1-1-2013

Framework For Quantifying And Tailoring Complexity And Risk To Manage Uncertainty In Developing Complex Products And Systems

Darrell Williams
Wayne State University,

FRAMEWORK FOR QUANTIFYING AND TAILORING COMPLEXITY AND RISK TO MANAGE UNCERTAINTY IN DEVELOPING COMPLEX PRODUCTS AND SYSTEMS

by

DARRELL D. WILLIAMS

DISSERTATION

Submitted to the Graduate School

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

2013

MAJOR: INDUSTRIAL ENGINEERING

Approved by:

____
Advisor Date

____

____

____

____

DEDICATION

To my lovely wife *Catherine* and my three children *Candice*, *Faith*, and *Lucas*—words cannot express my gratitude for your sacrifice, perseverance, and support.

I am so proud of you!

ACKNOWLEDGEMENTS

I would like to thank my primary advisor, Dr. Ratna Babu Chinnam for his time, mentoring, and painstaking editing of my manuscripts. I will always be grateful for your investment in my work and your desire for my success.

I would like to acknowledge my dissertation committee members Dr. Leslie Monplaisir, Dr. Toni Somers, Dr. Kyoung-Yun Kim, and Dr. Alper Murat for their feedback and support of my dissertation, course work, and preparation--all of which have contributed to my growth as a professional and a researcher.

I would like to thank the Industrial and Systems Engineering faculty for their ongoing support of the Global Executive Track (GET) program--it is a timely and ambitious endeavor, and I was privileged to be a beneficiary of your work.

I would like to thank my executive cohort team members for their friendship, support, and collaboration. The insights I have gleaned from your dialogue and critiques were as valuable as they were entertaining, and perhaps the single thing I will miss most after graduation.

I would like to acknowledge my parents Darrell and Sandy who instilled in me the discipline to continue pursuing my goals until they are accomplished in full--a trait I have endeavored to pass onto my own three children.

Thank you Maggie for the kindness you showed a young man you hardly knew 25 years ago by supporting him as he began this journey--I am proud you can see it brought to completion.

Finally, I am grateful for God's countless blessings which have allowed me to accomplish this lifelong goal (*Jeremiah 29:11).*

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: Overview

I. Introduction

In recent years there has been a renewed interest in product complexity due its negative impact on launch performance (Gokpinar, Hopp, 2010; Sosa, 2008; Sosa, Eppinger, 2004; Tani and Cimatti, 2008; Tatikonda and Rosenthal, 2000). Today's complex products are marked by increasingly sophisticated subsystems, greater functionality, and a higher degree of component interaction (Kim and Wilemon, 2003; MacCormack, Verganti, 2001; Mihm, Loch, 2003). A study by Meyers and Wilemon determined that underestimating complexity was the most common error repeated by new product development (*NPD*) teams (Canada, 2010; Gidado, 1996; Keizer, Vos, 2005; Kim and Wilemon, 2003; Smith, 1992). It was concluded that the companies that successfully manage complexity enjoy a competitive advantage (Browning and Eppinger, 2002; Kim and Wilemon, 2003).

The continued growth in technology has propelled the defense industry into one of the most challenging times in its history which is evidenced by the numerous cost and schedule overruns plaguing acquisitions[1] programs in recent years (Anonymous, 2010; Harned, 2003; Schwartz, 2010). Defense contractors remain under constant pressure to develop higher performing systems for less cost (Accountability Office, 2008; Anonymous, 2010; Defense, 2010; Engineers, 2010; Harned, 2003). Recent reports indicate the Department of Defense (DoD is actively seeking to cancel or significantly curtail acquisition programs that experience significant cost growth (Schwartz, 2010). Adding to this challenge is the acceleration of new technologies into these products (Kim and Wilemon, 2003) which require more learning and adaptation from

[1] Defense Acquisition is the process by which the US government acquires weapon systems. This includes the purchase, or procurement, of an item or service which encompasses the design, engineering, construction, testing, deployment, sustainment, and disposal of weapons or related items purchased from a contractor. (Schwartz, 2010)

the organization that are further driving integration effort and cost (Tatikonda and Rosenthal, 2000).

Large-scale military projects are defined as CoPS projects (Complex Products and Systems) based on their unique characteristics including: 1) high degree of customization, 2) limited volume, and 3) heavy focus on systems engineering and integration (Hobday, 1998). Other examples of CoPS projects include cell phone networks, industrial construction projects, and offshore oil rigs (Hobday, 1998),(Yeo and Yingtao, 2009).

PD organizations are often concerned with complexity due to its impact on risk (Kim and Wilemon, 2003; Tatikonda and Rosenthal, 2000). The defense industry relies heavily on systems engineering (SE) [2] processes to help manage complexity and risk (Group, 2010; Sargis Roussel and Deltour, 2012). In this research we present novel methods for improving complexity and risk management that are consistent with current systems engineering practices (Group, 2010). The methods are initiated from preliminary customer requirements in order to be available at the early phases of *resource planning* and *proposal development.* The models also allow for continual updates to be made as new information becomes available and improve their predictive power.

As complexity increases, the level of uncertainty in projects also increases (Kim and Wilemon, 2003; Tatikonda and Rosenthal, 2000). Applying formalized risk management processes helps to reduce this uncertainty (Browning, Deyst, 2002; Group, 2010; Institute, 2008). Literature has indicated that a lack of effective risk management will negatively impact project success (Institute, 2008). Unfortunately, there have been no studies to quantify this relationship

[2] SE is an interdisciplinary approach which encompasses both the technical management and coordination of processes across the technical team.

and determine *how much* risk management is needed to achieve PD success. In this dissertation we present methods for reducing and managing development risk.

II. Key Literature

Complexity[3] has meaning across many fields including engineering, finance, computer science, biology, etc. (Tani and Cimatti, 2008). However, despite its broad application there remains no universal definition of the concept (Browning and Eppinger, 2002; Langlois, 2002; Tani and Cimatti, 2008; Tatikonda and Rosenthal, 2000). This may be in large part because the elements that complexity studies have been applied to have been so diverse. Measures of complexity have included such elements as: (1) number of *components* and their (2) *interactions*, (3) number of component *types*, (4) degree of *predictability*, (5) overall *order* in the system, etc. (Tani and Cimatti, 2008). For this research our focus is on *product development (PD) complexity* which we define as a function of the absolute complexity of the product (technological) and the organization's ability to develop it (organizational) (Clift and Vandenbosch, 1999; Garcia and Calantone, 2002; Tornatzky, 1982). This definition is consistent with Kim and Wilemon's (2003) concept that complexity should encompass all the difficulties and uncertainties posed by the technology during the development—including a consideration of the organization's tasks and people (Kim and Wilemon, 2003). Figure 1-1 below highlights this relationship.

[3] Complexity comes from the Latin word complexus meaning twisted together.

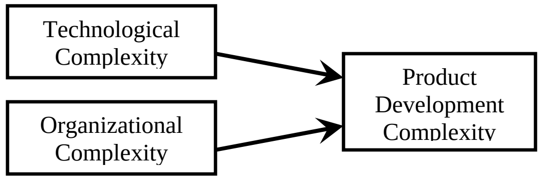

Figure 1-1: Product Development Complexity
(Product development complexity includes elements of both technological and organizational complexity)

Therefore, when using the term complexity in this research it refers to *product development* complexity and the elements that comprise it.

In planning for complexity Tatikonda (2000) suggests organizations assess the novelty of their projects and adjust them accordingly (and explicitly) in the '*front-end*' of development (Tatikonda and Rosenthal, 2000). Unfortunately, complexity remains difficult to quantify which is particularly true of CoPS projects which are highly unique (Gidado, 1996; Kim and Wilemon, 2003; Sosa, 2008), (Hobday, 1998; Hobday, 2000; Yeo and Yingtao, 2009).

Gokpinar, et al. (2010) proposed a method of quantifying complexity based on a product's subsystems and interactions. Here the product is represented as a network diagram based on the number of change notices[4] initiated (or received) by each subsystem group throughout development. The sum of all nodes (subsystems) and links (CN communications) in the system provides a measure of overall system complexity (Gokpinar, Hopp, 2010). Unfortunately Gokpinar's approach is only capable of calculating complexity after the design is complete which limits its application in resource planning. Yu, et al (2010) suggest resource planning models need improvement to understand the magnitude of resources needed to support all NPD projects (Yu, Figueiredo, 2010). This research extends the work of Gokpinar, et. al by establishing a method for the early quantification of product complexity in PD projects.

[4] A change notice is a formal document used to communicate product updates or design changes to departments within an organization.

As development complexity increases, the level of uncertainty and program risk also increases (Kim and Wilemon, 2003; Tatikonda and Rosenthal, 2000). To manage this uncertainty formal risk management processes are employed in many CoPS industries (Browning, Deyst, 2002; Institute, 2008). Improvements in risk management practices are needed in terms of resource planning, risk identification, and risk mitigation (Akintoye and MacLeod, 1997; Kutsch and Hall, 2010), (Chapman, 2001; Holzmann and Spiegler, 2011; Tchankova, 2002), (Kutsch and Hall, 2010; Mojtahedi, Mousavi, 2010). Identifying opportunities to reduce risk early on, and improve risk management processes will help in managing PD complexity and uncertainty.

Browning, et al. (2002) developed a method to quantify program risk based on performance requirements (Browning, Deyst, 2002). The model achieved this by summing together the risk assessments of several key requirements based on their projected likelihood and performance functions (as PDFs). (AT&T, 1993; Group, 2010). While the method is novel in providing a quantitative assessment of risk, it is limited to addressing performance risk singularly, with no concurrent assessment of schedule or cost risks. This research extends Browning's work by applying a similar method to all areas of program risk concurrently (including performance, cost, and schedule)[5]. The improved method has the benefit of being applied at an earlier point in development (concept selection) to aid in early risk planning.

