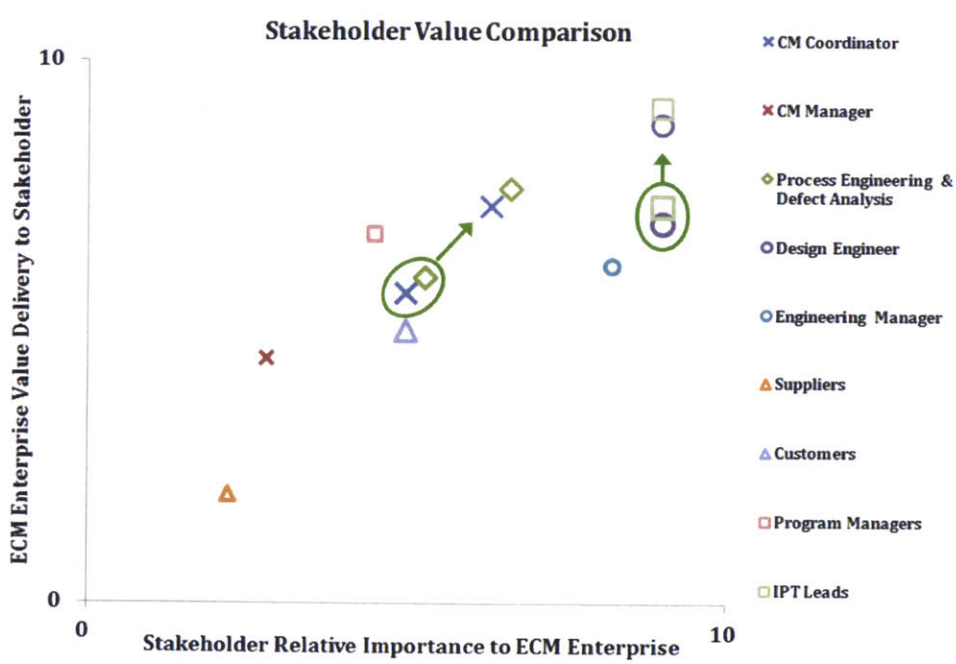


Figure 19. Stakeholder Value Comparison within ECM Enterprise.

6.4 Proposed Process and Stakeholder View

Analysis of stakeholder value delivery highlighted the intersection of process, information, and stakeholder view elements. Figure 20 illustrates the relative change in key process interactions that would support the holistic enterprise vision. The following shifts would increase the effective management of hardware change risk prior to design baseline:

- *Information Exchange Process:* The goal is to execute the shorter-term introduction of data mining models and PDM multidimensional visualizations that will enable design and process users to more quickly evaluate "Big Data" within the product development environment. Longer-term introduction of CP metrics and network graphical relationships can only happen with adjustment to the CCB process and PDM data collection.

- *CCB Configuration Control:* The goal is to improve future design and change impact analysis by first altering the CCB process to collect CP relationships, which are fundamental to the introduction of CP analysis and metrics.

- *Detail Design Peer Reviews:* The goal is establishing systematic change analysis capability, with greater process emphasis on CP metrics, defect modeling, and multidimensional data visualization that help reduce uncertainty.

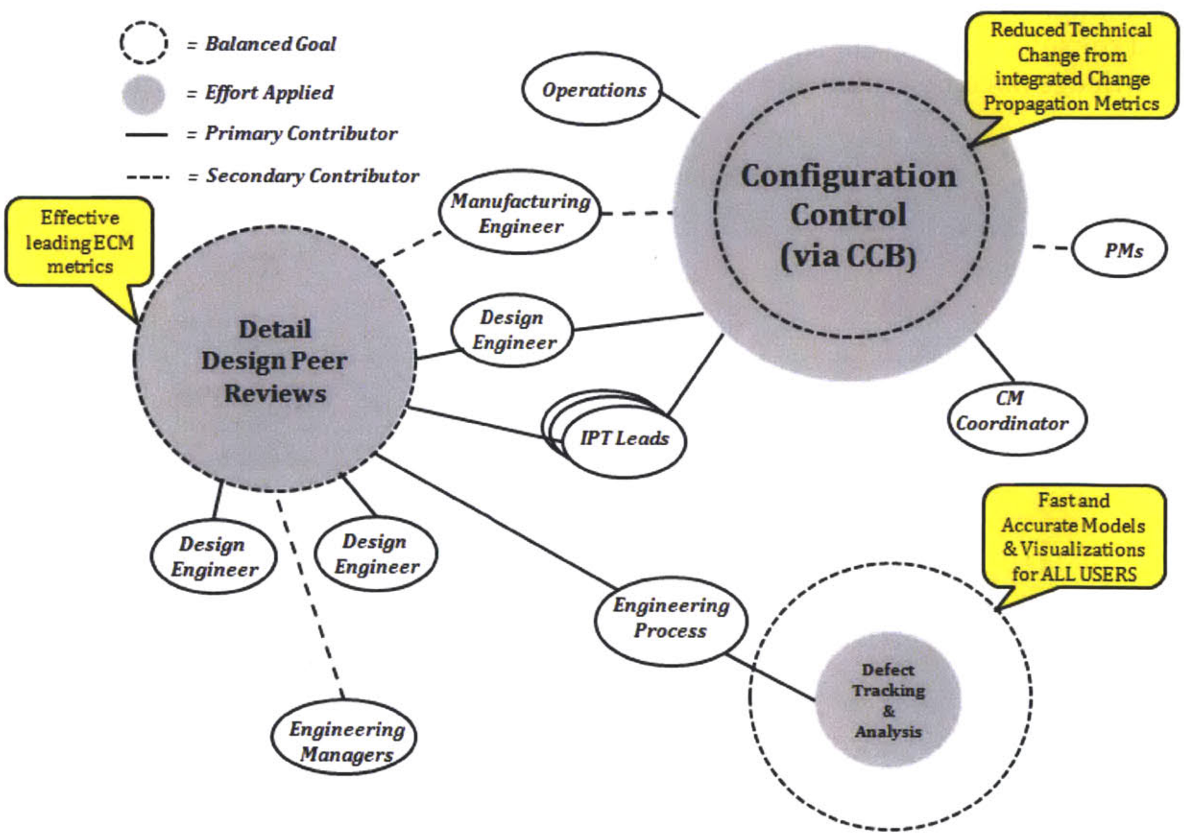


Figure 20. Network Diagram of ECM Stakeholders vs. Relative Effort Applied to ECM Sub-Processes.

6.5 Discussion of Conceptual Causal Relationships

The current enterprise system is characterized by causal relationships that reinforce positive effects or balance interactions with negative effects. Figure 21 illustrates these dynamic interactions with a conceptual causal loop diagram. There are four primary feedback loops:

- *Loop #1. Controlling Technical Change (Reinforcing Loop):* The previous standardization of legacy processes into the current PDM system was intended to increase productivity by:
  - Enabling all engineers (and other functions) to more directly and frequently access all configuration data within one PDM system and,
  - Integrating administrative workflows into the PDM system.
- *Loop #2. Information Barriers (Balancing Loop):* However, the knowledge transfer is not immediate, and the addition of numerous new users and process interactions increases complexity with more socio-technical interfaces. As more people use the system, more bugs and training is necessary for coordinated understanding. The resulting effort required to attain user needs (i.e. communication overhead) makes it more difficult to realize improved value. This limits significant improvements in collaborative productivity, which then limit the transfer of ECM knowledge. The net effect is to limit (i.e. balance) the effectiveness of Loop #1.
- *Loop #3. User-Centered Information (Reinforcing Loop):* Investment in various engineering user-centered functions reduces the effect of the communication overhead, which then results in increased knowledge transfer for controlling technical change. These functions may include multidimensional data models, dashboards and visualizations that would promote a common understanding of desired configuration relationships. The net effect is to reinforce the effectiveness of Loop #1.
- *Loop #4. Understanding Systematic Change (Reinforcing Loop):* Investment in processes that capture change propagation relationships and leverage associated metrics would increase the knowledge of systematic change risk. The net effect is to reinforce the effectiveness of Loop #1.

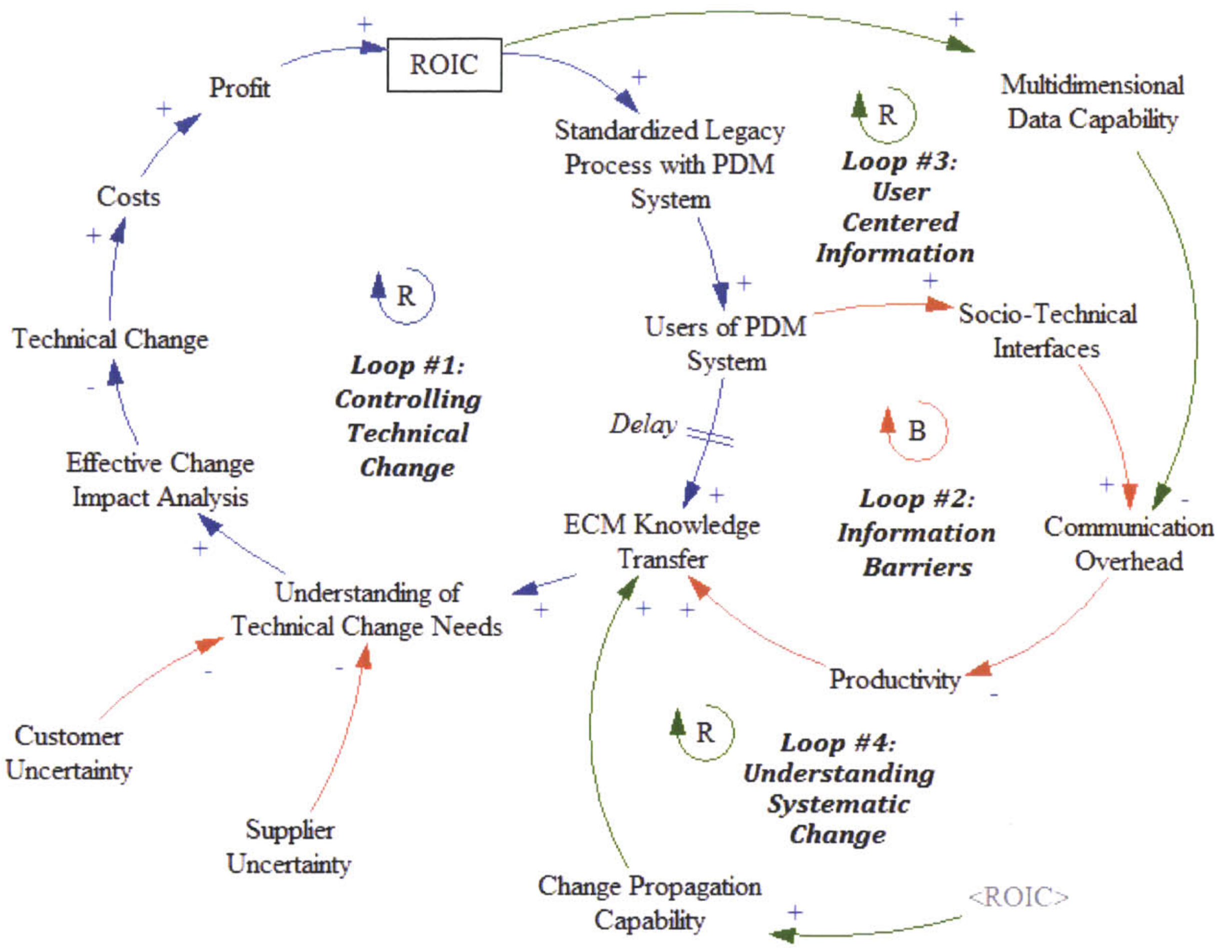


Figure 21. Conceptual Illustration using Causal Loop Diagram.

6.6 Summary

This enterprise vision requires a holistic approach to transforming the enterprise. While some data models and visualization enablers can offer near term value, the most significant operational capability and tactical metrics can only be attained from incremental collection of project data relationships over the course of several years. However, the timing of this transformation aligns with the greater company strategic vision to gain a competitive advantage by leveraging greater cross-business collaboration and one PDM system. Figure 22 summarizes the key view element interrelationships proposed by this holistic vision.

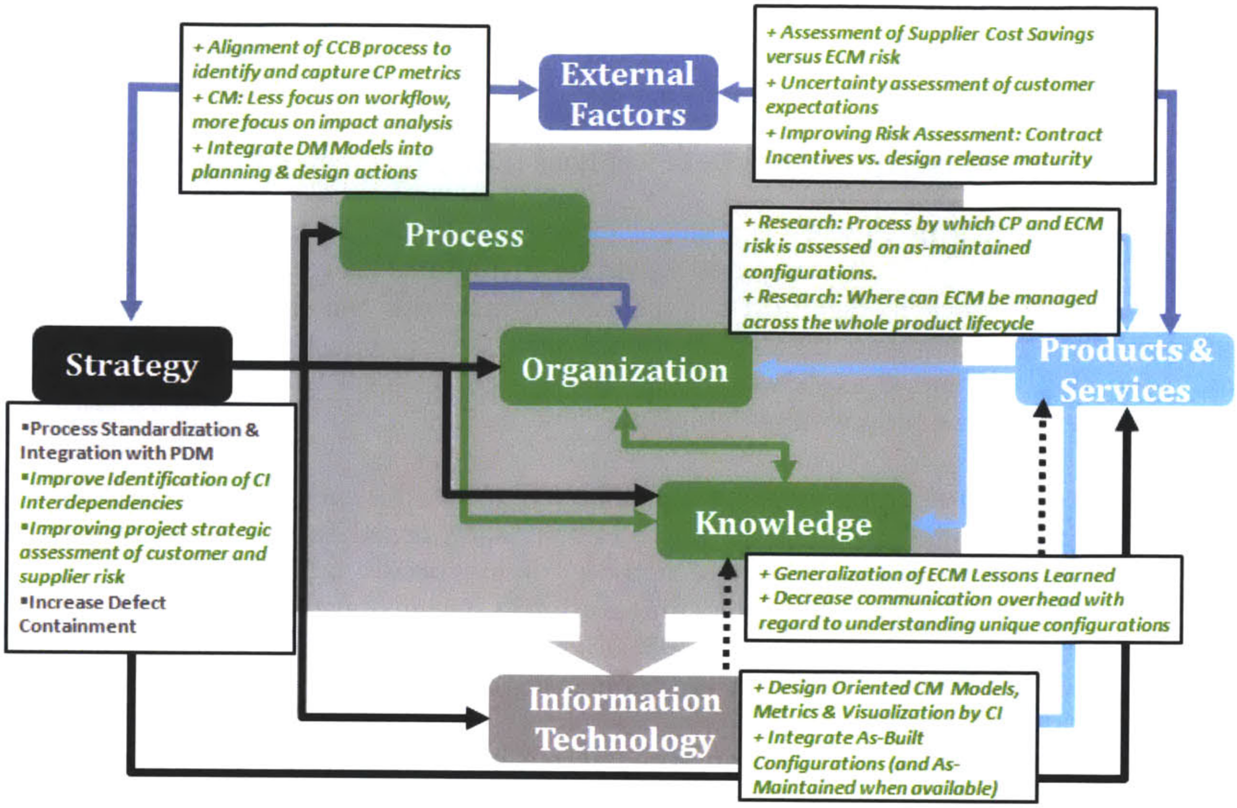


Figure 22. Adapted from ESD38 Lecture 6 (Nightingale, 2012). Holistic Vision with ECM Enterprise View Element Interrelationships.