Risk management practices continue to be a essential part of PD success for many organizations (Akintoye and MacLeod, 1997; Chapman, 2001; INST, 2002; Tchankova, 2002; Thompson and Perry, 1992). However, it remains unclear how much risk activity is necessary to ensure PD success (Kutsch and Hall, 2010). Using data from a recent development project this research addresses this question to provide guidance in early risk planning. This research

[5] Risk in this research is defined as the measure of future uncertainty associated with achieving program objectives for product performance, cost, or schedule (Simpleman, 2006)

extends the work of Kim and Wilemon (2003) by validating the predictive relationship between complexity and risk, to aid in early risk identification.

This dissertation is organized into five chapters including a high level introduction and literature review (Chapter 1), a detailed presentation of methods (Chapters 2-4), and consolidated results and conclusions (Chapter 5). The methods chapters (2-4) are prepared and presented as independent works that incorporate their own specific introduction and literature review sections. This format is intended to provide a more comprehensive presentation of each method section, as well as facilitate their individual publication into journals. For instances where different methods are grounded in common literature topics, some minor overlap may be found between literature review sections. However, care has been taken to minimize overlap as much as possible in the presentation of this dissertation.

III. Proposed Methodology

In this dissertation we present novel methods for assessing, quantifying, and coordinating complexity and risk in the early development of CoPS projects. The methods are targeted toward improving the accuracy of *budget allocations* and *organizational resource plans* which in turn will support successful execution.

The current state process for project planning in CoPS products is shown in Figure 1-2, beginning with receipt of customer requirements and the allocation (assignment) of those requirements to the responsible subsystems (Group, 2010). The completed requirement allocations are then used to establish the product architecture (PA) and facilitate final concept selection (AT&T, 1993; Group, 2010). After *concept selection* is complete, the *budget allocations* are established for each of the subsystem groups.

Throughout the project planning process information flows into the *organizational resource plan* to ensure staffing and resources are being allocated. The project planning process is iterative, with information flowing back to previous steps to ensure new details are being considered and adjustments being made.

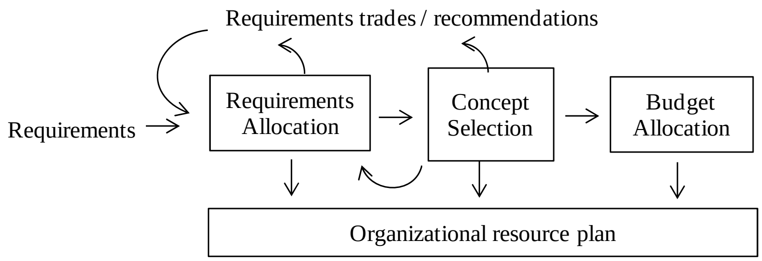

Figure 1-2: Current State Project Planning Process
(Current state process is initiated with receipt of customer requirements)

Because complexity and risk have a major impact on the expenditure of resources it is essential to include a thorough assessment of both in the early planning process. The methods described in the following three chapters support this goal.

In Chapter 2 a complexity assessment model is presented that translates customer requirements allocations into a complexity score during the concept development stage for use in early resource planning.

The complexity estimation method is validated using data from a recent defense industry project. The process for calculating complexity includes steps for the summing of requirements, assessment interaction strengths, determination of complexity weighting, and quantification of resources needed. The complexity weighting assessment is based on a construct derived from the literature that includes such variables as product novelty, organizational capability, design flexibility, and logistics challenges. The construct includes measures of both organizational and technological complexity to provide a robust measure of development difficulty.

Once implemented the complexity assessment model can be used to guide decisions for the concept selection, requirements allocations, and organizational resource planning (reference Figure 1-3). The complexity assessment also facilitates an analysis of the misalignment that may exist between the organization and product structures (termed the ‘coordination deficit’) (Gokpinar, Hopp, 2010; Sosa, Eppinger, 2004). This capability enables the organization to tailor product complexity to their resources (or vice-versa) before costly design investments are made--thereby avoiding the common issues of over commitment of development capacity for innovative products such as Yu, et al (2010) describes (Yu, Figueiredo, 2010). The method builds on the work of Gokpinar et. al (2010) by providing an early quantification of product complexity *before* concept selection, which facilitates an early assessment of coordination deficit and resource planning.

Chapter 3 builds on the work of Kim and Wilemon (2003) by exploring the relationship between complexity and risk. This is done by extending the complexity assessment method from the previous chapter to use in predicting the amount of *risk activity* needed to support program success. The analysis is performed through a correlation of complexity and risk data from the same CoPS project. For development groups found to be practicing minimal risk management activity, performance metrics are analyzed to determine if it resulted in negative project performance. Based on the findings a method for estimating the amount of risk activity needed is proposed, to improve the probability of launch success.

A method of quantifying risk management *effectiveness* is also presented for use in continuous improvement activities. Collectively the research in Chapter 3 provides insight for the improved planning, identification, and measuring of risk activity for complex projects.

Chapter 4 extends the work of the previous two chapters by presenting a method for generating early risk profiles of design concepts to assess PD risk. The method employs a technique for quantifying requirements risk which was developed by Browning, et al. (2002). While Browning's method is limited to addressing only performance or cost risk singularly, this research has extended his work to include a concurrent assessment of performance, cost, and schedule risk simultaneously—resulting in a more robust risk profile. The improved risk assessment is also initiated at an earlier point in the development process to support initial concept selection.

Literature indicates that design decisions affect the level of risk in a project (Kim and Wilemon, 2003; Browning, et al., 2002) and therefore provide an opportunity for early risk avoidance. This research explores that concept further by proposing that a common list of PD decisions (identified by Krishnan and Ulrich, 2001) be used to assess the level of risk in a design concept. The approach extends the use of Krishnan and Ulrich's work by employing it as a risk planning tool. Utilizing this framework enables the product team to find options for reducing design risk and tailor the design solution during concept development.

The risk profiling method is framed as a design trade decision using optimization. The goal is to select the design elements that minimize the probability of having below threshold performance (i.e. risk). The analysis is based on a CoPS project example.

Collectively, Chapter 4 addresses the areas of risk process management, risk taxonomies, early risk assessment, and technical decisions in order to quantify and tailor risk of design concepts.

The integration of the three chapters of research into the current-state planning process (from Figure 1-2) yields the improved planning process shown in Figure 1-3 (below). Boxes shown in gray shading represent process steps that are carry-over from the original process.

This revised process improves information flow between development steps and improves the alignment of the organizational resources to project(s)—thereby improving PD execution.

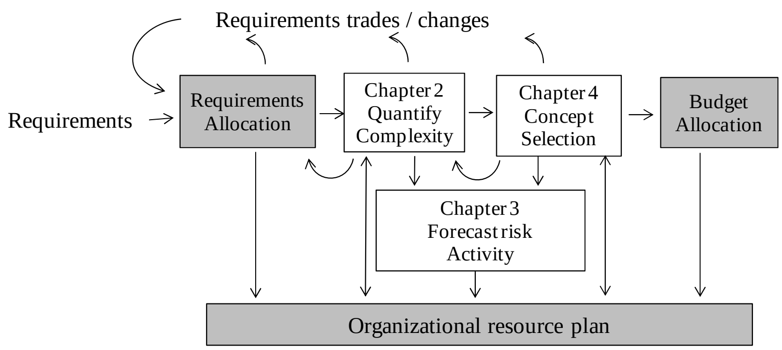

Figure 1-3: Improved Project Planning Process
(The revised process significantly increases information flow to support complexity and risk planning)

The *complexity quantification* method (Chapter 2) is performed immediately after *requirements allocation* to provide information directly to the *organizational resource plan*. A double sided arrow is shown between these blocks to represent instances when complexity is tailored to organizational resources.

In some cases *Concept selection* may require adjustments in requirements (in the form of trades or modifications) that need to feed back into the initial *requirements allocation* process.

Information from the complexity assessment is then used to forecast required risk activity as presented in Chapter 3. The complexity assessment also informs resource planning in the areas of staffing and budget requirements.

The final step before budget allocation is *Concept selection*. Using the method described in Chapter 4 of this research, a concept with design attributes that minimize risk is recommended.

When attempting to reduce risk through design trade-offs, a feedback loop has been provided which ties back to the *requirements allocation* process and triggers a re-assessment of complexity.

The final concept selection will affect the *Forecasted risk activity* needed, and the *organizational resource plan*. A double-sided arrow is shown between *Risk Concept selection* and *Organizational resource plan* to represent instances where project risk is being tailored to organizational capabilities.

The added information these methods provide in early planning stages will improve the alignment between product complexity and the organization. The improved alignment results in improved PD performance. A detailed review of each method is presented in the following sections.

Although the methods proposed in this research are designed to be initiated during the *concept* phase, they are expected to be re-iterated throughout development to help refine and improve the solution over time.

CHAPTER 2: Framework for Quantifying Complexity in Developing Complex Products and Systems

I. Introduction

As product development (*PD*) organizations struggle to keep pace with increasing technology demands, managing development complexity has become a major concern (Eppinger, Whitney, 1994; Kim and Wilemon, 2003; Morelli, Eppinger, 1995; Tani and Cimatti, 2008; Williams, 1999). The challenges often begin when forecasting development timing and cost without having sufficient understanding of the complexity up front (Kim and Wilemon, 2003; MacDonell, 2002). The issues continue through development as the design team attempts to manage competing requirements and understand the subsystem interactions (Mihm, Loch, 2003).