7. Exploration of Data Mining Models in the ECM Enterprise

7.1 Introduction

The enterprise utility of data mining techniques is a second thesis question within this research. The intent of this section is to test our hypothesis, which states that classification data mining models can provide value by quickly deciphering relationships between defect codes, stages of development, and those defects that are uncontained. Furthermore, using these models for prediction may introduce a leading technique for mitigating change risk. The net effect would be to bolster design engineer awareness of probable defects, cue the allocation of resources, and provide a quantitative measure for assessing the impact of design changes.

7.2 Building a Program Defect Classification Model

This section will test the research hypothesis by attempting to create an accurate (<20% testing and validation error) and useful classification data mining model. Three separate iterations of testing will attempt to create increasingly complex models for different applications, and analyze them for discussion. All models will use Program #1 data for training, Program #2 data for testing and refining the model, and Program #3 data for independent validation of the final model, which will illustrate some generalization to new data (i.e. no over-fitting).

Hypothesis Testing vs Exploratory Data Analysis (EDA) of Selected Program Data

In order to identify if there were significant differences in defect attributes, *Unsupervised Learning* methods using frequency and bivariate histograms were used to test whether there was a difference in technical change activity between the three similar program defect profiles. Alternatively, if there were no assumptions or questions, this investigation would function as general exploratory data analysis for general learning. In this study, hypothesis testing was employed to determine:

- If there are interesting multivariate relationships, and if they confirm or deny with our understanding of defect activity. For instance, based on stakeholder interview data, the first bloom in change activity should occur in Stage 4, following the baseline of most detail design documentation in Design Stage 3.

- Which programs should be used to train, test, or independently validate the model.

Distinct similarities and differences emerged across categorical predictor variables of the three different development programs. By looking at variable plots, we see that all three programs have similar features across most of the categorical predictors. All programs exhibited similar distributions across the predictor variables, with Program #1 and Program #3 showing the most similarity (See Appendix A). There were notable variations in Program #2, which exhibited a higher defect containment percentage, lower relative defect percentage in the Stage 4, and slightly higher defect origination on specific types of change documents as shown in Figure 23. Though there were notable distributions, most uncontained defects for all programs were identified in the Stages 3 and 4 as illustrated in Figure 24.

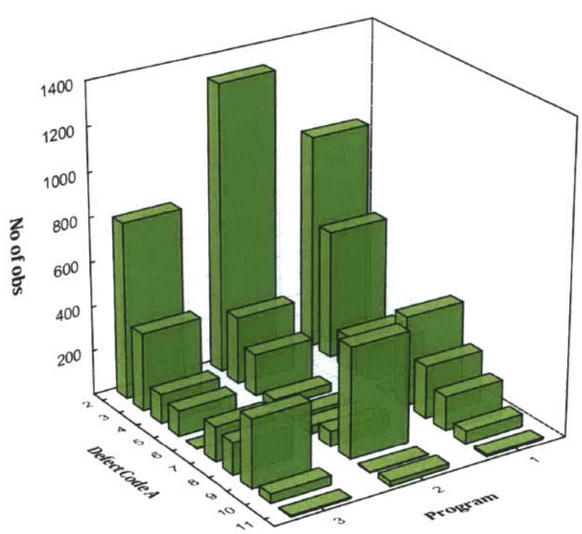


Figure 23. Bivariate Histogram of "Defect Code A" by Program.

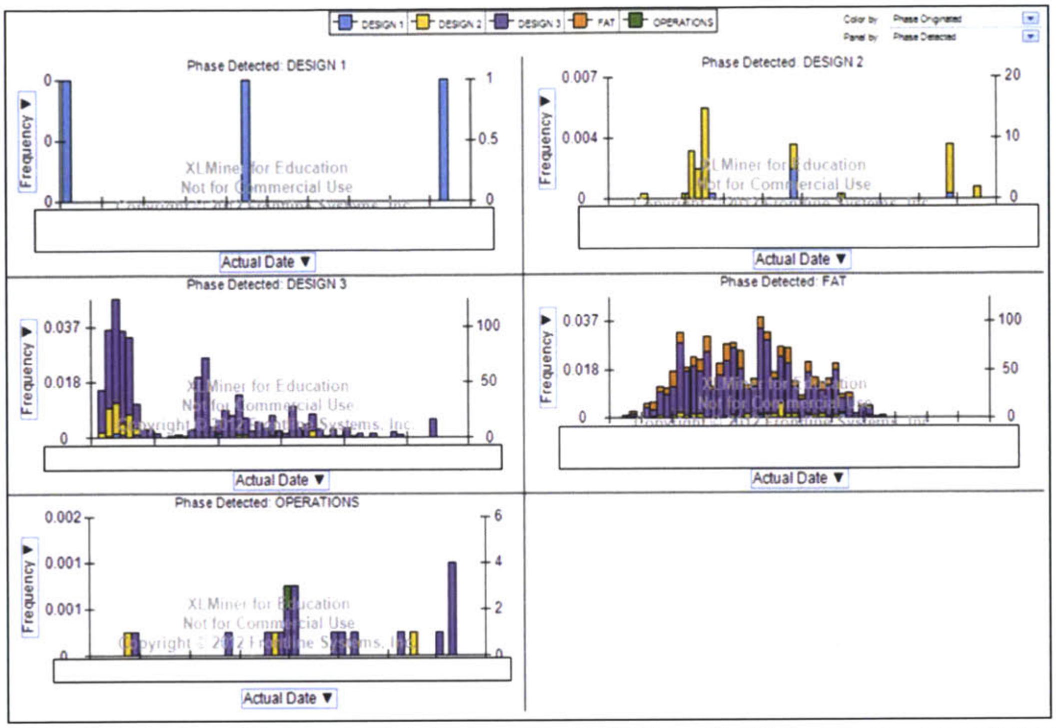


Figure 24. Defect Detected Frequency Histograms by IPDP Stage for Program #1 colored by Stage of Document Origination.

Finally, there were notable differences in the distribution of uncontained defects and their delay across the programs. For instance, Figure 25 shows how Program #2 exhibited much higher defect containment (Containment Delay = 0) than Programs #1 or #3, which both detected defects after three stages from their baseline origination.

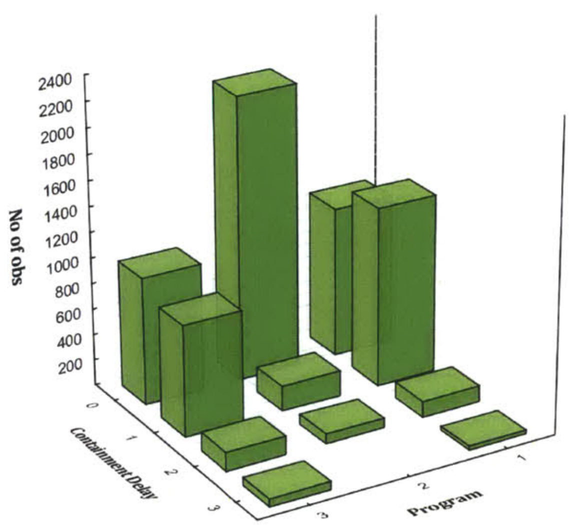


Figure 25. Bivariate Histogram of Stage Delay from Document Origination to Defect Detection by Program.

Therefore, this exploration found:

- Programs #1 and #3 are generally more similar, which indicates that Program #2 should be included in model testing to capture relationships.
- Distributions of defect codes and stage dynamics are generally comparable with some fluctuations.
- Programs #1 and #3 exhibited uncontained defects that were delayed by as many as three stages, and occurred as often as contained defects.

Feature Selection

Moving onto statistical techniques, Pearson Chi-Square testing was employed to identify the most significant relationships due to the discrete categorical nature of the data. The measure is based on calculations of expected frequencies in a two-way table. If there is no relationship, then we expect there to be an equal number of two-way table choices. However, if there is a relationship, then Chi-Square values will become increasingly significant with greater deviation from an equal two-way pattern (Statsoft, 2012). Figure 26 shows the results of the Chi-Square testing, which clearly shows the relationships that we already identified from hypothesis testing with unstructured learning methods.

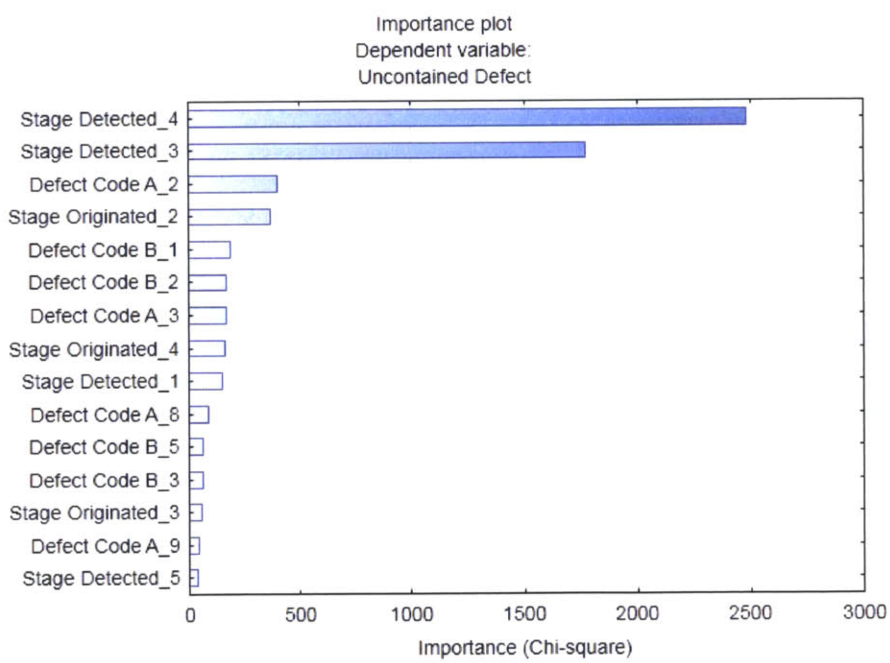


Figure 26. Importance Plot for Program #1 and #2.

Methods for Increasing Accuracy and Generalization

Stratified Random Sampling was used to enable the algorithm to systematically over-sample rare observations (i.e. *Uncontained Defect* variable) from Program 1 training data and Program 2 testing data. To further reduce the risk of over-fitting, the option for *pruning on the minimum error tree* was a standard selection for the CART algorithm. Within this selection, CART first creates the full tree model, and then leverages samples that were not used as "prune" back branches to find the simplest final tree that retains the same accuracy (Statsoft, 2012). However, the most successful models employed *V-fold cross-validation*, which ran in concert with stratified random sampling to use successive samples of testing data to improve the accuracy and reduce the complexity of the tree structure.

Multiple Models and Lift Charts

Statistica Data Miner Recipes enabled the parallel development of multiple models for accuracy comparison and evaluation of lift charts. Model lift charts provide a visual summary of the usefulness of one or more models in comparison to a baseline condition. This baseline (i.e. no relation to configuration baseline) is the condition where there is no model, which simply indicates the probability that if a classifier value occurs (i.e. classifier is true) it would be the same as the probability of that value naturally occurring in historical data. Higher "Lift Values" indicate the relative performance given the percentage of known data, though all lift values eventually diminish to the baseline (1.0 lift value) when 100% of the data is known. Lift charts are therefore important to comparing the performance of alternative models.

7.3 Model Building - Iteration #1: Classifying Defects related to Stage 3

The intent of this first iteration was to crawl before we walk with a simpler model of uncontained defects with (1) higher importance defect codes as defined by Pearson Chi-Square testing, (2) defects detected in Stage 3, or (3) defects whose parent documents were originated in Stage 3. Table 2 identifies these selected variables.

	Variable name	Type	Role
1	Defect Code A_2	Categorical	Input
2	Defect Code A_3	Categorical	Input
3	Defect Code A_8	Categorical	Input
4	Defect Code A_9	Categorical	Input
5	Defect Code B_1	Categorical	Input
6	Defect Code B_2	Categorical	Input
7	Defect Code B_3	Categorical	Input
8	Defect Code B_5	Categorical	Input
9	Stage Detected_3	Categorical	Input
10	Stage Originated_3	Categorical	Input
11	Uncontained Defect	Categorical	Target

Table 2. Iteration #1 - Input and Target Variables for Program #1 & 2 Model.

The first iteration independently developed a CART model, Boosted Trees model, and a Neural Network model for comparison. The Boosted Trees and CART models provided the best overall performance with 3.63% training error on Program #1 and 5.11% error with testing on Program #2. Table 3 shows the classification confusion matrix for the CART model, which was selected as the best candidate model for independent validation against Program #3 due to its simplicity and easily interpreted Tree Graph.

	Summary Frequency Table (Prediction) Table: Uncontained Defect(2) x Model-1-Prediction(2)			
	Uncontained Defect	Model-1-Prediction 0	Model-1-Prediction 1	**Row Totals**
Count	0	416	9	425
Column Percent		91.83%	2.01%	
Row Percent		97.88%	2.12%	
Total Percent		46.22%	1.00%	47.22%
Count	1	37	438	475
Column Percent		8.17%	97.99%	
Row Percent		7.79%	92.21%	
Total Percent		4.11%	48.67%	52.78%
Count	All Grps	453	447	900
Total Percent		50.33%	49.67%	

Table 3. Iteration #1 - Classification Confusion Matrix for Tested CART Model on Program #1 & 2 Data.

Figure 27 shows model lift charts for classification of *contained defects* (value = 0) and *uncontained defects* (value = 1). Generally, these charts have very similar 2.0 lift values when between 10-40% of data is sampled and tested. While the CART model looses lift for both selected defect containment values, it was still selected for its relatively high accuracy, simplicity, and interpretability of the tree graph as shown in Table 3 and Figure 27.