Complexity growth has been seen in nearly every industry (Gokpinar, Hopp, 2010; Sosa, Eppinger, 2004; Williams, 1999), and has been particularly aggressive for CoPS products which employ some of the most sophisticated technology available (Engineers, 2010). Development difficulty for CoPS products has been steadily increasing as reflected in their growing system cost (Jones, 2010). As development costs continue to rise (Emden, Calantone, 2006; Harned, 2003; Jayaram and Narasimhan, 2007; Jones, 2010), organizations have found the goals of cost control cannot be achieved *after* these hi-tech systems are fielded[6].

Studies indicate that 85% of lifecycle costs are locked-in after only 15% of detailed design is complete (Jones, 2010; Sosa, Eppinger, 2003). Hence, the goal of cost reduction for CoPS projects must be addressed in the earliest possible stages of design and development (Group, 2010). Improved estimates in the early concept stage will help the customer to establish an accurate budget, and allow the contractor to avoid costly overruns (Jones, 2010).

[6] A fielded system is one which has been produced, delivered, and is in use in by the end customer.

CoPS products require a higher level of coordination due to their component integration and design process iteration[7] (Gidado, 1996),(Sargis Roussel and Deltour, 2012),(Schmickl and Kieser, 2008). Many traditional PD processes that were designed to handle tasks as sequential, parallel events are quickly becoming inadequate (Williams, 1999; Zhang, Qiu, 2006). Furthermore, considering the magnitude that design information is increasing due to innovation and new technology, the task becomes even more challenging (Williams, 1999). Practitioners understand that managing overall performance requires a clear understanding of development interactions (Tani and Cimatti, 2008),(Kim and Wilemon, 2003),(Henderson and Clark, 1990), (Yassine, Joglekar, 2003). In fact, attempting to optimize subsystem performance independently can often lead to sub-optimal results for the entire system (Tani and Cimatti, 2008) (Kim and Wilemon, 2003) (Henderson and Clark, 1990). The result is that a lack of effective coordination is now being cited as one of the primary barriers to innovation by senior managers (Emden, Calantone, 2006; Gokpinar, Hopp, 2010).

Two major challenges impacting an organization's ability to manage PD complexity are: 1) the inability to accurately quantify development complexity up-front, and 2) inability to properly allocate resources within the organization to balance product requirements with available capacity (Gokpinar, Hopp, 2010; Yu, Figueiredo, 2010). This paper aims to squarely address these challenges by extending the work of Gokpinar, et al. (2010) and Yu (2010) with a method for the early quantification of complexity for early PD planning.

PD scholars have strived to develop universal, cross-industry methods for managing complexity. Unfortunately, unique challenges faced in each industry often warrant a specific approach (Hobday, 1998). In his research on CoPS projects, Hobday (1997) recommends avoiding generalizing research between mass-produced goods vs. defense industry products

[7] Design iteration is reworking or re-processing that is inherent in the development process (Yassine, et al., 2003)

because of the differences in their characteristics (Hobday, 1998). In keeping with Hobday's recommendation, the focus of this paper is to develop a methodology for managing complexity of CoPS projects specifically, although it may be adapted in the future to accommodate other types of PD projects. The proposed framework is validated on a new defense industry project which results in more accurate resource estimates and better understanding of development uncertainty up front.

The paper commences with a review of the existing literature, then describes the proposed conceptual framework and methodology for quantifying and managing PD complexity. A case study from the defense industry follows. Finally, we conclude with recommendations for practitioners and researchers.

II. Literature Review

A. Complexity

In recent years there has been a renewed interest in PD complexity due its negative impact on project performance in terms of: lead time, cost, assembly issues, and reliability (Gokpinar, Hopp, 2010; Sosa, 2008; Sosa, Eppinger, 2004; Tani and Cimatti, 2008; Tatikonda and Rosenthal, 2000). In order to determine the overall impact that complexity will have on PD performance, it is first necessary to develop an accurate predictor of complexity, and the research includes several measures.

In several cases throughout the literature we find that complexity includes a measure of both quantity of elements and the interaction of those elements (Gokpinar, Hopp, 2010; Langlois, 2002; Tani and Cimatti, 2008; Tatikonda and Rosenthal, 2000; Williams, 1999). Therefore, there is consensus that as the number of components and/or interactions increase in a system, complexity will also increase (Browning and Eppinger, 2002; Gokpinar, Hopp, 2010; Langlois, 2002; Tani and Cimatti, 2008).

In the work of Gokpinar, et. al, (2010) complexity is assessed as a function of *product architecture*. This is accomplished by creating a network for the product, and introducing a variable called "centrality" that serves as a proxy for complexity in the system (Gokpinar, Hopp, 2010). The *centrality* of each node in the network is then calculated by counting the number of links that originate and/or terminate at that node. This approach is intended to provide a direct correlation of the degree of integration that each node has in the system. Finally, the sum of all centralities for all the nodes in the system provides a measure of overall system complexity (Gokpinar, Hopp, 2010). It is clear that this methodology follows the theory of complexity being a measure of: (1) number of *components* and their (2) *interactions* as described above. Ongoing research has also indicated additional measures of complexity including: (3) number of component *types*, (4) degree of *predictability*, and (5) overall *order* in the system (Tani and Cimatti, 2008). Unfortunately, determining the relative impact of these five elements on the overall complexity has not yet been determined.

In the earlier work by Griffin (1997) he defines complexity in terms of the number of *functions* embodied in the product (Kim and Wilemon, 2003). This definition is different from the concept of complexity being described through *product architecture*, and instead analyzes it in terms of *performance* attributes. This approach has the benefit of being more applicable in areas outside of manufacturing, such as services industries or processes (Kim and Wilemon, 2003).

Yet another view of product complexity has emphasized it as a measure of *design effort* required by the organization to develop the product (Jacobs, 2007). Examples of such measures include: the degree of newness, novelty or customized components in a product (Garcia and Calantone, 2002; Griffin, 1997; Hobday, 1998; Tatikonda and Rosenthal, 2000). Project size has also been used as a measure of complexity including such variables as number of new technologies employed or percentage of development done in-house (Kim and Wilemon, 2003),(Tatikonda and Rosenthal, 2000). While each of these definitions can be seen as an attribute of the product, they relate directly to the concept of design *difficulty*.

Complexity has been shown to impact project success in many areas and continues to be relevant in PD literature (Sosa, Eppinger, 2004),(Williams, 1999),(Gokpinar, Hopp, 2010). Kim and Wilemon (2003) examine cases where product complexity impacts development projects with late delivery, over budget, under performance, etc. In assessing complexity's role in NPD performance, several sources are identified and categorized as either technological or organizational (Kim and Wilemon, 2003).

> <u>Technological complexity:</u> defined as the: (1) degree of required integration, (2) amount of (Garcia and Calantone, 2002), (Kim and Wilemon, 2003), (Sosa, 2000) innovation or novelty, (3) number of functions, or (4) type product architecture employed (i.e. integrated or modular) . While not all projects will contain metrics for each of these attributes, it is a common list of measures that are available in most. Effective measures of technological complexity are needed to align production processes and other organizational elements to the development tasks, in order to optimize efficiency (Tani and Cimatti, 2008).

> <u>Organizational complexity:</u> includes elements of people, processes, and tools used throughout development. Because product innovation drives multi-disciplinary activities, it is closely tied to the company structure and capabilities of its workers (Clift and Vandenbosch, 1999; Garcia and Calantone, 2002; Gupta, Raj, 1986). This is consistent with Hobday's concept of complexity as a function of the breadth of knowledge required for development. Increasingly complex projects require a wider range of skills and capabilities for development (Hobday, 1998). The more experienced an organization is with the technology area, the more efficient it will be at managing the project as a result of such factors as: formalization of company processes, effectiveness of the organizational structure, education of the workforce, and the operating culture (Clift and Vandenbosch, 1999; de Visser, de Weerd-Nederhof, 2009; Kim and Wilemon, 2003; Swink, Talluri, 2006).

Consolidating the elements of technological and organizational complexity reveals that development effort is a function of both the absolute complexity of the product, as well as the organization's ability to develop it (Clift and Vandenbosch, 1999; Garcia and Calantone, 2002; Tornatzky, 1982). Downs and Mohr, 1976 referred to these aspects as the *primary* and *secondary* attributes of innovation. The *primary* attributes are defined as those inherent in the innovation itself, whereas the *secondary* attributes include those of the organization, setting, and actors involved (Tornatzky, 1982). Mohr notes that a fatal flaw of much innovation research is that it does not include an assessment of both the primary and secondary elements. This definition is consistent with Kim and Wilemon's concept that complexity should encompass all the difficulties and uncertainties posed by the technology during the development—including a consideration of the organization's tasks and people (Kim and Wilemon, 2003).

Baccarini (1996) cites two basic dimensions of complexity including: (1) the number of varied and inter-related parts /components, and (2) the degree of complication or intricacy. While the first dimension can be clearly observed and quantified, the second dimension (degree of intricacy) is far more subjective because it involves a measure of the difficulty in understanding or working with the project. In concept, this dimension is closely related to the Downs and Mohr view that there is a *secondary* aspect to complexity which again involves the organization / context of the project (Tornatzky, 1982).