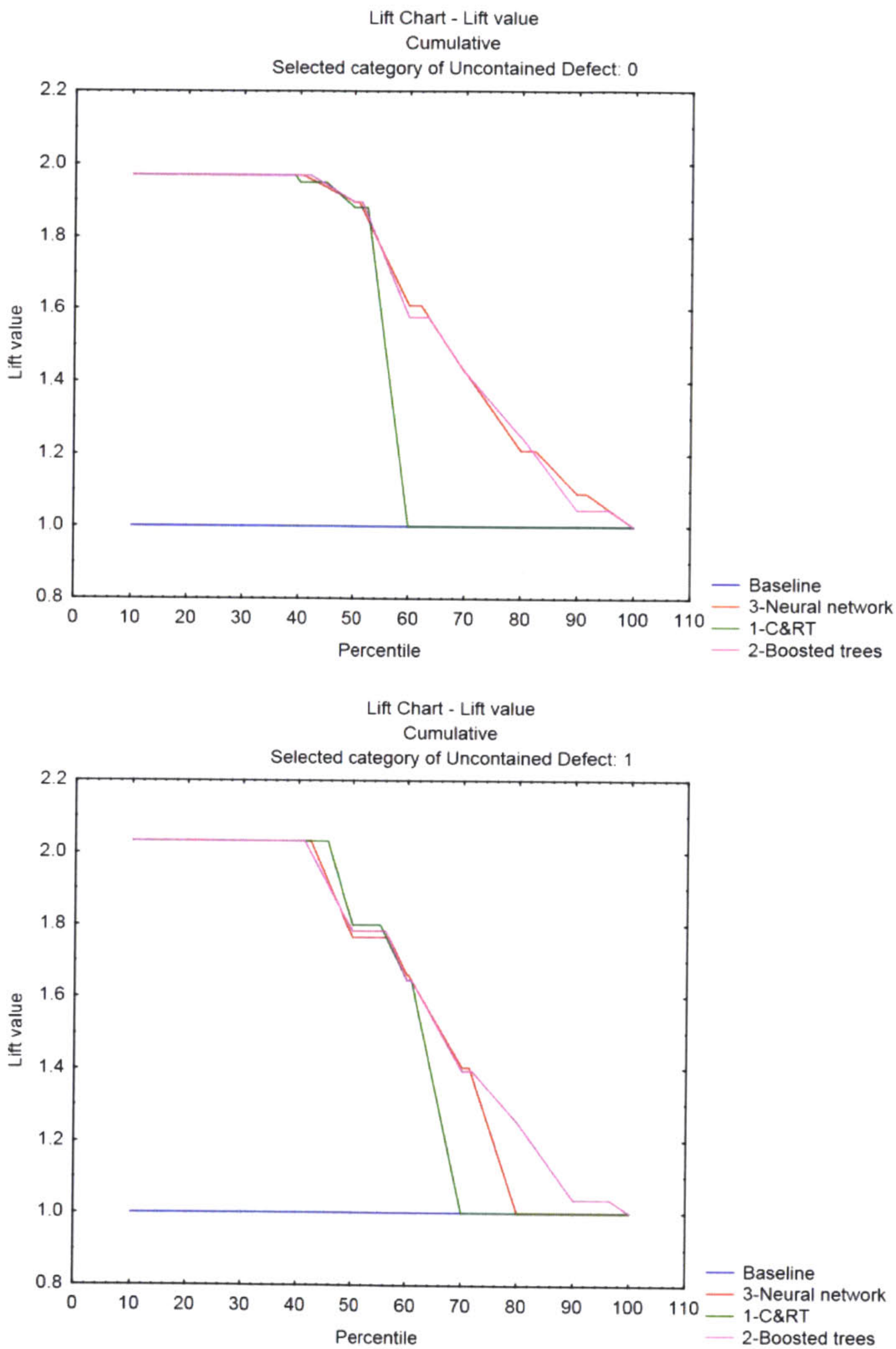


Figure 27. Iteration #1 - Lift Chart for Several Models trained on Program #1 & #2 Data.

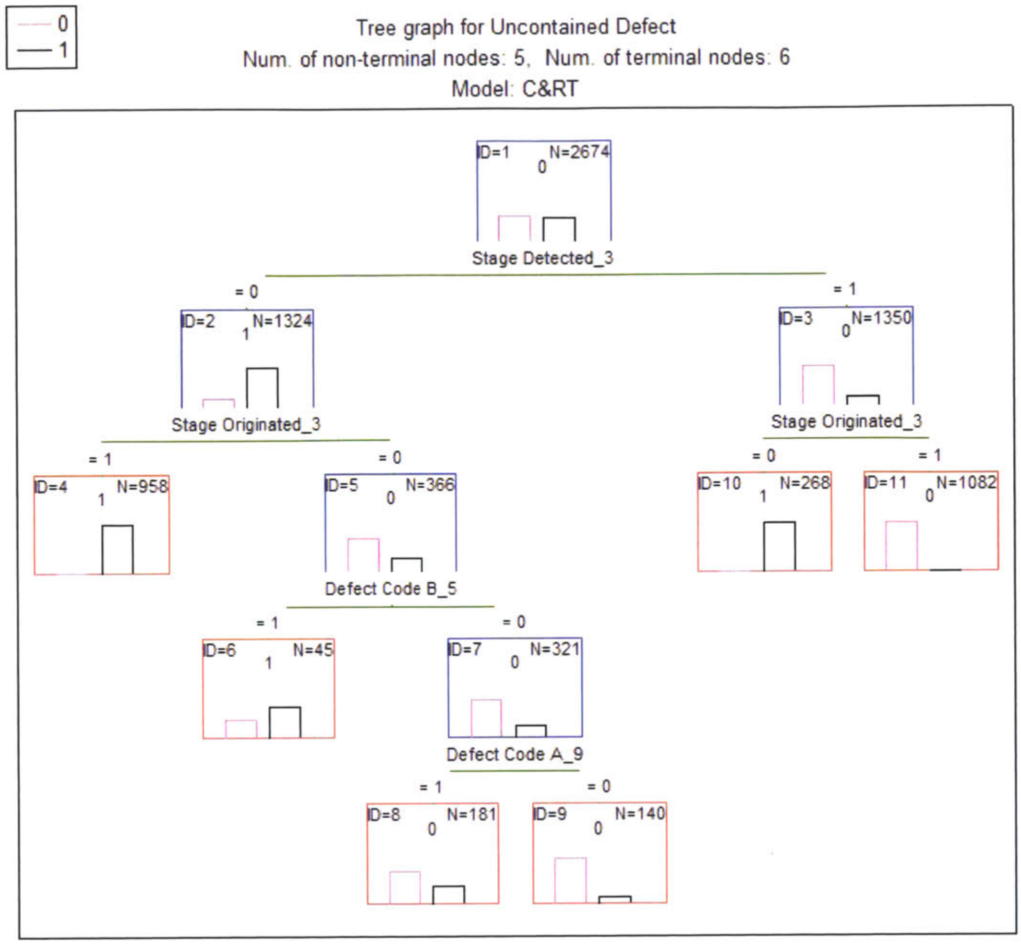


Figure 28. Iteration #1 - CART Model Graph using Programs #1 & #2 Data with 5% error.

Figure 28 shows the CART Tree Graph, which enables the user to evaluate the behavior of the model and determine relationships within the data. We know the model used data from both Programs #1 & 2 with 5174 total defects, including 1807 uncontained defects and 3367 contained defects. However, by removing the outliers, the model is only using 2674 observations. The tree interpretation is fairly straight forward by following the binary rules through decision nodes (blue) to the terminal leafs (red), each of which contain data that explain the decision or terminal results. For instance the top decision node has the following characteristics:

- The unique ID number (e.g. ID=1) for reference purposes.
- The number of total Uncontained Defects considered between training and testing data (e.g. N=2674).
- The histogram of observations within that node with the dominating target variables value.

- Decision nodes have attributes at their branches that start or continue an "If X, Then Y" statement that eventually leads to a terminal value (e.g. Stage Detected_3).

Right Branch and Terminal Leaves:

- If Stage Detected_3 (value=1) and not Stage Originated_3 (value=0), then there are 268 uncontained defects (Terminal Node ID=10).

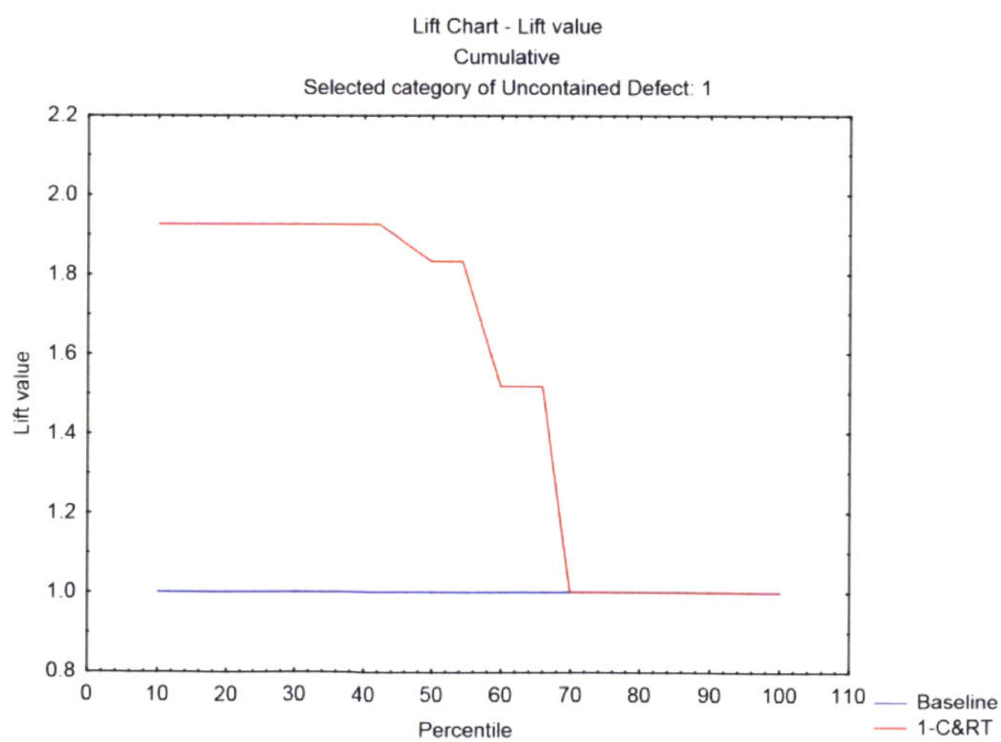
- If Stage Detected_3 (value=1) and Stage Originated_3 (value=1), then there are 1082 contained defects (Terminal Node ID=11).

Left Branch and Terminal Leaves:

- If not Stage Detected_3 (value=0), and Stage Originated_3 (value=1), then there are 958 uncontained defects (Terminal Node ID=4).

- If not Stage Detected_3 (value = 0) and not Stage Originated_3 (value = 0), and then there are 9588 uncontained defects (Terminal Node ID = 4); et cetera.

The deployed CART model was then independently validated against Program #3 to simulate a real world deployment and to verify there was no over-fitting to the training and testing data. Figure 29 shows the results of independent validation that retained an average 2.0 lift value with only 9% error.



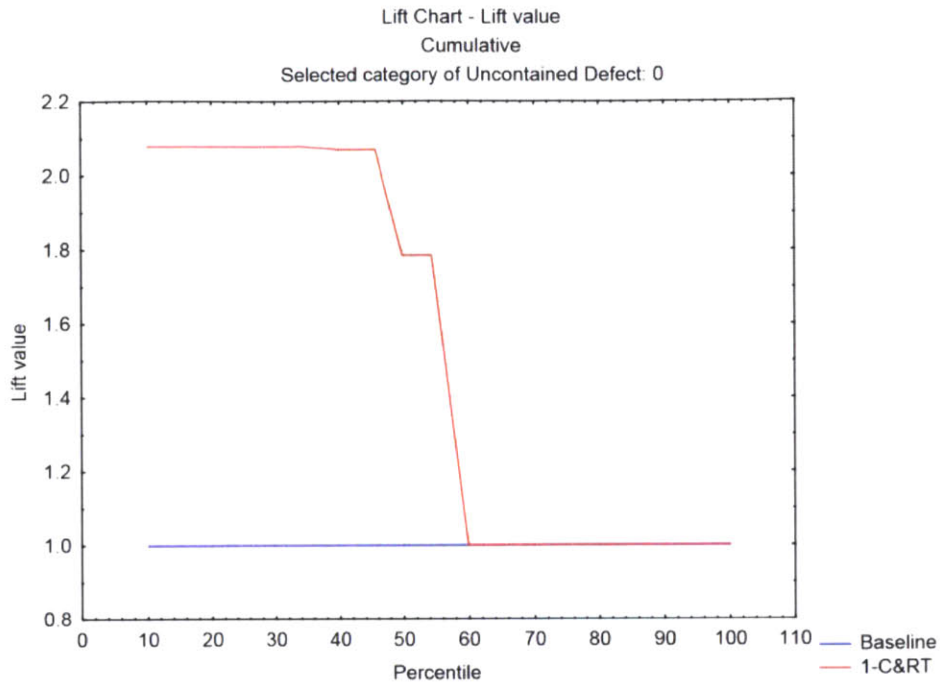
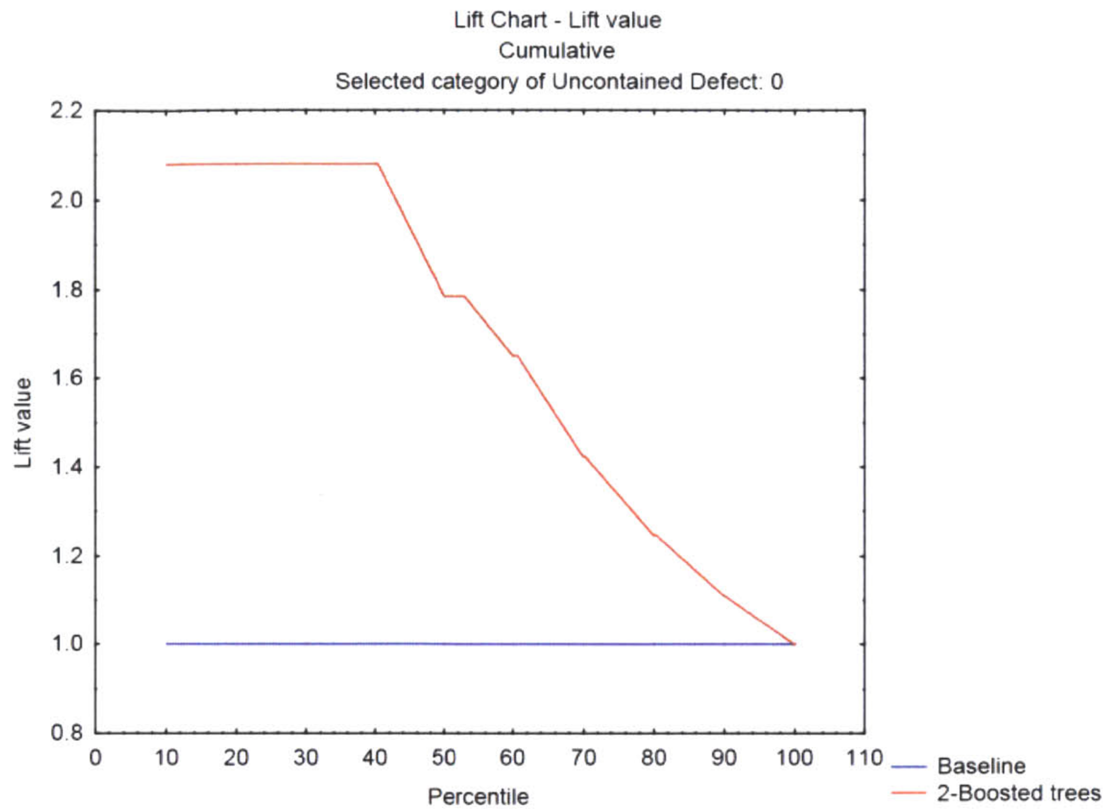


Figure 29. Iteration #1 - Lift Chart for CART Model deployed against Program #3 Data with 9% error.

Alternatively, the Boosted Tree Model was then independently validated against Program #3 to verify there was no over-fitting. Table 4 and Figure 30 show the results of independent validation that confirm it performs no better than the CART model with an average 2.0 lift value and an average 9% error. Therefore, this iteration validates the hypothesis that a classification model of defects is not only possible, but also accurate and useful in illustrating simple data relationships.