In order to capture and assess complexity within projects, Baccarini's suggests it be measured in terms of: (1) differentiation (i.e. number of inputs / outputs, separate tasks, and specialties involved) and (2) degree of integration between tasks, teams, technologies, etc. (Baccarini, 1996; Larson and Gobeli, 1989). This second element (degree of integration) represents *his* expression of the organizational / contextual aspect of complexity and is defined by the coordination, communication, and control of the organization (Baccarini, 1996). This theory again supports the distinction of complexity having both a product element, and an organizational (execution) element (Kim and Wilemon, 2003). In this research we maintain consistency with this approach and treat *complexity* as a function of both organizational and technological complexity, which we term *product development (PD) complexity*. This definition will allow us to more accurately assess the total impact of complexity on launch performance.

Table 2-1 (below) provides a summary of the key complexity constructs and their measures from the literature. As indicated, complexity constructs have been proposed which include elements of technological complexity, organizational complexity, or both.

Measures	Complexity Aspect(s)	Author
Cognitive capabilities required	Technological	Stata, (1989)
Number of parts in the product	Technological	Murmann (1994)
1) Number of different core technologies in the product 2) Diversity of core technologies in integration	Technological	Meyer and Utterback (1995)
1) Number of varied and inter-related parts 2) Degree of complication or intricacy.	Technological	Baccarini (1996)
Degree of product modification required: a) Simple Projects: reengineering projects and minor modification to existing projects b) Complex Projects: major modifications and projects with new-to-the-world products	Technological	Clift and Vandenbosch (1999)
1) Number of product components to specify /produce 2) Extent of the interactions to manage between these components (parts coupling) 3) Degree of product novelty	Technological	Novak and Eppinger (2001)
1) Number of customized components 2) Breadth of knowledge and skill involved in design	Technological	Hobday (1998)
Interdependencies of technologies	Technological	Tatikonda and Stock (2003)
Technology compatibility between elements	Technological	Kim (2003)
Uncertainty in achieving functional requirements	Technological	Suh (2005)
1) Number of physical modules 2) Degree of dependency	Technological	Kasaki & Heikkila (2002)
1) Degree of centrality / interaction across subsystems	Technological	Gokpinar, Hopp, and Iravani (2010)
Number of design decisions made	Organizational	Baldwin and Clark (2000)
Degree of acceleration / compressed steps in PD	Organizational	Cooper (1990)
Degree of understanding of technology involved (experience)	Organizational	McDonough (1993)
1) Number of alternatives & dimensions per alternative. 2) Extent to which dimensions are measurably the same 3) Order of information presentation 4) Familiarity with the kind of decision task 5) Incomplete info regarding dimensions of alternatives	Technological and Organizational	Hogarth (1980)
1) Number of different disciplines or departments involved in a project (nodes) 2) Intricacy of the design itself	Technological and Organizational	Larson and Gobeli (1989)
1) Number of functions designed into the product 2) Degree of coordination needed for development	Technological & Organizational	Griffin (1997)
1) Number of functional areas involved in the project 2) Intensity of the interaction between the elements from the different functional areas in the project 3) Difficulty of cooperation between the functional areas involved in the project	Technological and Organizational	Sbragia (2000)
The nature, quantity, and magnitude of organizational subtasks and subtask interactions posed by the project	Technological and Organizational	Tatikonda and Rosenthal (2000)
1) Structural uncertainty (elements and dependencies) 2) Uncertainty in goals & methods	Technological and Organizational	Williams (1999)

Table 2-1: Complexity Measures and Constructs
(Adapted from Kim and Wilemon (2003) and Jacobs (2007))

Despite the numerous constructs developed for complexity assessments, research has not yet identified a consistent method of scoring complexity at the earliest stages where the information is most valuable for resource planning. CoPS projects provide a significant challenge for such quantified complexity assessments due to their unique attributes. It is this challenge that motivates this research.

B. Management Tools

Over the last several decades, many techniques have been introduced to manage PD complexity (MacCormack, Verganti, 2001), although no single method has yet yielded perfect results (Sosa, Eppinger, 2004),(Gokpinar, Hopp, 2010),(Langlois, 2002),(Eppinger, Whitney, 1994),(Campagnolo and Camuffo, 2010),(Browning, Co, 2001). However, what has been confirmed is that traditional project management tools are unsuitable for today's complex products (Harned, 2003; Jones, 2010; Williams, 1999), hence, the ongoing research in this area.

Since the early 1960's researchers have developed and refined several tools to help manage complexity in product development. Tools such as Pert charts and Gantt charts were used for traditional project management, while more sophisticated tools like Design Structure Matrices (DSM), network models, and simulation programs were added to address more complicated applications (Browning, Co, 2001; Campagnolo and Camuffo, 2010; Eppinger, Whitney, 1994; Gokpinar, Hopp, 2010; Langlois, 2002; Sosa, Eppinger, 2004). Although these later tools generally found their way into use through defense and aerospace projects, they now permeate numerous PD industries that are a testament to the technology growth being experienced across these areas.

Although Pert charts and Gantt charts remain in wide use at PD firms because of their simplistic approach, they are generally ineffective at managing critical elements of the design iteration process (Mihm, Loch, 2003; Zhang, Qiu, 2006). Complex systems cannot be represented effectively with Pert and Gantt charts so DSM tools have been used to improve design architectures, organizational interactions, process flows, etc. (Browning, Co, 2001; Browning and Eppinger, 2002; Danilovic and Browning, 2007). This can be achieved by eliminating unnecessary coupling, consolidating multiple elements through modularity, or simply highlighting opportunities for concurrent engineering of non-dependent tasks (Danilovic and Browning, 2007),(Browning, Co, 2001).

The DSM was first introduced in the early 1960's by Steward. The DSM is a matrix that lists elements of a system (i.e. product, process, organization, etc.) along the top and left side of a matrix (reference Figure 2-1 below). When there is a relationship between any two of the elements it is indicated by placing a mark at the intersection of the row and column of elements (Browning, Co, 2001; Browning and Eppinger, 2002; Danilovic and Browning, 2007).

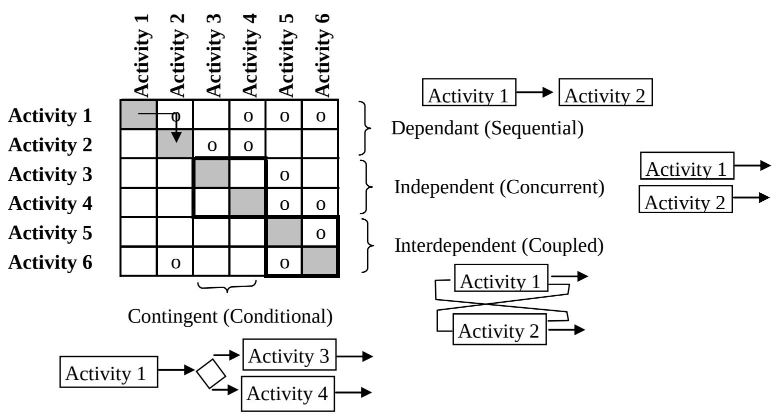

Figure 2-1: Design Structure Matrix (Browning, Co, 2001)
(DSMs can be used to represent product architectures, organizational interactions, or process flows)

DSM tools can be used to describe relationships in a physical architecture (using simple binary measures of a 1 or 0), indicate dependencies between variables, input and output flows, process steps, etc. (Browning, Co, 2001; Browning and Eppinger, 2002; Danilovic and Browning, 2007). Since the introduction of the DSM there has been substantial research into methods for quantifying interaction strengths between variables to better understand system behaviors (Zhang, Qiu, 2006). In the work of Eppinger and McCord (1993) the DSM tool was applied to the problem of team integration for a complex engine development project. The DSM was created by representing product development teams (PDTs) as system elements (rows and columns) of the matrix. The product development teams include membership from various specialties including PD, CAD, manufacturing, production control, finance etc. Dependencies were measured in terms of information flow between PDTs and reflected with scores of high, medium, or low based on the frequency of meetings. This scoring approach provided a quantitative indication of link / dependency significance, and was used to effectively regroup PDTs as part of the study. Future studies aimed to extend this research by using more quantitative models for assessing dependencies and PD complexity. Although the approach may be valid for quantifying the projected information flow between groups (i.e. high, medium, low), it is unknown if these measures provide sufficient accuracy of measurement metrics—suggesting a more quantitative method was needed.

Eppinger (1994) extended this work through the use of a numerical DSM to include explicit measures of dependency strength for prioritizing the partitioning and tearing of interactions in the DSM. Additionally it was suggested that these measures could represent a product of multiple measures such as information certainty and dependency significance (Eppinger, Whitney, 1994). With a measure of *certainty* included in the matrix interactions it introduces the element of risk into the analysis. Recall that risk is defined as the product of *likelihood* x *consequence*[8] (Browning, Deyst, 2002; Simpleman, 2006; Technology, 2002) which correlates closely with Eppinger's suggestion of certainty and significance. So while the research explicitly provides for quantification of dependencies / interactions, it implicitly introduces a key concept in attempting to include risk in the analysis (Eppinger, Whitney, 1994). It is suggested that collecting information for a numerical DSM can be done in number of ways including qualitative

[8] In risk management, consequences are scored in terms of their significance to project cost, performance, or timing.

assessment by SME (subject matter experts), testing, or formalized task-sensitivity charts (Eppinger, Whitney, 1994).

Eppinger (1994) provided for further DSM research related to developing work transformation models to explore methods to quantify task rework / iteration using probabilities (Smith and Morrow, 1999). Here dependencies are identified and scored between tasks, and corresponding rework functions including time and probability of occurrence are developed for each activity (Smith and Morrow, 1999). The approach allowed for a calculation of *total* development time, inclusive of the iteration/rework in design activities. A significant contribution was the ability to compare and predict lead times from various DSM strategies employing varying degrees of overlap in the process steps. Comparing directly the lead time required to complete activities in series –vs- parallel provided a means to directly measure the risk / return of coupling task—as is done in many industries today to accelerate product development lead time (Smith and Morrow, 1999).

Browning (2001) reviewed four key DSM applications (including: component base, team based, activity based, and parameter based) to demonstrate the maturity and usefulness of the tool in analyzing systems in terms of the product, process, and organizational structures. He included a review of both static models (representing subsystem components that all exist simultaneously), as well as time-based DSM models which reflect the flow through a process[9].

In the subsequent work of Browning and Danilovic (2006), an approach to compare DSMs across different project domains[10] was developed (termed Design MM). This technique is

---

[9] The methodology employed in this research will primarily be of the static model type, representing elements from an activities-based DSM by analyzing interactions / information sharing between PDT's in completing requirements coordination and engineering work.

[10] PD project domains may include the product system, process system, organization of people, system of tools and IT, and system of goals, requirements, and requirements. Changes in one domain will impact other domains throughout the PD process.

developed to improve coordination and decision-making by providing a means of analyzing decisions in different contexts, in order to understand how changes in one domain (i.e. product system) will create changes in another domain (i.e. process system) (Danilovic and Browning, 2007). Sosa (2008) builds off of this research by using DSMs to step through multiple “domains” in order to identify the design-team interactions that must occur to support the changes being made. He uses a similar strategy of linking matrices across domains when he introduces the affiliation matrix to correlate product architecture to organizational interactions[11] (Sosa, 2008).

Research on DSM has demonstrated it is a robust tool that can be easily applied to complex problems (Browning, Co, 2001; Browning and Eppinger, 2002; Danilovic and Browning, 2007; Jones, 2010). Unfortunately, a key limitation of DSM is its inability to handle dynamic simulation. As projects become too complex to model mathematically, it is often useful to create simulations to monitor their behavior to predict outcomes, and identify key variables (Smith, 1998). Techniques such as network modeling have grown in popularity in systems engineering circles because of their powerful applications, making them ideal to represent complex product architectures (Smith, 1998; Zhang, Sun, 2001).

C. Network Models and Product Architecture

In its most basic form, complexity can be represented by the number of elements (nodes) and their interactions (links) (Closs, Jacobs, 2008). Network diagrams provide an effective tool for managing complexity through component interactions (termed design propagation) throughout development (Ulrich, 1995). These tools help focus engineering on the coordination required between elements. In recent years network diagrams have proliferated across many areas of

[11] I will build on this research by starting with product requirements (rather than components) and perform a similar affiliation matrix translation, but include a measure of strength of interaction. The result will be a more timely (earlier) model for complexity assessment.

science and technology because of their effective representation of complex systems (Gokpinar, Hopp, 2010).

Network diagrams can be generated based on product architecture (PA) and are comprised of several elements (or nodes) that are linked to one another based on their relationships (Ulrich, 1995). Product architectures map product functions into physical elements, providing a direct reflection of complexity (Hobday, 1998; Novak and Eppinger, 2001). In each place a link or dependency is established, organizational interactions will be required, driving coordination effort and costs[12] (Novak and Eppinger, 2001).

Ulrich (1995) describes four product architecture topologies that are commonly employed including: integral, slot, sectional, and bus. He asserts that no single product architecture is optimal in all cases, and that organizations should be judicious to choose the best strategy for their needs, as each option will drive unique coordination / assembly requirements (Ulrich, 1995). On a continuum of simple to complex (i.e. modularity-to-integration respectively), product architectures consisting of higher degrees of integration will require higher levels of effort / coordination to manage the interfaces (Fixson, 2007; Novak and Eppinger, 2001; Schmickl and Kieser, 2008; Sosa, 2000). This situation becomes exacerbated where PA's are inconsistent with existing communication patterns or processes within the organization (Antonio, Richard, 2009; Gokpinar, Hopp, 2010; Ulrich, 1995). Because PA mappings can be established in multiple configurations based on the same set of requirements it provides an opportunity to tailor complexity to some degree—based on the level of modularity and component interactions desired (Gokpinar, Hopp, 2010; Ulrich, 1995; Wu, De Matta, 2009).

[12] This methodology proposed in this research is consistent with the concept that complexity can be measured based on the number of subsystem nodes and their dependencies.

When analyzing *product* network models it's important to recognize that the links represent static coupling of physical components, and do not reflect the degree of interaction among the engineering staff (through the organizational structure) (Browning, Co, 2001). Therefore, as design changes begin propagating through subsystems in the network (i.e. changes to subsystem A forces changes to subsystem B, forcing changes to subsystem C, etc.) the design activity should be identified and coordinated by project management. This concept was the foundation for the work of Clarkson et al. (2004), Jarratt et al. (2005), and Sosa (2008) in studying design propagation effects and predicting their communication patterns within organizations (Sosa, 2008). Establishing an appropriate product architecture model will ultimately determine the design team interactions needed (Ulrich, 1995).

In recent years research has explored product architecture's alignment to organizational structure and the resulting impact on launching complex products (Antonio, Richard, 2009; Gokpinar, Hopp, 2010; Shane and Ulrich, 2004; Sosa, Eppinger, 2004). It has been demonstrated that organizational structure itself is established and evolves through the architecture of the products (Henderson and Clark, 1990; Shane and Ulrich, 2004). Employing this approach ensures dedicated subsystem teams are established to address each element (subsystem) of the product. Although the concept of *product structure* influencing *organizational structure* is not new it has been gaining attention in recent years (Shane and Ulrich, 2004). In the research by Sosa (2004) the alignment between design interfaces and team interactions was studied on a large-scale air craft project to determine the degree of consistency. An alignment matrix was generated by overlaying the design interface (product) matrix with a team interaction (organization) matrix. The results revealed that the majority of interactions (over 90%) showed alignment, particularly among elements which were understood by the team as having strong interactions (Sosa, Eppinger, 2004).

In the subsequent work of Gokpinar, et al. (2010), the authors introduced the term *coordination deficit*[13]) to attempt to quantify the *alignment* between organization structure and product architecture and determine its impact on launch success. They concluded that inconsistencies between these hierarchies can cause deteriorating project performance (Gokpinar, Hopp, 2010; Sosa, Eppinger, 2004). Although the research was insightful, the calculation method was of limited use for concept development because it was generated from projects which were already fully designed using Engineering Change Orders (ECNs)[14].

The growing challenge for PD teams is to manage the coordination between functional groups as the system interactions become more pronounced (Kim and Wilemon, 2003; Schmickl and Kieser, 2008). Research suggests it is beneficial to align the organization structure to the product architecture when developing complex products (Gokpinar, Hopp, 2010).

Each product architecture contains some level of interaction that is not "seen," meaning interactions are still occurring at levels below what the product architecture hierarchy reflects (Sosa, Eppinger, 2003; Ulrich, 1995). Therefore, in establishing the product architecture, the goal is to group components in such a way as to maximize the interaction between their internal elements, and minimize the links (or coupling) required to other (external) elements (Campagnolo and Camuffo, 2010; Gokpinar, Hopp, 2010; Ulrich, 1995). Sosa, et. al (2003) builds off of this research by considering the level of modularity –vs- integration in a product architecture and how it affects PD performance (Sosa, Eppinger, 2003; Sosa, Eppinger, 2004).

[13] Coordination deficit is a metric introduced by Gokpinar et al. (2010), defined as the mismatch between organizational structure and product architecture. Coordination deficit was found to be positively associated with quality problems.

[14] It is here that the research will be extended by proposing a method for calculating coordination deficit before the design concept is complete.

D. Modularity

Modular systems are defined as those containing few physically connected or interacting elements (Fixson, 2007),(Mikkola, 2006),(Sosa, Eppinger, 2003). They are the opposite of *integrated* systems that contain many design interfaces across systems elements, forming a functionally distributed model (Mikkola, 2006), (Henderson and Clark, 1990). The concept behind modularity is to “break-up complex systems into discrete pieces that can then communicate with one another through standardized interfaces within a standardized structure” (Langlois, 2002). Although the application of modularity to organizational structure is somewhat new, the theory itself has been around since the early 1960’s, in product design, and before that in the social sciences (Simon, 1962; Alexander, 1964) (Langlois, 2002).

Due to the nature of systems hierarchies, modularity and integration occurs on multiple levels of a system simultaneously (Sosa, Eppinger, 2003) (e.g. systems are comprised of subsystems, subsystems are comprised of subassemblies, subassemblies are comprised of components, etc.). As such it is possible for highly modular systems to contain very integrated subsystem elements (Sosa, Eppinger, 2003). Sosa, et al (2003) suggested coordination across modular systems requires more management effort to ensure the required interactions occur (Sosa, Eppinger, 2003). Therefore, in order to manage the product development effort efficiently, it is necessary to establish the most appropriate modularity for the system (Fixson, 2007; Henderson and Clark, 1990; Mikkola, 2006).

Sosa, et al (2003) make a distinction between establishing an architecture at the *product* level vs. the *functional* level. Often a single system *function* will require input from several *product* elements so the mapping between these architectures is not one-to-one. Therefore the product and functional architectures will be distinctly different, even though they represent the same total elements of the system. This has made establishing architectures a key challenge for PD firms (Sosa, Eppinger, 2003). Recall that modularity in product design will also impact organizational structure so it is essential to address this need (Langlois, 2002).

Where functional architectures are needed which do *not* align to the existing organizational structure, adjustments should be made to align the working teams to the product requirements (Gokpinar, Hopp, 2010). Consider the example of innovative technologies that drive new organizational groups or reporting relationships to address the specialized team interactions (Sosa, Eppinger, 2003; Sosa, Eppinger, 2004). For this reason the requirements allocation process should be an integral part of the organizational planning and design.