	Summary Frequency Table (Prediction) Table: Uncontained Defect(2) x Model-2-Prediction(2)			
	Uncontained Defect	Model-2-Prediction 0	Model-2-Prediction 1	Row Totals
Count	0	416	9	425
Column Percent		91.83%	2.01%	
Row Percent		97.88%	2.12%	
Total Percent		46.22%	1.00%	47.22%
Count	1	37	438	475
Column Percent		8.17%	97.99%	
Row Percent		7.79%	92.21%	
Total Percent		4.11%	48.67%	52.78%
Count	All Grps	453	447	900
Total Percent		50.33%	49.67%	

Table 4. Iteration #1 - Classification Confusion Matrix for Tested Boosted Tree on Program #1 & 2 Data.



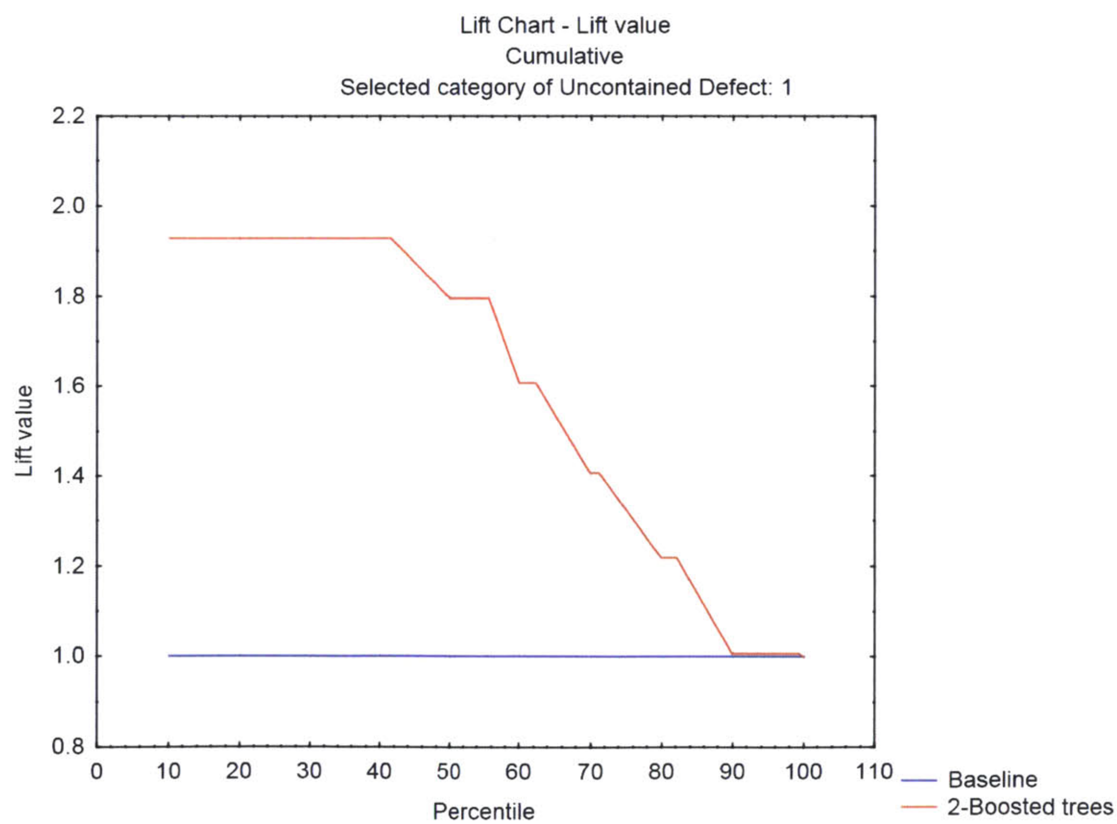


Figure 30. Iteration #1 - Lift Chart for Boosted Tree Model deployed against Program #3 Data with 9% error.

7.4 Model Building - Iteration #2: Classifying Defects Across Multiple Stages

The intent of this second iteration was to build a model of uncontained defects with broader applicability using (1) higher importance defect codes as defined by Pearson Chi-Square testing and (2) defects whose parent documents were originated in Stages 2, 3, or 4. While several defect codes were not included due to insignificance or invariance, in this iteration the defect codes were binned to match the two-tier structure (i.e. Defect Code A-B). Even with this binning, more *n*-dimensions increased the feature complexity. This effort will provide further fidelity that more complex relationships can be accurately modeled. Table 5 identifies these selected variables.

	Predictor importance Response: Uncontained Defect Model: C&RT	
	Variable Rank	Importance
Cause Code_702	100	1.000000
Cause Code_502	95	0.948805
Stage Originated_STAGE 4	81	0.811976
Cause Code_206	77	0.770631
Cause Code_701	77	0.768881
Cause Code_301	63	0.628647
Cause Code_207	61	0.610509
Cause Code_307	60	0.603467
Stage Originated_STAGE 3	59	0.588746
Cause Code_202	43	0.432127
Stage Originated_STAGE 2	42	0.422673
Cause Code_704	34	0.335875
Cause Code_302	26	0.256227
Cause Code_203	23	0.229389
Cause Code_401	21	0.208931
Cause Code_504	21	0.214842
Cause Code_A9	21	0.211006
Cause Code_201	20	0.199912
Cause Code_A8	13	0.132836
Cause Code_703	8	0.076376
Cause Code_406	1	0.006829

Table 5. Iteration #2 – Variable Ranking (Pearson Chi-Square) Chart for Input and Target Variables.

A CART model, Boosted Trees model, and Neural Network model were again independently developed on Program #1 & 2 data. In this iteration, the CART model was the clear winner with 18.7% testing error, which is better than the general 80% accuracy threshold. The Neural Net and Boosted Trees performed relatively poorly with 81% and 58% training error rates respectively. Table 6 and Figure 31 show the classification confusion matrix for the CART model, where we see the prediction of *uncontained defects* is more accurate (97%) than that of *contained defects* (i.e. false positives for *uncontained defects*). This is the accuracy that matters to decision-makers, since there are unequal misclassification costs when attempting to identify an *uncontained defect*. The enterprise would rather error on the side of a false positive with little relative cost, rather than miss an opportunity to analyze a more detrimental uncontained defect relationship.

Summary Frequency Table (Prediction) Table: Uncontained Defect(2) x 1-C&RT Prediction(2)	Uncontained Defect	1-C&RT Prediction 0	1-C&RT Prediction 1	**Row Totals**
Count	0	926	267	1193
Column Percent		99.36%	50.38%	
Row Percent		77.62%	22.38%	
Total Percent		63.34%	18.26%	81.60%
Count	1	6	263	269
Column Percent		0.64%	49.62%	
Row Percent		2.23%	97.77%	
Total Percent		0.41%	17.99%	18.40%
Count	All Grps	932	530	1462
Total Percent		63.75%	36.25%	

Table 6. Iteration #2 - Classification Confusion Matrix for Tested CART Model on Program #1 & 2 Data.

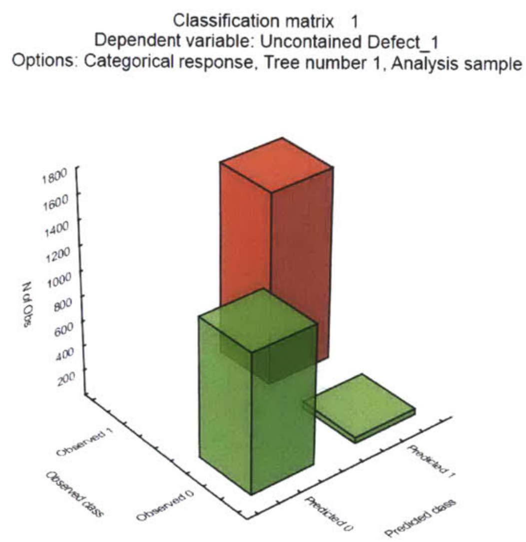
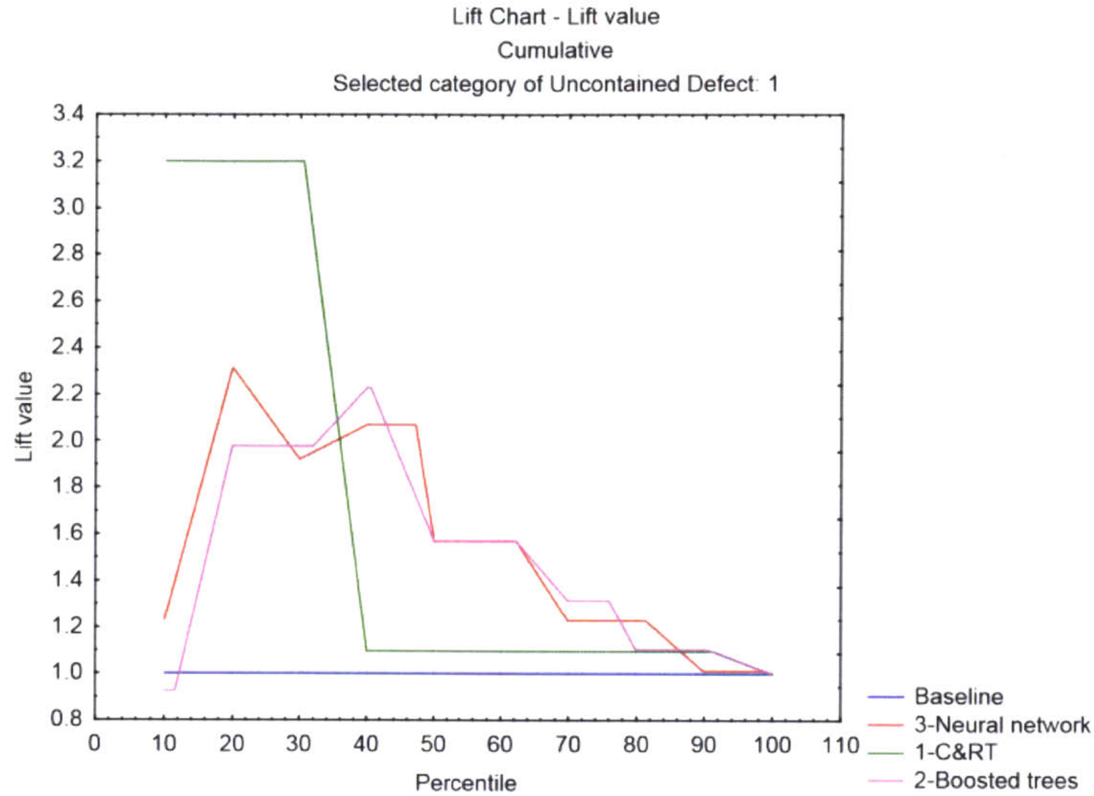


Figure 31. Iteration #2 – Graph of Classification Confusion Matrix for Tested CART Model for uncontained defects (Value = 1).

Figure 32 shows model lift charts for classification of *contained defects* (value = 0) and *uncontained defects* (value = 1). The *uncontained defect* classification shows significant 3.2 lift value with 10-30% of known data. Conversely, the *contained defect* classification shows a rather insignificant 1.22 lift value. However, the model performs very well in comparison to the other two models when we consider the simplicity of the algorithm.



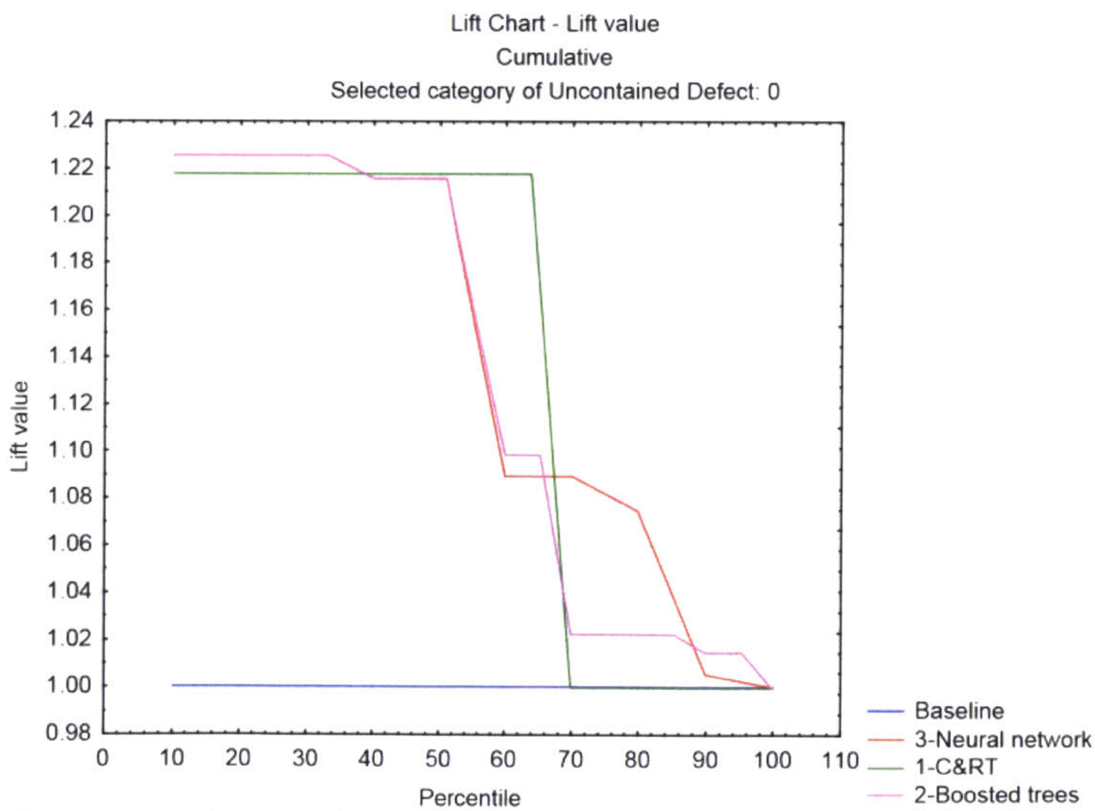


Figure 32. Iteration #2 - Lift Chart for Several Models trained on Program #1 & #2 Data.