Requirements allocation begins with the decomposition of system requirements into functional areas in order to create a preliminary architecture (AT&T, 1993; Group, 2010). An understanding of requirements priorities is needed to guide the system designers to ensure the most essential capabilities are maintained by each configuration (Karlsson, 1996). A common approach is to prioritize based on the importance of the function to overall system performance (Firesmith, 2004). In DoD projects requirements are often grouped using a three-tier rating scale including:

- Tier 1: Requirements deemed “essential” to system performance. These represent the highest priority and are non-tradeable, allowing zero flexibility in achieving the threshold performance levels.
- Tier 2: Requirements with limited flexibility in threshold performance, and may be traded-off (i.e. not met) in order to meet higher Tier 1 priority goals when necessary.
- Tier 3: Requirements with the most flexibility. Defined as tradeable against Tier 1, Tier 2, and other Tier 3 requirements in order to optimize the overall system performance.

As requirements are allocated to subsystem teams the principles of modularity should be employed to ensure node grouping & interfaces maximize the interaction occurring within the functional discipline, while minimizing the formal interaction required across other functional disciplines (Sosa, Eppinger, 2003). This will also serve to isolate subsystems for ease of redesign if necessary (Hölttä and Otto, 2005). An understanding of the existing organizational structure and division of labor is essential for this step so as not to introduce unnecessary coordination deficiencies (Gokpinar, Hopp, 2010).

While optimizing the system modularity it is important avoid consolidating groups (nodes) to the point of overburdening a single subsystem team (or node) with excessive *internal* interactions (Newman, 2006). Taken to the extreme this would resemble one single node for the entire system, with all elements contained within. Clearly this would be ineffective with no formal communication structure of any kind (Newman, 2006). The challenge is to find the proper balance to minimize complexity and maximize operational effectiveness.

A simple illustration of improved modularity and decreased complexity is shown in Figure 2-2 below. In the initial product structure note there is substantial interaction occurring across the low voltage and high voltage nodes (or subsystems). Consolidating these nodes into one single group called 'electrical' subsystem results in the more simplified formal structure shown to its right.

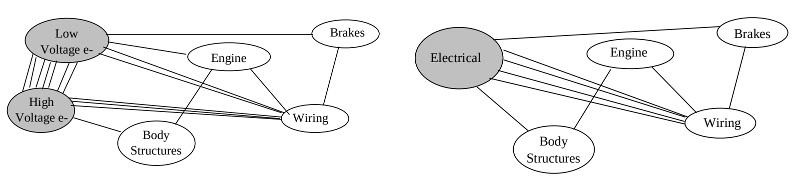

Figure 2-2: Modularity Improvement
(Reflects modularity improvement and decreased complexity based on consolidated nodes)

While the simplified structure may not eliminate the *need* for interactions to occur entirely, it will reduce the amount of interaction needed *between* the consolidated elements, as well as across the external elements (Langlois, 2002; Newman, 2006). In the simplified (modularized) structure, a single point of contact could be used to coordinate the interaction from both areas, thereby improving efficiency. For internal communications (i.e. between the high and low voltage areas), less interaction *effort* would be needed due to less formality of communication, co-location of staff & functions, commonality in skills set, etc. (Gomes and Joglekar, 2008) This supports the findings by Gomes, 2008 which found cross-element communications required more effort to manage than inter-element communications.

E. Novelty

Novelty is a common variable used in complexity studies (Eppinger, 2001). Although the defense industry recognizes Technology Readiness Levels (TRLs) as a measure of novelty, in practice it provides limited differentiation among competing subsystems or projects. Because most program requirements are requested to be at a common TRL level to reduce development uncertainty (i.e. level 7 or above)[15] the metric becomes less of a discriminator. Furthermore, requirements at the same TRL level do not necessarily possess equivalent design / integration complexity.

Novelty is commonly associated by researchers as a key contributor to project uncertainty and risk (Hobday, 1998; Novak and Eppinger, 2001). However, it's important to note that the mere addition of a new process or new technology into a system does not necessarily result in greater PD complexity (Kim and Wilemon, 2003). This is only the case when it contributes to a lack of understanding during the development process (McDonough, 1993). This subtlety is likely the reason the correlation between product newness and complexity has not always been consistent

[15] A TRL assessment provides a measure of a technology's maturity level in order to indicate the development risk associated with it. TRL level 7 indicated the technology has been field tested at the proto-type level. This is often the minimum anticipated TRL level for pre-production defense contracts.

between studies. For example McQuiston's (1989) research which found the correlation between newness and complexity significant (r = 0.463), while Griffin (1997) concluded the opposite with a correlation of r = -0.06). Adding to this confusion is the fact that novelty / newness can be measured in a number of different ways including: unique capability, design approach, components material / technology, or integration of elements (Kim and Wilemon, 2003; Novak and Eppinger, 2001).

Novelty is understood to be measured on a continuum and is a *relative* term. As a measure it is influenced significantly by the experience of the development organization or engineering team. As such, a technology which is "novel" for one organization may be more common for another, demonstrating that novelty *as an attribute* does not reside solely in the product itself, but also as a function of the organization's experience. This may lend insight into why previous studies have reported conflicting results in terms of correlation of novelty and complexity (Kim and Wilemon, 2003). They may have been measuring different aspects of the same variable. Therefore, the concept of novelty residing both within the product and within the organization appears valid.

In this research we capture this dual-aspect of novelty by first assessing the *intrinsic* (internal) element of the technology via the 'requirements burden' analysis, and next, quantifying the *organizational* (or external) novelty of the technology via the 'difficulty multiplier.' The multiplier quantifies the experience of the team in both the industry, and the technology being developed. For state-of-the art technologies which are being managed by teams with very little experience, the difficulty multiplier will be at an extreme—indicating a maximum coordination effort is needed (Hobday, 1998). Conversely, if a more experienced team can be assigned to develop the novel technology, a significant reduction in coordination burden can be realized, resulting in less effort / budget needed by the program. Such assessments can be done when the quantitative complexity model is established.

F. Project Size

Project size has been considered an element of complexity in several studies (MacDonell, 2002), although it has not been a universally applied metric because of its inconsistent application (Kim and Wilemon, 2003). For example, project size has been represented in several ways including the number of components, functions, or technologies integrated into a product (Kim and Wilemon, 2003), it is not always the case that these elements adequately describe the concept. Tatikonda and Rosenthal (2000) suggest that project size only captures a part of complexity, arguing that small-sized projects can have highly integrated (complex) designs while large projects can have highly modular (simple) designs (Tatikonda and Rosenthal, 2000). Baccarini (1996) suggests that project size is distinctly different than project complexity (Baccarini, 1996). For the purpose of this research, project size as an 'absolute' value will not influence complexity scores. Instead, complexity assessments will be performed at the subsystem levels, and calculations will be normalized within each project and compared as ratios across projects. Normalizing complexity scores within each project will ensure consistency between subsystems since: (1) the methodology is based on a physical *counting and scoring* of requirements and (2) requirements can often be specified at varying degrees of abstraction between projects. This means that the level of detail specified in the requirements for project A, may not be consistent with the level of detail specified for project B (Sharman and Yassine, 2004; Ulrich, 1995). Also, as different projects are specified by different customers it drives additional variation in the level of requirements abstraction (between projects).

As an example of varying abstraction, consider the case of a common flashlight being specified in a request for quote (RFQ) by a DoD customer. Such a device may be described by one hundred separate requirements comprising five unique subsystems to ensure the device can be used in the various operating environments. In contrast, a complex laser light projector may be specified in an RFQ by only twenty key performance requirements based on how it will be utilized. It's important to note that in defining project requirements, the customer will only specify requirements to the degree necessary to describe minimum system performance, and no more (Group, 2010). Specifications beyond this point are viewed as unnecessary because they add cost and constraints to the system. In practice, it is better to allow the design engineer to determine the constraints of his/her particular design (Group, 2010)

The example demonstrates that assessing the absolute complexity score of two separate projects for comparison may be misleading based on a *requirements-counting* method. However, using the counting method to comparing the *relative* complexity of subsystem *within* the same project will be accurate if the level of requirements abstraction[16] remained consistent. So while requirements abstraction may not be consistent *between* projects, it remains consistent *within* projects, making *relative* comparisons of subsystem complexity projects valid (Sharman and Yassine, 2004). Furthermore, comparing subsystem complexity scores versus historical / actual development costs (over-time) can provide a means of determining an absolute complexity—a technique that has been used extensively in software development estimates (MacDonell, 2002).

[16] Requirements abstraction refers to the level of hierarchy that requirements have been established.

G. Coupling

Coupling of functions is a key attribute of complexity which significantly impacts coordination effort. This is the case for both the initial product design, as well as subsequent changes resulting in design iterations of the product (Novak and Eppinger, 2001). The technical communications required to manage this effort is impacted by organizational elements such as distance between groups, formal organization structures, information lead time, communication media, etc. (Sherman, 2004; Sosa, Eppinger, 2004). Products which are modularized and managed across multiple organizations represent extreme cases of coupling complexity—this is common for systems integrators working on major CoPS projects such as in the defense or communications industries (Hobday, 1998; Hobday, 2000). For design teams already operating under timing constraints, additional coordination burdens such as distance / location can significantly increase labor burdens (Sherman, 2004).

CoPS projects often require highly tailored components and/or unique materials for their applications which may become long-lead items and create immediate risk to the program (Hobday, 1998; Hobday, 2000). In these instances, additional coordination effort is needed to ensure: appropriate suppliers have been selected, timely communication with the manufacturers is occurring, and the materials are being fabricated on schedule (as they are often on the critical path with no room for delay) (Harland, Brenchley, 2003).

In the following sections we present a method for assessing complexity which draws from the literature in terms of complexity measures and applicable tools. In particular, the framework uses a DSM and network model to build on the work of Sosa (2004) and Gokpinar et al. (2010). The method is used to provide an early assessment of product complexity based on customer requirements. It will also extend the work of Yu et al. (2010) by providing a detailed model for resource allocation to avoid the issues of over commitment of development capacity for innovative products.

III. Conceptual Framework and Methodology

The methodology developed is designed to be consistent with industry practices to facilitate easy adoption. It integrates traditional systems engineering (SE) steps with established complexity management tools, while leveraging data derived from complexity measures based on current literature. The process steps include: 1) Requirements Analysis and Allocation, 2) Scoring Requirements for Complexity and Effort, 3) Differentiating Difficulty of Requirements, and 4) Translating Requirements into a Development Network Model for Further Analysis.

Based on the initial results, adjustments can be made to the requirements (or resources) and re-calculated in order to tailor the product complexity to the organization's capabilities (or vice-versa). A simplified example is presented in this section to describe the methodology. In the subsequent Case Study and Analysis section we present actual data from a CoPS project.

Step 1: Requirements Analysis and Allocation

The first logical step in assessing and managing the complexity is to understand each requirement ($r_i, i \in [1, \dots, n]$) and allocate the same to the subsystem teams ($s_j, j \in [1, \dots, m]$), leading to the requirements allocation matrix (RAM) illustrated in Figure 2-3. The process of reviewing all requirements and allocating them to responsible subsystem teams is a common practice in systems engineering to initiate concept development. It is here that our method begins, and extends this practice by recognizing subsystem teams as either "primary" requirements owners (P), or as "secondary" (S) owner(s). The subsystem team that is directly responsible for the performance requirement is assigned primary ownership, while subsystem teams that provide significant input (or are closely coupled) to the requirement being measured are assigned secondary ownership. In allocating the requirements, the principles of modularity should be employed in order to maximize the interaction occurring within the subsystem area, while minimizing the formal interaction required *across* subsystem disciplines (Hölttä and Otto, 2005).

Subsystems

Requirements	1	2	3	4	5	6	7	8	9	…	*m*
1	S								P		
2		S	S		S				P		
3		P									
4		S			P		S		S		
5				P	S		S		S		
6			P						S		
7						P					
8								P	S		
…											
n	S			S	S						P

P: Primary design responsibility
S: Secondary responsibility; Provide key input to primary owner

Figure 2-3: Requirements Allocation Matrix ($\boldsymbol{RAM}$)
(Captures allocation of requirements to subsystem teams as primary and secondary owners)

Step 2: Scoring Requirements for Complexity and Effort

Once the requirements are allocated, the proposed methodology seeks to translate the allocations into effort scores for the efficient apportionment of product development resources. While there are several ways to accomplish this, we recommend the process of weighting each 'P' and 'S' in the requirements allocation matrix with a number corresponding to the subsystem's level of aggregate effort for the particular requirement (e.g., participation and coordination in preliminary design, detailed design, implementation, integration and testing, and supporting system verification and validation efforts related to the requirement). Here, P denotes the effort needed by the primary owner to coordinate the requirement, and S denotes the effort needed by the secondary owner(s) to coordinate the requirement. It is commonly the case that the primary owners will contribute a larger percentage of their time to the managing of the requirement than the secondary owners. The P and S scores represent the ratio of these efforts based on responsibilities.

For ease of translating the requirements allocations into effort scores we suggest initially assigning a '1' to all secondary requirement owners, indicating they will contribute the lowest overall effort toward the requirement fulfillment. The primary requirements owner(s) ('P') should then be assigned a comparative value reflecting their relative level of effort. In our example, each of the secondary requirements owners ('S') is assigned a score of '1' and each "primary owner" ('P') is assigned a value of '3', indicating the primary owners are estimated to spend roughly 3 times the amount of time / effort as the secondary owners in coordinating each requirement. The initial weighting of '3' is selected based on input from subject matter experts (SMEs) with past program experience. The resulting Requirements Effort Matrix (REM) is illustrated in Figure 2-4. For simplicity, during the initial problem set-up, one might choose to assign the same numeric value to each of the secondary owners and one common value to each of the primary owners that can later be adjusted to 'fine-tune' the model if necessary.

Subsystems

Requirements	1	2	3	4	5	6	7	8	9	…	*m*
1	1								3		
2		1	1		1				3		
3		3									
4		1			3		1		1		
5				3	1		1		1		
6			3						1		
7						3					
8								3	1		
…											
n	1			1	1						3

P: Primary design responsibility (3)
S: Secondary responsibility; Provide key input to primary owner (1)

Figure 2-4: Requirements Effort Matrix (REM)
(P and S tasks are assigned effort scales of '3–High Effort' and '1–Nominal Effort', respectively; P:S Effort Ratio = 3:1)

The final scaling of primary to secondary efforts (i.e. their ratio) should be determined based on performance history. It is understood that the P:S ratio may vary by subsystem group and will need to be calibrated (tailored) accordingly based on past results. Tracking the P:S ratio over time can also provide a measure of coordination efficiency between subsystem/integration groups. For example, if historical data indicates the ratio increased from 1:3 to 1:5 over time, this would suggest that coordination has become more efficient, as less relative effort is required by the secondary owner(s). This is a particularly useful metric to consider after an organization has implemented changes such as modified reporting structures, employee training, co-location, new hiring, etc.

Step 3: Differentiating Difficulty of Requirements

In practice it is understood that some requirements will be relatively more challenging than others to achieve. Therefore, the model allows for differences in the level of effort needed *between* individual requirements to be captured via an effort difficulty multiplier, $M_i, i \in [1, \ldots, n]$, that is applied to the requirements effort matrix.

The construct shown in Figure 2-5 was established for the multiplier calculations based a review of the literature, the applicability of variables, and the availability of the measurement data.[17]

[17] The construct used to calculate difficulty multipliers will vary based on product and industry. For this research the complexity variables were selected based on their: (1) perceived relevance to development difficulty, (2) their ease of quantification, and (3) their availability to the team.

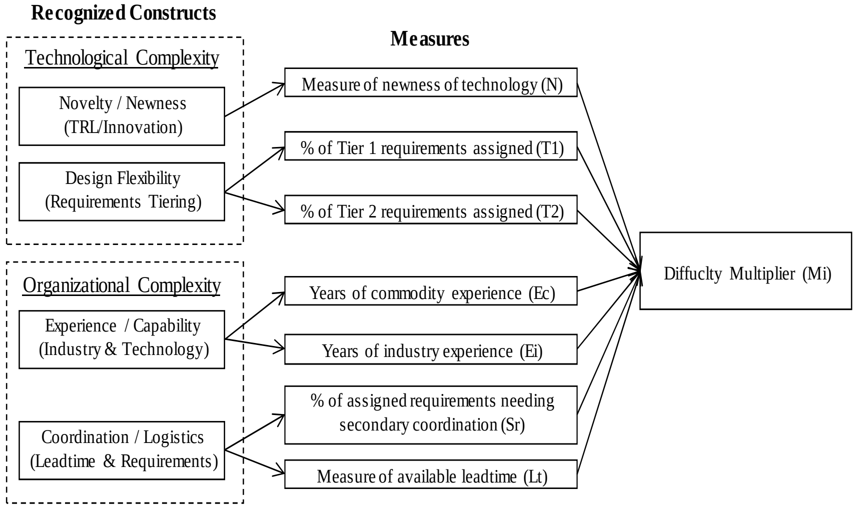

Figure 2-5: Difficulty Multiplier Construct[18]
(Model includes elements of both technological and organizational complexity)

The difficulty multiplier incorporates measures of both technological and organizational complexity utilizing the following four variables:

1. Novelty - defined as the anticipated design challenge of the team's requirement(s)
2. Flexibility – defined as the allowable tolerance in the design requirements, and measured as a function of the requirements tier data[19]
3. Capability - defined as the team's proficiency in both the commodity and the industry, scored in terms of years of experience
4. Coordination - defined as the efficiency of the organization's structure and processes, quantified as a function of secondary requirements responsibility and available slack-time design development

The determination of the difficulty multiplier construct will heavily depend on the nature of the project/industry and availability of data. For effectiveness, one should derive an appropriate parameterized function for estimating M_i based on a review of previous complexity variables /

[18] For reference, a table listing the complete scoring criteria for each variable has been included in Appendix A.
[19] In DoD projects, requirements priorities/flexibility is commonly defined through tiering such that: Tier 1 requirements have highest priority and by definition are non-tradeable, allowing for zero tolerance / flexibility in achieving the specified threshold performance levels, Tier 2 requirements have limited flexibility and may at times be traded-off (i.e. not met) in order to achieve the higher priority Tier 1 goals. Tier 3 requirements typically provide the most flexibility and tradeable against Tier 1, Tier 2, and other Tier 3 requirements to optimize the overall system performance.

constructs, and interviews with SMEs. The case study section provides the details of how the multipliers were derived for this research.

In the illustrative example below (Figure 2-6), multipliers M_i have been assigned to each individual requirement and reflect a range of 1.0 – 1.5. This indicates that the most difficult requirement (M_i=1.5) will take 50% more effort (time/resources) than the simplest requirement(s) (M_i=1.0).