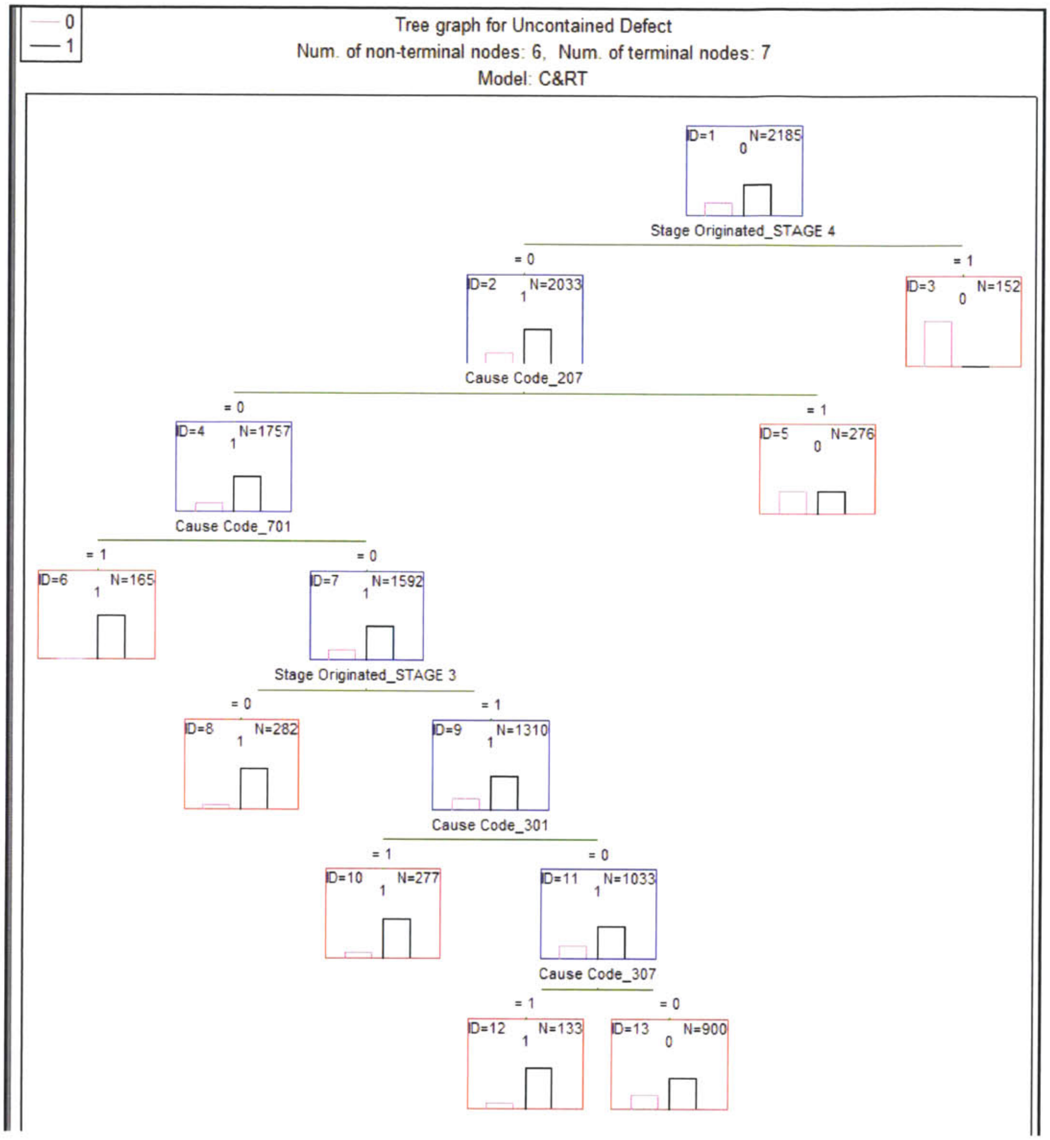


Figure 33. Iteration #2 - CART Model Graph using Programs #1 & #2 Data with 18.7% error.

Figure 33 shows the CART Tree Graph from the second iteration, which again illustrates significant data relationships. Specifically, it provides a breakdown of several groups of uncontained defects including approximately 19% code-0207, 33% not originated in Stage 3, 32% Stage 3 originated with code-301, and 16% Stage 3 originated with code-307.

In comparison to the first iteration, the deployed CART model was less effective when independently validated against Program #3 with a 34% error rate. While this test failed to meet our 20% error threshold and indicated some over-fitting to Program #1 and 2 data, the tree graph still presents some relational value. This is bulk of the error is associated with We also see in Figure 34 the corresponding reduction in lift performance to 1.4 and 1.8 lift respective values.

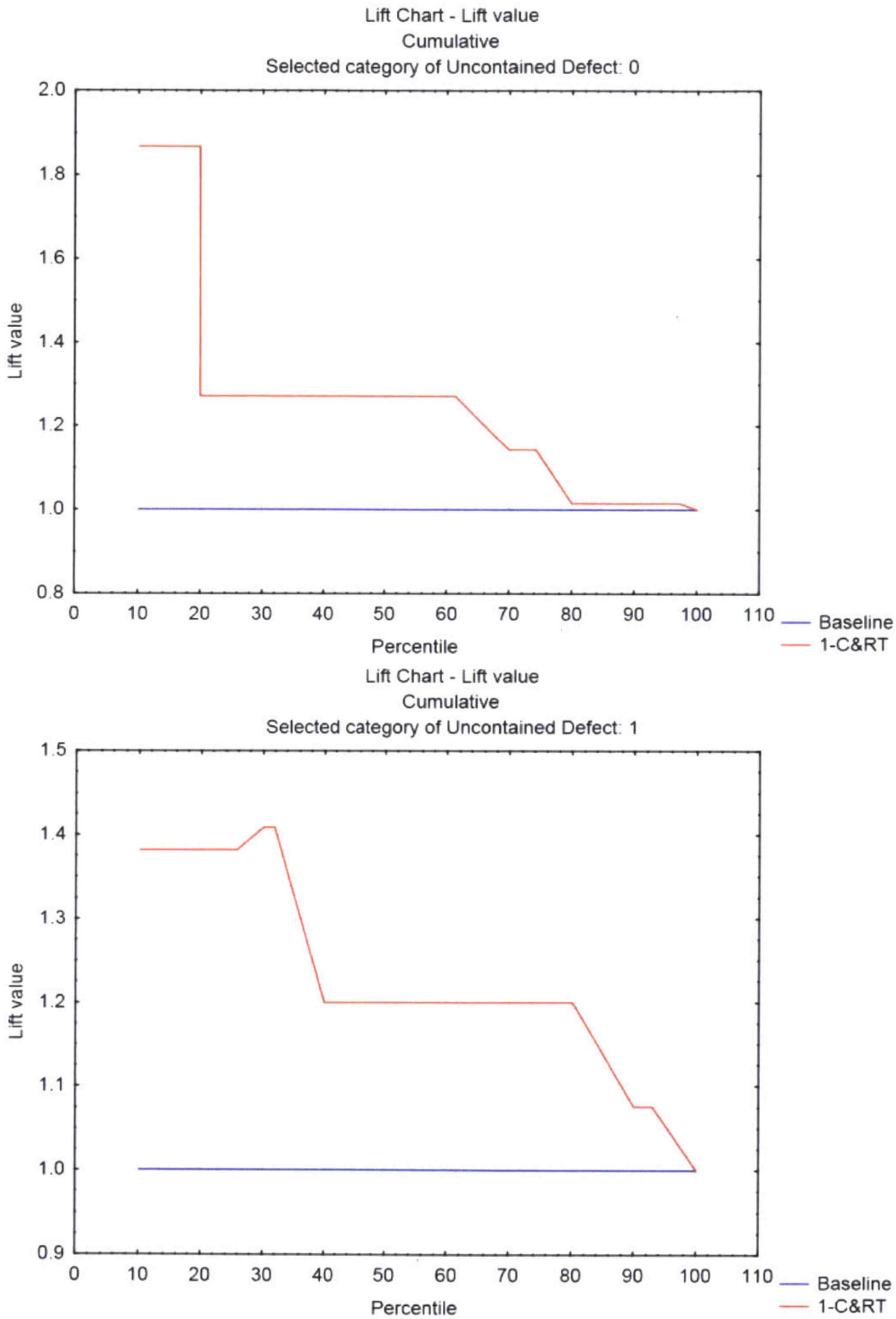


Figure 34. Iteration #2 - Lift Chart for CART Model deployed against Program #3 Data with 34% error.

7.5 Model Building - Iteration #3: Predicting an Uncontained Defect

A predictive model would be valuable to detail design teams, who review subsystems prior to their initial baseline release and ensure the design documentation meets requirements. For instance, a predictive model could indicate if a design document was likely to exhibit an uncontained defect during the product development lifecycle. For this iteration, only defect attributes that were known in early design stages were used to simulate a predictive model. Consequently, only stage originated and likely defect codes were used for experimentation.

However, testing failed to produce an accurate predictive model with between 35-65% misclassification error on testing. Various algorithms were employed including Neural Networks, CART, CHAID, Boosted Trees, MARS, Support Vector Machines and Random Forests. Several ensembles using up to five MLP-based Neural Networks were also employed, but failed to perform better than 40% error testing or independent validation. Additionally, feature complexity (i.e. input variables) was varied between one to four stages of development; this included configurations of between three and fifteen input variables against an *uncontained defect* target variable. Table 7 shows the custom prediction function of the best performing (although unsatisfactory) MARS model.

Independent and dependent variables	Custom predictions Value
Stage Originated_4	0
Stage Originated_3	1
Stage Originated_2	0
Stage Originated_1	0
Defect Code A_2	1
Defect Code A_3	0
Defect Code A_7	0
Defect Code A_8	0
Defect Code A_9	0
Defect Code B_1	0
Defect Code B_3	0
Defect Code B_2	1
Defect Code B_5	0
Defect Code A_10	0
Defect Code A_11	0
Defect Code B_7	0
Defect Code B_6	0
Defect Code B_4	0
Defect Code A_6	0
Defect Code A_5	0
Defect Code A_4	0
Uncontained Defect	0

Table 7. MARSpline Prediction Result for Uncontained Defect attribute with 35% Testing Error.

7.6 Near Term Likelihood of Employment

Based on knowledge of the enterprise PDM system capabilities and CM practices, a predictive model may soon be possible once existing defect profiles are associated with configuration items (CI) and their specific attributes including *Commodity Type* and *Document Type* attributes. This data is tracked by projects, but not necessarily collected in an integrated process that enables the analysis of one complete profile for a given CI. While these relationships were unavailable at the time of this research, the firm is aggregating these relationships across historical project defect profile to enable greater depth of analysis.

7.7 Concepts for Data Mining Models and Change Propagation Metrics

The importance of understanding day-to-day aspects of operational processes and dependent product data management cannot be understated. Even within the same company, specialized functions that operate within the same business area or program may not fully understand all of the technical processes and product data interactions with co-lateral functions. Previous research indicates that internal change is often difficult in light of cultural assumptions that may promote mismatching between problem root causes and their corresponding solutions (Klein, 2004). Furthermore, even if proper alignment is identified with small but important aspects of process, the inertia required to institute lasting change may be underestimated.

In the selected enterprise, the argument for improving systems that identify technical change risk is somewhat complicated by the enterprise culture. This is particularly important in the context of an enterprise that leans more towards experience-based versus technocratic decision-making (Klein, 2004). Like many modern but specialized organizations, this enterprise relies both on experience within a functional area and demonstrated technical competence as the basis for legitimacy. With fewer proactive and synchronous measures for managing risk of technical change, this hardware engineering change management enterprise leans more towards experience-based decision-making. With the recent implementation of a powerful and new PDM systems, which are also closely integrated with MRP systems, there exists an opportunity for integrating data collection measures that can support proactive change analysis.

Adjusting the CM Process and Data Collection to support identification of Change Propagation

Configuration Management directives establish the propagation phenomena as a strategic imperative, the vision for employment, and the value proposition for ECM stakeholders. Supporting standard procedures would outline sub-process requirements for the following:

- Definition of key metrics.
- CCB process requirements for documenting CP relationships.
- Application guidelines within ECM sub-processes.

Implementation of Data Mining Models into ECM Information Systems

The existing ECM enterprise can implement data mining classification models to improve user understanding of project defect data with automated analysis and visualizations. Furthermore, if complete CI-specific defect profiles can be established, then those additional data features may enable the creation of a predictive defect model or collection of smaller models.

Implementing Project Defect Classification Models into Defect Containment Systems

C&RT Models can produce fast and accurate models for classifying both linear and non-linear project defect relationships. This would include visualization outputs of trees that any practitioner could easily interpret. In addition, more robust models such as Boosted Tree, Neural Network, MARS or ensemble models can be integrated to provide classification models for existing project data. These models and related analysis can provide:

- Classification response for understanding non-linear defect relationships.
- Prediction response for non-linear defect relationships of a subsystem being analyzed.

Integration of metrics into a Design Review Dashboard

The existing enterprise uses networked peer review tool to schedule design reviews for specific subsystems and components, and documents meeting results to track compliance with engineering processes. The integration of predictive defect and change propagation metrics into this tool would increase awareness of risks and cue corresponding design discussions. Figure 35 illustrates a conceptual view of a dashboard that integrates these metrics.

Peer Review	Configuration Item	Commodity Type	Document Type	Probable Change Characteristic	Probable CAI / CRI / CPI	Predicted Defect Class & Number of Defects		
						Uncontained Defect	Defect Code A	Defect Code B
1	Subsystem 1	Antenna Equipment	Assembly Dwg	Acceptor	CAI Metric	0	X	X
2	Subsystem 2	Radome	Assembly Dwg	Multiplier	CPI Metric	X	X	X
3	Subsystem 3	Power	Detail Dwg	Reflector	CRI Metric	X	X	X
4	Subsystem 4	RF Antenna - CCA	Detail Dwg	Acceptor	CAI Metric	0	X	X
5	Subsystem 5	Structure	Purchased Item Dwg	Acceptor	CAI Metric	0	X	X

Figure 35. Conceptual Design Peer Review Dashboard using Predicted Defect and CP Metrics.

7.8 Summary

Data mining models are both executable and likely to provide value to stakeholders who are looking for automated and accurate models of technical change relationships. Specifically, this analysis evaluated CART models for classifying defects from similar projects, and concluded they can accurately generalize relationships to similar projects. Predictive models would provide higher comparative value if paired with existing hardware detail design review processes, which seek to increase defect containment to early design stages. Within these hardware detail design processes, defect containment models would also complement change propagation methods and metrics proposed by previous research (Giffen et al, 2009).

8. Conclusions and Implications for Management

This research investigates cross-functional hardware ECM capabilities within a contemporary firm to identify qualitative factors that contribute to the effective management technical design change, and explore the utility of data mining techniques in the context of specific ECM processes. At a macro-level, this thesis used a case study framework to identify common factors that can complicate the effective and efficient management of technical change. Qualitative empirical data from engineering stakeholder interviews provides rich contextual discussion of key interdependencies, and enables analysis of these common factors in the context of enterprise architecting principles. At a micro-level, this thesis "zooms" in to investigate the state of systematic change analysis through the lens of specific technical management processes. Quantitative empirical defect data is then mined from actual NPI projects to explore the creation of data-mining models and their utility to selected sub-processes. Furthermore, a holistic vision is generated, which has near-term implications for the specific enterprise, and highlights important interactions that generalize to contemporary industry.

While ECM is often regarded as collection of engineering support processes, insight derived from qualitative and quantitative study also highlights the interactive importance of configuration management with softer knowledge management policies. Findings highlight the need for socio-technical alignment between processes, PDM systems, and roles to enable more robust and data-driven techniques for systematic analysis of technical change. Other findings indicate that data-mining models and other multi-dimensional visualizations are viable techniques for reducing the knowledge overhead required to fully assess the impact of technical design changes. While this thesis is specific to the selected firm, these findings logically generalize to the contemporary defense industry. This thesis provides contextual understanding of technical management challenges and recommendations for managers, who are interested in synchronizing people and systems to promote ECM capabilities. Conclusions are summarized in the context of primary thesis questions, and general findings are summarized in Table 8.

	Detailed Questions	Findings	Executable Knowledge?	Questions Answered?
Qualitative Factors driving technical change	Are there instances of management or process policy resistance?	Endogenous Factors: Significant qualitative discussion corroborated some previous research findings and highlighted specific ECM enterprise needs: + Improved efficiency of PDM system use. Process & Stakeholder alignment to increase knowledge sharing component. + Systematic change identification capability and corresponding view element interdependencies. Exogenous Factors: Customer and supplier interdependencies with regard to CM and PDM interoperability, and opportunity for alignment of incentive policies.	YES: Identified areas of policy resistance, provided holistic vision and supporting strategic recommendations. However, transformation plan is required to implement policy actions.	Yes
Utility of Data Mining Models	Which data mining algorithms that provide accurate results and generalize to similar defect profiles?	Classification and Regression Trees (CART) provide simple, accurate, and easily interpretable models. Enhanced tree algorithms like CHAID provide visualizations. Alternatively, Boosted Trees, Random Forests, and NNs provide more robust computations and flexibility with modeling more complex dimensional space.	YES: Classification Models. NO: Prediction Models are conceptual only. However, near-term data initiatives will likely support proper function.	Yes
	Where could defect models provide the greatest benefit to line operations and program management during new product	Predictive Data Mining models would provide the greatest value to design teams. Classification models would also provide the following value: + Supporting CM and design teams with fast data mining of project defect attributes with easily interpretable visualizations.	YES: Automated classification of project -specific defect relationships would be useful.	Yes
	What aspects of existing enterprise element interactions detract from the utility of the concept?	New PDM System and management of Project CM data was only recently aligned to provide CN and Defect Data by individual CIs. Consequently, there is no completed project data, from which additional attributes can be drawn.	YES: Synchronization of data collection across PDM and QDM systems would be useful to specific CIs.	Yes

Table 8. Supporting Research Questions and Findings.

8.1 What common qualitative factors contribute to unintended hardware technical change activity in new product development enterprises?

This thesis identifies several common qualitative factors that contribute to technical change. The chapters provide a detailed discussion in regard to the following factors:

- *Inability to systematically survey technical change.* The enterprise is more reliant on tactical measures for resolving need for change and assessing their impact. Without clear process directives to collect causal relationships between change action, project leads have no ability to identify change propagation or derive CI-specific metrics for assessing future design change impact.

- *Application of Lessons Learned.* Technical change lessons are more likely to inform redesign or upgrade activity of the same product configuration that spawned the lesson. These lessons are generally not applicable to later unique configuration "branches" that are more likely on NPI projects.

- *Relative engineer defect inexperience may be a factor.* Though discussion was limited, there was some indication that experienced engineers, who are not aligned with product specific documentation requirements, often contribute to administrative design changes.

- *Efficient Access to Configuration Data.* The inability to efficiently access needed information can directly contribute to communication overhead, which reduces situational awareness to the impact of proposed changes. This discussion has implications for PDM process and system initiatives.

- *Software vs. Hardware Technical Change.* Less restrictive SW change control processes enables higher rejection rates of formally proposed changes; this offers analytical opportunity that is not often found in HW processes that are less tolerant of rejected formal changes and often fail to record informal HW CM activity.

- *Supplier and Customer Uncertainty.* Lack of alignment between internal and external stakeholders may introduce misunderstanding of both administrative documentation requirements and technical change impact analysis.

8.2 Can these qualitative insights be integrated with data mining models to develop leading tactical measures for helping to mitigate hardware technical change?

Yes. While this thesis is only a first effort for modeling project-defect relationships, classification models offer potential for generalizing configuration-specific relationships from often-unwieldy PDM systems. In particular, if models are tailored to user needs and integrated within QDM analysis, then engineering and IPT users will use them as decision aids.

While experimentation did not produce an accurate predictive model, this research assessed that proper relational data could be aggregated to support a future model implementation. However, more historical associations between configuration items must be aggregated within QDM systems to provide CI-specific defect profiles by *Commodity Type* and *Document Type* attributes. For example, such predictive models (either single or ensemble models) would provide design engineers a quick and automated method of predicting whether a design document is likely to present a costly three-stage-delay uncontained defect. If integrated with existing design review processes, data-driven models would provide an unbiased aid for IPT Leaders seeking to allocate scant resources to the most detrimental design defects.

8.3 Concepts for Leveraging Product Data Management (PDM) Processes and Systems

PDM concepts proposed by this research could bolster the hardware process and tool architecture within the selected firm. The first concept illustrates the use-case for aggregating CI-specific defect attributes in order to build automate predictive models. The second concept establishes associative attributes between change action documents to enable change propagation analysis and metrics.

While specific to one enterprise, sense-making from interviews and analysis of quantitative data provided insight that is likely to apply to other firms that use integrated product development processes.

8.4 Transforming the ECM Enterprise

Even within the same business unit, specific engineering functions and NPI projects may have unique configuration management processes that are tailored to their needs. In order to institute more effective surveillance of technical change as proposed by this research, engineering firms should carefully assess dominant view element interactions, stakeholder value exchange, cultural factors, and initiatives that reinforce positive feedback loops.

8.5 Potential branches to research theory

Evaluation of project defect containment in the context of coordination-type communication within design peer review teams

This research discussed but did not evaluate the wealth of knowledge from design peer review databases, which document the reviews of detail design components and assemblies prior to their released baseline. Further research may consider a single or cross-case study of detail design peer reviews to evaluate the significance of coordination-type communication to a projects final defect containment rate. For instance, a predictive model may be formulated using historical directional communication data inputs to a DSM (Morelli et al, 1995) or an ES-MDM (Bartolomei, 2012), and then compared to those projects defect containment rates. The accurate modeling of coordination-type communication and actual defect containment rates would provide a powerful tool for strengthening communication linkages within the organization. Alternatively, a similar analytical approach could be applied to coordination-type communication within CCB Teams. Another research branch could draw engineer survey data of PDM system queries.

Evaluation of ECM discipline across industries

While research findings may generalize to other hardware ECM enterprises, further research may investigate substantive differences across different commodity types or industries. Further research may employ s multiple case study method with cross-case analysis to evaluate impact of key factors to different disciplinary fields, in order to demonstrate broad applicability of findings.

Estimation of technical change cost impact

Defect cost was unavailable or non-existent for the majority of the defect records. This was due partly to the intricacies of data collection, and how change impact cost is tracked from different perspectives. Fundamentally, technical change impact analysis addresses the need to meet requirements, and the cost of taking such action within the context of the program budget; this perspective defines the lens through which change cost analysis is conducted. In general terms, change cost is defined as the resource cost associated with implementing an approved change into a TDP (and MDP, Testing, Manuals, etc.) and associated products as defined by the change action effectivity (i.e. which serial number will be affected by the change). Consequently, a bottoms-up approach to accounting for numerous cross-functional impacts from one individual change is sometimes impractical, especially when considering hundreds to thousands of successive changes. Detailed research into the process architecture that enables data-driven cost estimation would benefit the technical change impact analysis.

9. References

Bartolomei, J. Hastings, D., de Neufville, R., and Rhodes, D. Engineering Systems Multiple-Domain Matrix: An Organizing Framework for Modeling Large-Scale Complex Systems. Systems Engineering Vol. 15, No. 1, 2012.

Bertimas, D., O'Hair, A., Pulleyblank, W. The Analytics Edge. Preliminary Copy, Dynamic Ideas, 2012.

Boas, R. C. Commonality in Complex Product Families: Implications of Divergence and Lifecyle Offsets. MIT Engineering Systems Division, Doctoral Dissertation, September 2008.

Breiman, L. Statistical Modeling: The Two Cultures. Statistical Science, Vol. 16, No. 3 (Aug. 2001), 199-215.

Campo, M., and Frost, A. "Advancing Defect Containment to Quantitative Defect Management". CrossTalk: The Journal of Defense Software Engineering, December 2007, p. 25.

Danilovic, M. and Browning, T.R. "Managing Complex Product Development Projects with Design Structure Matrices and Domain Mapping Matrices." *International Journal of Project Management 25:* 300-314, 2007.

Defense Acquisition University. Systems Engineering Process. DAU ACQuipedia, web accessed on 2013. < https://dap.dau.mil/acquipedia>

de Neufville, R. and Field, F. "Thesis Definition and Preparation: Some General Guideline." Internal Publication, Revised September 2010. Engineering Systems Division, MIT.

Eckert, C., Clarkson, P., and Zanker, W. Change and Customization in Complex Engineering Domains. Research in Engineering Design, Vol. 15, 2004, 1–21.

Giffin, M., de Weck, O., Bounova, G., Keller, R., Eckert, C., and Clarkson, J. "Change Propagation Analysis in Complex Technical Systems." *Journal of Mechanical Design:* 131 (8), 2009.

Glaiel, F. Agile Project Dynamics: A Strategic Project Management Approach to the Study of Large-Scale Software Development Using System Dynamics. S.M. Thesis. MIT System Design and Management Program, 2012.

Hayes, R., Wheelwright, S., Clark, K. Dynamic Manufacturing: Creating the Learning Organization. The Free Press, New York, 1988.

Hines, E. K. Lifecycle Perspectives on Product Data Management. S.M. and S.M. Thesis. MIT Technology and Policy Program and Department of Aeronautics and Astronautics, September 2005.

Jarrett, T. A. W., Eckert, C. M., Caldwell, N. H. M., and Clarkson, P. J., 2011, 856 "Engineering Change: An Overview and Perspectives on Literature," Res. Eng. 857, Des., 22, pp. 103–124.

Jarrett, T., Eckart, C., and Clarkson, P. J. Pitfalls of Engineering Change: Change Practice during Complex Product Design. Advances in Design, Springer Series in Advanced Manufacturing (2006), 413-423.

Klein, J. True Change: How outsiders on the inside get things done in organizations. Jossey-Bass, 2004.

Knight, M. System Dynamics Homework #5: Brook's Law. MIT System Project Management (ESD.36) Course. October, 2011.

Krishnamurthy, K. and Law, K. H. Change management for collaborative engineering. Computing in Civil Engineering, Vol. 2 (1995), 1110-1117.

McConnell, S.. "Brooks' Law Repealed?" IEEE Software Magazine, November-December 1999.

Morelli, M., Eppinger, S., Gulati, R. Predicting Technical Communication in Product Development Organizations. The International Center for Research on the Management of Technology, Sloan School of Management, May 1995.

Nightingale, D., Rhodes, D. Architecting the Future Enterprise. Massachusetts Institute of Technology, ESD.38 Lecture 1, 21 February 2012.

Nightingale, D., Rhodes, D. "Enterprise Systems Architecting: Emerging Art and Science within Engineering Systems." MIT Engineering Systems Symposium, March 2004.

Nisbet, R., Elder, J., and Miner, G. Handbook of Statistical Analysis and Data Mining Applications. Academic Press, 2009.

Pasqual, M. Multilayer Network Modeling of Change Propagation for Engineering Change Management. S.M. and S.M. Thesis. MIT Technology and Policy Program and Department of Aeronautics and Astronautics, 2007.

Schmueli, G., Patel, N., Bruce, P. Data Mining for Business Intelligence. Second Edition.John Wiley & Sons, 2010.

Smith, R., Eppinger, S., and Gopal, A.. Testing an Engineering Design Iteration Model in an Experimental Setting. Working Paper #3386-92-MS, MIT Sloan School of Management, February 1992.

Smith, R., Eppinger, S. A Predictive Model of Sequential Iteration in Engineering Design. Management Science: 43 (8), August 2007.

Sosa, M.E., Eppinger, S.D., and Rowles, C.M. "Are Your Engineers Talking to One Another When They Should?" Harvard Business Review: 133-142, 2007.

StatSoft, Inc. (2011). STATISTICA (data analysis software system), version 10. www.statsoft.com.

Ulrich, K. and Eppinger, S. Product Design and Development. Fifth Edition. MacGraw-Hill. New York, NY, 2012.

Watt, F (2012). Engineering Documentation Control Handbook Configuration Management And Product Lifecycle Management. Fourth Edition, Swindon, U.K. british Informatics Society Limited, 2010.

Weill, P. Innovating with Information Systems: What Do the Most Agile Firms in the World Do? Sixth e-Business Conference PwC and IESE Barcelona, Spain, 27 March, 2007.

Yin, R. *Applications of case study research.* Beverly Hills, CA: Sage Publishing. 1993.

Yin, R. *Case Study Research: Design and Methods.* Thousand Oaks, CA: Sage Publications, 2003.

Appendix A: Dynamics of Introducing Inexperienced Staff

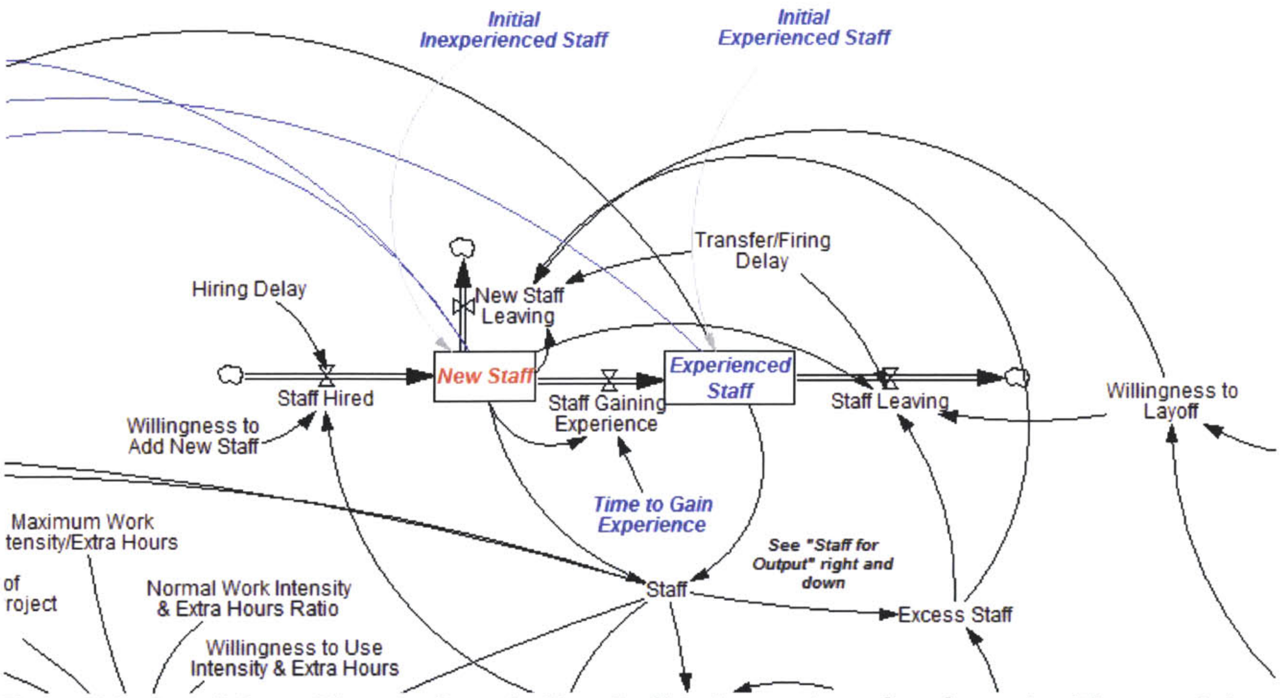


Figure 36. Casual Loop Diagram Part 1: New Staff is inexperienced and require Time to Gain Experience (Knight, 2011).

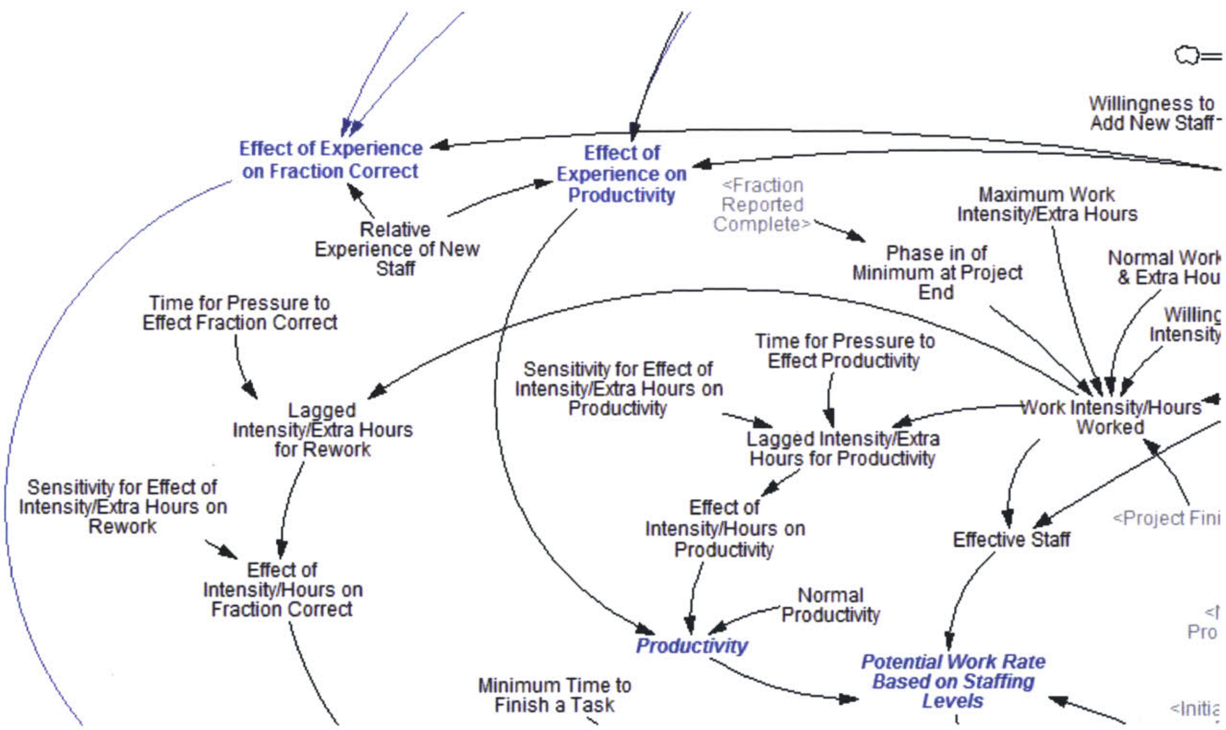


Figure 37. Casual Loop Diagram Part 2: Inexperience increases the Effect of Experience on Fraction (of work) Correct and reduces Productivity (Knight, 2011).

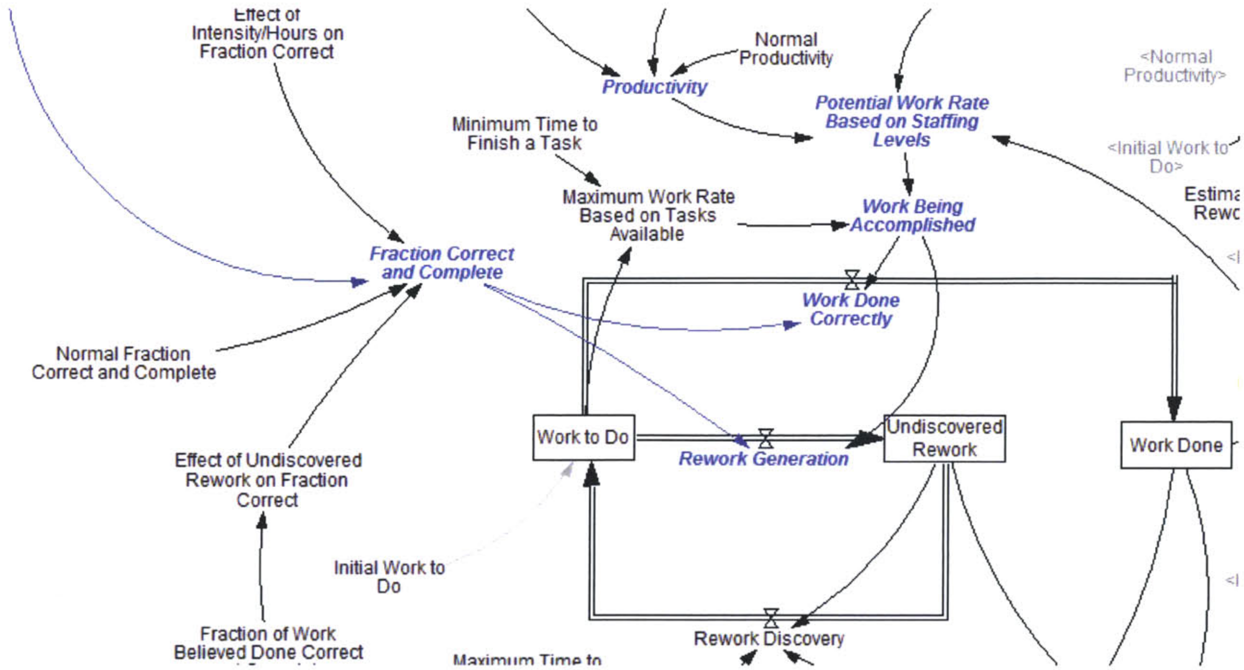


Figure 38. Casual Loop Diagram Part 3: Reduced Fraction (of Work) Correct & Complete and reduced Productivity drive Rework Generation (Knight, 2011).

Appendix B: Exploration of NPI Program Data

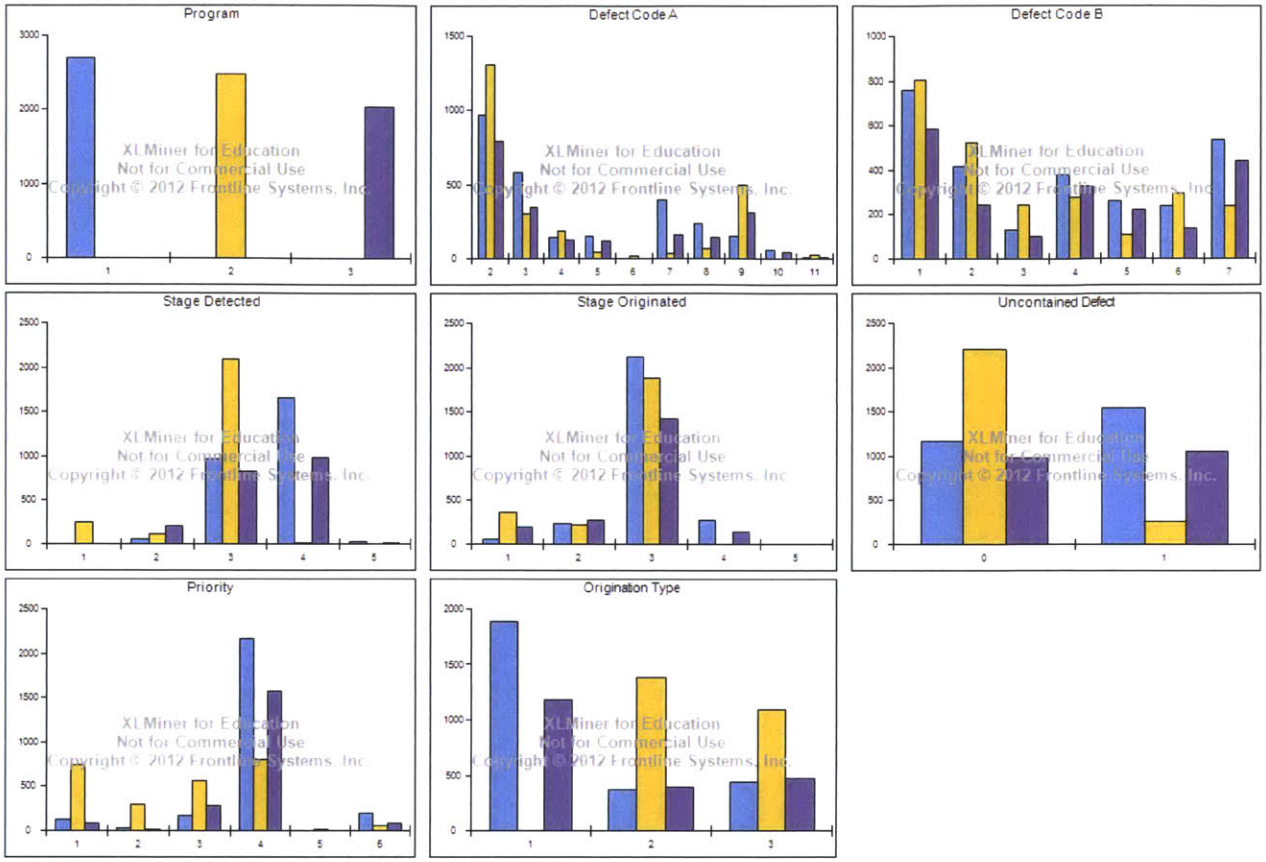


Figure 39. Defect Attribute Histograms for Programs #1, #2, #3.

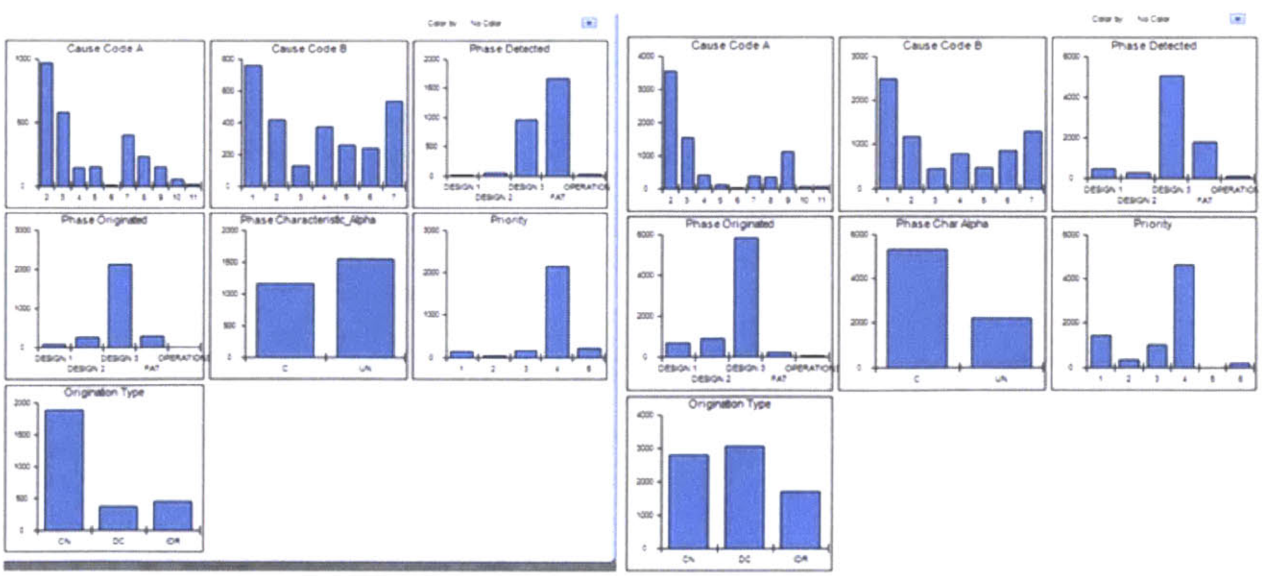


Figure 40. Box Whicker Plots of Defect Data by Cause Code for Programs #1 and #2.

Appendix C: Iteration #1 - CART Model Code

Note: Derived from Statistica PMML using Program #1 & #2 Data.


```
<?xml version="1.0" encoding="Windows-1252" ?>
<PMML version="2.0">
<Header copyright="STATISTICA Data Miner, Copyright (c) StatSoft, Inc.,
www.statsoft.com."/>
<DataDictionary numberOfFields="11">
	<DataField name="Uncontained Defect" optype="categorical">
		<Value value="0" NumericValue="0"/>
		<Value value="1" NumericValue="1"/>
	</DataField>
	<DataField name="Defect Code A_2" optype="categorical">
		<Value value="0"/>
		<Value value="1"/>
	</DataField>
	<DataField name="Defect Code A_3" optype="categorical">
		<Value value="0"/>
		<Value value="1"/>
	</DataField>
	<DataField name="Defect Code A_8" optype="categorical">
		<Value value="0"/>
		<Value value="1"/>
	</DataField>
	<DataField name="Defect Code A_9" optype="categorical">
		<Value value="0"/>
		<Value value="1"/>
	</DataField>
	<DataField name="Defect Code B_1" optype="categorical">
		<Value value="0"/>
		<Value value="1"/>
	</DataField>
	<DataField name="Defect Code B_2" optype="categorical">
		<Value value="0"/>
		<Value value="1"/>
	</DataField>
	<DataField name="Defect Code B_3" optype="categorical">
		<Value value="0"/>
		<Value value="1"/>
	</DataField>
	<DataField name="Defect Code B_5" optype="categorical">
		<Value value="0"/>
		<Value value="1"/>
	</DataField>
	<DataField name="Stage Detected_3" optype="categorical">
```

```
                <Value value="0"/>
                <Value value="1"/>
        </DataField>
        <DataField name="Stage Originated_3" optype="categorical">
                <Value value="0"/>
                <Value value="1"/>
        </DataField>
</DataDictionary>
<TreeModel
        functionName="classification"
        modelName="ITrees"
        splitCharacteristic="multiSplit">
<MiningSchema>
        <MiningField name="Uncontained Defect" usageType="predicted"/>
        <MiningField name="Defect Code A_2"/>
        <MiningField name="Defect Code A_3"/>
        <MiningField name="Defect Code A_8"/>
        <MiningField name="Defect Code A_9"/>
        <MiningField name="Defect Code B_1"/>
        <MiningField name="Defect Code B_2"/>
        <MiningField name="Defect Code B_3"/>
        <MiningField name="Defect Code B_5"/>
        <MiningField name="Stage Detected_3"/>
        <MiningField name="Stage Originated_3"/>
</MiningSchema>
<Node score="0">
<targetPrediction name="0" value="5.01121914734480e-001"/>
<targetPrediction name="1" value="4.988780852655520e-001"/>
        <TRUE/>
        <Node score="1">
<targetPrediction name="0" value="1.948640483383369e-001"/>
<targetPrediction name="1" value="8.051359516661631e-001"/>
                        <SimplePredicate     field="Stage     Detected_3"     operator="equal"
value="0"/>
                <Node score="1">
<targetPrediction name="0" value="0.00000000000000e+000"/>
<targetPrediction name="1" value="1.00000000000000e+000"/>
                                <SimplePredicate field="Stage Originated_3" operator="equal"
value="1"/>
                </Node>
                <Node score="0">
<targetPrediction name="0" value="7.049180327868885e-001"/>
<targetPrediction name="1" value="2.950819672131115e-001"/>
                                <SimplePredicate field="Stage Originated_3" operator="equal"
value="0"/>
                        <Node score="1">
```

```
&lt;targetPrediction name="0" value="3.77777777777778e-001"/&gt;
&lt;targetPrediction name="1" value="6.22222222222222e-001"/&gt;
                    <SimplePredicate field="Defect Code B_5"
operator="equal" value="1"/>
            &lt;/Node&gt;
            &lt;Node score="0"&gt;
&lt;targetPrediction name="0" value="7.50778816199377e-001"/&gt;
&lt;targetPrediction name="1" value="2.49221183800623e-001"/&gt;
                    <SimplePredicate field="Defect Code B_5"
operator="equal" value="0"/>
                &lt;Node score="0"&gt;
&lt;targetPrediction name="0" value="6.40883977900553e-001"/&gt;
&lt;targetPrediction name="1" value="3.59116022099448e-001"/&gt;
                        <SimplePredicate field="Defect Code A_9"
operator="equal" value="1"/>
                &lt;/Node&gt;
                &lt;Node score="0"&gt;
&lt;targetPrediction name="0" value="8.92857142857143e-001"/&gt;
&lt;targetPrediction name="1" value="1.07142857142857e-001"/&gt;
                        <SimplePredicate field="Defect Code A_9"
operator="equal" value="0"/>
                &lt;/Node&gt;
            &lt;/Node&gt;
        &lt;/Node&gt;
    &lt;/Node&gt;
    &lt;Node score="0"&gt;
&lt;targetPrediction name="0" value="8.01481481481481e-001"/&gt;
&lt;targetPrediction name="1" value="1.98518518518519e-001"/&gt;
            <SimplePredicate field="Stage Detected_3" operator="equal"
value="1"/>
        &lt;Node score="1"&gt;
&lt;targetPrediction name="0" value="0.00000000000000e+000"/&gt;
&lt;targetPrediction name="1" value="1.00000000000000e+000"/&gt;
                <SimplePredicate field="Stage Originated_3" operator="equal"
value="0"/>
        &lt;/Node&gt;
        &lt;Node score="0"&gt;
&lt;targetPrediction name="0" value="1.00000000000000e+000"/&gt;
&lt;targetPrediction name="1" value="0.00000000000000e+000"/&gt;
                <SimplePredicate field="Stage Originated_3" operator="equal"
value="1"/>
        &lt;/Node&gt;
    &lt;/Node&gt;
&lt;/Node&gt;
&lt;/TreeModel&gt;
&lt;/PMML&gt;
```

Appendix D: Iteration #1 - Evaluation Data of Boosted Tree Model

Note: PMML Code was too large to include for this model.

	Summary Frequency Table (Prediction) Table: Uncontained Defect(2) x 2-Boosted trees Prediction(2)			
	Uncontained Defect	2-Boosted trees Prediction 0	2-Boosted trees Prediction 1	Row Totals
Count	0	451	6	457
Column Percent		95.55%	1.40%	
Row Percent		98.69%	1.31%	
Total Percent		50.11%	0.67%	50.78%
Count	1	21	422	443
Column Percent		4.45%	98.60%	
Row Percent		4.74%	95.26%	
Total Percent		2.33%	46.89%	49.22%
Count	All Grps	472	428	900
Total Percent		52.44%	47.56%	

Table 9. Iteration #1 - Classification Confusion Matrix for Boosted Tree Model.

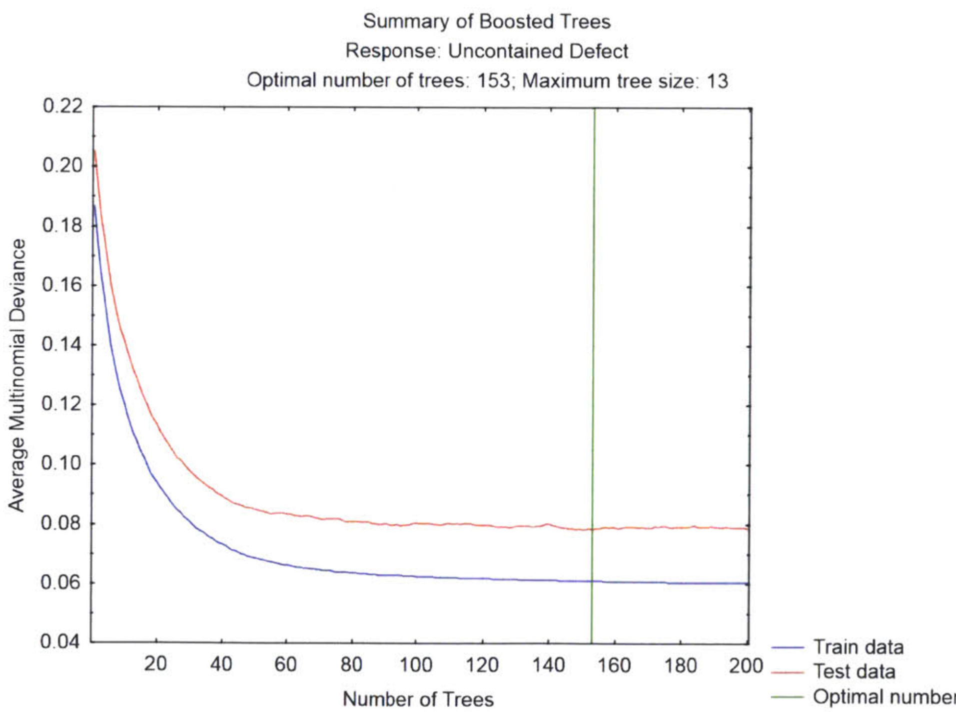


Figure 41. Iteration #1 - Optimal Number of Boosted Trees Graph.

Appendix E: Iteration #2 - CART Model Code

Note: Derived from Statistica PMML using Program #1 & #2 Data.


```
<?xml version="1.0" encoding="Windows-1252" ?>
<PMML version="2.0">
<Header copyright="STATISTICA Data Miner, Copyright (c) StatSoft, Inc., www.statsoft.com."/>
<DataDictionary numberOfFields="22">
	<DataField name="Uncontained Defect" optype="categorical">
		<Value value="0" NumericValue="0"/>
		<Value value="1" NumericValue="1"/>
	</DataField>
	<DataField name="Cause Code_201" optype="categorical">
		<Value value="0"/>
		<Value value="1"/>
	</DataField>
	<DataField name="Cause Code_202" optype="categorical">
		<Value value="0"/>
		<Value value="1"/>
	</DataField>
	<DataField name="Cause Code_203" optype="categorical">
		<Value value="0"/>
		<Value value="1"/>
	</DataField>
	<DataField name="Cause Code_206" optype="categorical">
		<Value value="0"/>
		<Value value="1"/>
	</DataField>
	<DataField name="Cause Code_207" optype="categorical">
		<Value value="0"/>
		<Value value="1"/>
	</DataField>
	<DataField name="Cause Code_301" optype="categorical">
		<Value value="0"/>
		<Value value="1"/>
	</DataField>
	<DataField name="Cause Code_302" optype="categorical">
		<Value value="0"/>
		<Value value="1"/>
	</DataField>
	<DataField name="Cause Code_307" optype="categorical">
		<Value value="0"/>
		<Value value="1"/>
	</DataField>
	<DataField name="Cause Code_401" optype="categorical">
```

```
        <Value value="0"/>
        <Value value="1"/>
</DataField>
<DataField name="Cause Code_406" optype="categorical">
        <Value value="0"/>
        <Value value="1"/>
</DataField>
<DataField name="Cause Code_502" optype="categorical">
        <Value value="0"/>
        <Value value="1"/>
</DataField>
<DataField name="Cause Code_504" optype="categorical">
        <Value value="0"/>
        <Value value="1"/>
</DataField>
<DataField name="Cause Code_701" optype="categorical">
        <Value value="0"/>
        <Value value="1"/>
</DataField>
<DataField name="Cause Code_702" optype="categorical">
        <Value value="0"/>
        <Value value="1"/>
</DataField>
<DataField name="Cause Code_703" optype="categorical">
        <Value value="0"/>
        <Value value="1"/>
</DataField>
<DataField name="Cause Code_704" optype="categorical">
        <Value value="0"/>
        <Value value="1"/>
</DataField>
<DataField name="Cause Code_A8" optype="categorical">
        <Value value="0"/>
        <Value value="1"/>
</DataField>
<DataField name="Cause Code_A9" optype="categorical">
        <Value value="0"/>
        <Value value="1"/>
</DataField>
<DataField name="Stage Originated_STAGE 2" optype="categorical">
        <Value value="0"/>
        <Value value="1"/>
</DataField>
<DataField name="Stage Originated_STAGE 3" optype="categorical">
        <Value value="0"/>
        <Value value="1"/>
```

```
        </DataField>
        <DataField name="Stage Originated_STAGE 4" optype="categorical">
                <Value value="0"/>
                <Value value="1"/>
        </DataField>
</DataDictionary>
<TreeModel
        functionName="classification"
        modelName="ITrees"
        splitCharacteristic="multiSplit">
<MiningSchema>
        <MiningField name="Uncontained Defect" usageType="predicted"/>
        <MiningField name="Cause Code_201"/>
        <MiningField name="Cause Code_202"/>
        <MiningField name="Cause Code_203"/>
        <MiningField name="Cause Code_206"/>
        <MiningField name="Cause Code_207"/>
        <MiningField name="Cause Code_301"/>
        <MiningField name="Cause Code_302"/>
        <MiningField name="Cause Code_307"/>
        <MiningField name="Cause Code_401"/>
        <MiningField name="Cause Code_406"/>
        <MiningField name="Cause Code_502"/>
        <MiningField name="Cause Code_504"/>
        <MiningField name="Cause Code_701"/>
        <MiningField name="Cause Code_702"/>
        <MiningField name="Cause Code_703"/>
        <MiningField name="Cause Code_704"/>
        <MiningField name="Cause Code_A8"/>
        <MiningField name="Cause Code_A9"/>
        <MiningField name="Stage Originated_STAGE 2"/>
        <MiningField name="Stage Originated_STAGE 3"/>
        <MiningField name="Stage Originated_STAGE 4"/>
</MiningSchema>
<Node score="0">
<targetPrediction name="0" value="2.96109839816934e-001"/>
<targetPrediction name="1" value="7.03890160183066e-001"/>
        <TRUE/>
        <Node score="1">
<targetPrediction name="0" value="2.43482538121003e-001"/>
<targetPrediction name="1" value="7.56517461878997e-001"/>
                        <SimplePredicate field="Stage Originated_STAGE 4" operator="equal"
value="0"/>
                <Node score="1">
<targetPrediction name="0" value="2.03187250996016e-001"/>
<targetPrediction name="1" value="7.96812749003984e-001"/>
```

<SimplePredicate field="Cause Code_207" operator="equal" value="0"/>
<Node score="1">
<targetPrediction name="0" value="1.818181818181 82e-002"/>
<targetPrediction name="1" value="9.818181818181 82e-001"/>
<SimplePredicate field="Cause Code_701" operator="equal" value="1"/>
</Node>
<Node score="1">
<targetPrediction name="0" value="2.22361809045226e-001"/>
<targetPrediction name="1" value="7.77638190954774e-001"/>
<SimplePredicate field="Cause Code_701" operator="equal" value="0"/>
<Node score="1">
<targetPrediction name="0" value="9.57446808510638e-002"/>
<targetPrediction name="1" value="9.04255319148936e-001"/>
<SimplePredicate field="Stage Originated_STAGE 3" operator="equal" value="0"/>
</Node>
<Node score="1">
<targetPrediction name="0" value="2.49618320610687e-001"/>
<targetPrediction name="1" value="7.50381679389313e-001"/>
<SimplePredicate field="Stage Originated_STAGE 3" operator="equal" value="1"/>
<Node score="1">
<targetPrediction name="0" value="1.26353790613718e-001"/>
<targetPrediction name="1" value="8.73646209386282e-001"/>
<SimplePredicate field="Cause Code_301" operator="equal" value="1"/>
</Node>
<Node score="1">
<targetPrediction name="0" value="2.82671829622459e-001"/>
<targetPrediction name="1" value="7.17328170377541e-001"/>
<SimplePredicate field="Cause Code_301" operator="equal" value="0"/>
<Node score="1">
<targetPrediction name="0" value="1.05263157894737e-001"/>
<targetPrediction name="1" value="8.94736842105263e-001"/>
<SimplePredicate field="Cause Code_307" operator="equal" value="1"/>
</Node>
<Node score="0">
<targetPrediction name="0" value="3.08888888888889e-001"/>
<targetPrediction name="1" value="6.91111111111111e-001"/>
<SimplePredicate field="Cause Code_307" operator="equal" value="0"/>

```
                                        </Node>
                                    </Node>
                                </Node>
                            </Node>
                        </Node>
                        <Node score="0">
<targetPrediction name="0" value="5.00000000000000e-001"/>
<targetPrediction name="1" value="5.00000000000000e-001"/>
                                <SimplePredicate   field="Cause   Code_207"   operator="equal"
value="1"/>
                        </Node>
                    </Node>
                    <Node score="0">
<targetPrediction name="0" value="1.00000000000000e+000"/>
<targetPrediction name="1" value="0.00000000000000e+000"/>
                            <SimplePredicate field="Stage Originated_STAGE 4" operator="equal"
value="1"/>
                    </Node>
</Node>
</TreeModel>
</PMML>
```

Appendix F: Supporting Computer Applications

XLMiner was selected among several available software applications for this thesis research due to the tool ease of use and sufficient range of analytical methods. XLMiner is a Microsoft Excel add-in application that allows users to import spreadsheet data, which was output from an enterprise PDM system for this research. The tool provides basic manipulation functions including sampling, data partitioning and scoring with the flexibility to modify those methods to suit the analysis of large or small datasets. XLMiner also allows basic visualization techniques through a *Chart Wizard* function that supports basic charting, scatter plots (and matrices), histograms, box plots, etc. The tool provides a comprehensive range of data mining algorithms to support statistical analysis and machine learning methods for data exploration, classification of categorical variables and prediction of numerical variables.

Statistica is a software application that provides a comprehensive suite of data analysis tools, which provide for statistical analysis, data visualization and data mining procedures (StatSoft, 2011). After identifying NPI projects and canvassing databases for defect data, this research used *Statistica* to conduct data exploration and preparation, as well as model building, evaluation and selection. *Statistica* provides an independent graphical user interface, which enables the import of data from various formats and export of models to enterprise business systems. In comparison to the simpler *XLMiner* add-in to MS Excel, *Statistica* provides several additional advantages including more advanced algorithms, guided data mining workflows, ensemble-building functions, robust evaluation tools, and graphics that enable expedited and thorough modeling architectures.

Vensim is a software application used to model system or policy interactions with system dynamics method. After identifying reference modes of key policy and process interactions, this research used *Vensim* to illustrate observed policy friction and proposed concepts with a simple *Causal Loop Diagram*. However, a fully developed system dynamics analysis is out of scope for this research.