Requirements \ Subsystems	1	2	3	4	5	6	7	8	9	….	*m*	Difficulty Multiplier
1	1								3			1
2		1	1		1				3			1.2
3		3										1
4		1			3		1		1			1.3
5				3	1		1		1			1
6			3						1			1
7						3						1
8								3	1			1.5
…												
n	1			1	1						3	1

P: Primary design responsibility (3)
S: Secondary responsibility; Provide key input to primary owner (1)

Figure 2-6: Difficulty Multipliers Appended to Requirements Effort Matrix
(Difficulty multipliers will be derived based on appropriate complexity variables -- tailored by product and industry)

The range of values for the difficulty multipliers will generally be comparable to the extreme values of development lead-times that can be experienced. In the automotive industry where product development lead times may range from 18 to 48 months based on vehicle complexity, this would translate to a lead time ratio of 18:48 (shortest to longest) or 1:2.7. Conversely, modern military development phases may range from 18 – 27 months, which roughly correlates to a ratio of 1:1.5. For illustrative purposes multipliers of 1.0 – 1.5 are assigned in Figure 2-6.

The process of calculating the overall requirements burden can now be done based on the requirements effort matrix (REM) and the difficulty multipliers ($M_i, i \in [1, \dots, n]$) as follows:

$$B_{r_i} = \sum_{j=1}^{m} REM_{i,j}.M_i$$

$$B_{s_j} = \sum_{i=1}^{n} REM_{i,j}.M_i$$

$$B = \sum_{j=1}^{m}\sum_{i=1}^{n} REM_{i,j}.M_i$$

where, B_{r_i} denotes the overall effort burden of requirement (r_i), B_{s_j} denotes the overall effort burden for subsystem/integration team (s_j), and B the overall effort burden across all requirements and subsystem/integration teams.

The total requirements effort needed by the subsystem groups, B, has been termed "*Requirements Burden*" and provides a measure of full PD complexity. The proposed method extends the work of Gokpinar and Hopp (2010) by providing an *early* assessment of complexity to be used for early planning and risk assessment. The methodology also extends the work of Yu et al. (2010) to provide a more detailed resource allocation prediction model for use in capacity planning. The effort scoring method is a function of subsystem interactions and complexity measures, and therefore is consistent with the literature on complexity (Gokpinar, Hopp, 2010; Langlois, 2002; Tani and Cimatti, 2008; Tatikonda and Rosenthal, 2000; Williams, 1999).

In the illustrative example above, the difficulty multipliers (M_i) are derived for each of the 'n' requirements individually, enabling the PD organization to clearly quantify the impact of individual requirements changes. Unfortunately, for very large projects employing hundreds or even thousands of requirements, this level of analysis may not be practical. In such cases, it is more efficient to derive multipliers for *groups* of related requirements or entire subsystems based on the full set of requirements they have been allocated. This alternative approach employs a *single* difficulty multiplier for the entire *column* (subsystem/integration team), rather than singularized requirements (for each *row*). The process for calculating the overall subsystem burdens is then:

$$B_{r_i} = \sum_{j=1}^{m} REM_{i,j}.M_j$$

$$B_{s_j} = \sum_{i=1}^{n} REM_{i,j}.M_j$$

$$B = \sum_{j=1}^{m}\sum_{i=1}^{n} REM_{i,j}.M_j$$

where $M_j, j \in [1, \ldots, m]$ now denotes the difficulty multipliers for each subsystem/integration team, and take the same general range of 1.0–1.5. Subsystems with the highest projected "requirements burden(s)," will reflect the highest levels of design uncertainty and projected risk. These scores also provide a good predictor of development effort (i.e. complexity) and cost allocation.

Step 4: Translation of Requirements Effort into a Development Network Model

The final step in the process involves translating the requirements effort matrix (weighted by the difficulty multipliers) into a Design Structure Matrix (DSM) and equivalent project network model. In the DSM both the columns and rows represent subsystem teams. The total effort required by each subsystem group to address their *primary-owned* requirements are shown along the diagonal of the DSM—this can be thought of as effort led by the subsystem group. The.sum of all secondary effort (support) needed between subsystem teams are reflected in the numbers above and below the diagonal. Note there is no distinction made between values placed above vs. below the diagonal such as the case in process DSMs. Instead, all values in the matrix

(Figure 2-8) represent effort provided *'from'* each subsystem listed in the column, *'to'* each subsystem listed in the row.

To demonstrate the process in Figure 2-7 one of the requirements (denoted *'n'*) is selected from the REM and stepped through its translation to the DSM. Here requirement '*n*' has been assigned to subsystem *m* as primary (indicated by the '3' in column '*m*'). Subsystems A, D, and E are assigned as secondary for the same requirement (as indicated by the '1' in the corresponding columns).

This information is transferred into the DSM as '3' units along the diagonal (intersection *m:m*) to account for *m*'s primary responsibility, and '1' unit of effort at the intersections of *m*:A, *m*:D, and *m*:E, indicating the secondary input being provided *'to'* Subsystem *m*, *'from'* subsystems A, D, and E.[20]

Subsystems

Requirements	A	B	C	D	E	F	G	H	I	….	*m*
1	1								3		
2		1.2	1.2		1.2				3.6		
3		3									
4		1.3			3.9		1.3		1.3		
5				3	1		1		1		
6			3						1		
7						3					
8								4.5	1.5		
….											
n	1			1	1						3

FROM **Subsystems**

Subsystems **TO**	A	B	C	D	E	F	G	H	I	….	*m*	Node
A	0											0
B		3										3
C			3						2			3
D				3	1		1		1			3
E		1.3			3.9		1.3		1.3			3.9
F						3						3
G							0					
H								4.5	1.5			4.5
I	1	1.2	1.2		1.2				6.6			6.6
….												
m	1			1	1						3	3

Figure 2-7: Translation to DSM
(All Primary input is captured along the diagonal of the DSM, and Secondary input is shown above and below the diagonal)

The process is repeated for each requirement in the allocation matrix until all numbers are summed into their respective cells in the DSM. The completed DSM is shown at the right in Figure 2-7.

[20] The cells along the diagonal are termed *node values* and reflect the magnitude of *primary* requirements that have been allocated to each subsystem. The numbers above and below the diagonal are termed *link values* and represent the coordination effort needed between all the subsystem areas—also termed secondary ownership.

Mathematically, the translation of the REM to the DSM matrix can be carried out as follows:

initialize: $DSM = zeros(n, n)$

$$for\ \ i = 1{:}n$$

$s_i = \text{argmax}_j\left(REM_{i,j}.M_i\right)$

$for\ \ j = 1{:}m$

$DSM(s_i, j) = DSM(s_i, j) + \left(REM_{i,j}.M_i\right)$

end

end

Here, s_i denotes the index of the primary subsystem for the requirement. The DSM is then translated directly into a project network diagram by creating nodes for each subsystem, and adding weighted arrows indicating the magnitude and direction of information that must flow from one group to the other (see Figure 2-8). Note that the size of each subsystem node corresponds to the magnitude of "primary" requirements burden the subsystem team has been assigned—this can be seen along the diagonal of the matrix

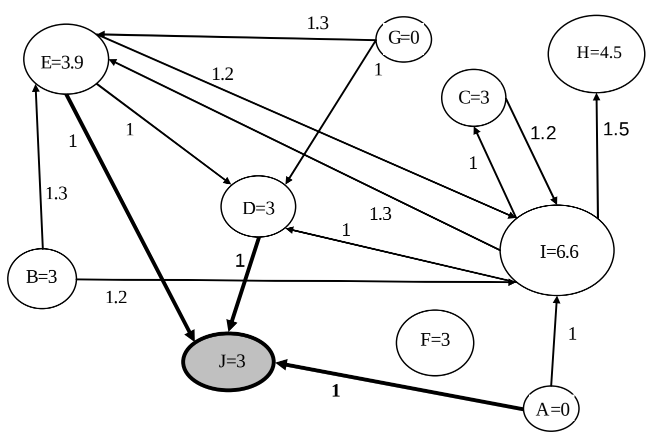

Figure 2-8: Project Network Diagram

(Primary input is reflected in the node values, and secondary input is reflected as link weights)

The project network diagram now represents total complexity based on initial customer requirements. In this form it is now possible to derive the *coordination deficit* that exists between the project's functional architecture and the company's organizational structure as presented by Gokpinar (2010). The benefit over this approach is that the product network diagram is based on customer requirements and established in advance of formal design work being completed.

To calculate the coordination deficit for the project, an organizational network diagram is also required. In our example an organizational network model is not derived as the literature provides many suitable methods. Sosa, et al. (2004) generated an organizational network model by identifying all groups responsible for product development, and interviewing key members of each team to determine the intensity of interactions between groups (Sosa, Eppinger, 2004). Gokpinar et al. (2010) generated an organizational network model by summing engineering change orders (ECO's) initiated and received by each subsystem group to determine the interaction strengths. This method only required existing information to be consolidated, as ECO data was already being captured (Gokpinar, Hopp, 2010). In recent years more sophisticated methods for generating organizational network models have been used through the use of social networking software, e-mail tracking, analyzing proximity of working groups, etc. (Sosa, Eppinger, 2004),(Gokpinar, Hopp, 2010),(Doreian and Stokman, 1997). Any of the above methods can be used to generate an organization network model for comparison to the development requirements driven project network diagram.

In addition to supporting early coordination deficit analysis, the complexity model we derive can also be used to guide resource allocations. This is done by using the effort ratios of each subsystem group (as calculated by the 'requirements burden') to allocate the available budget / resources for the program. This approach will enable staffing levels to be consistently applied to each area, using an objective method. Figure 2-9 below summarizes the steps of this process. Based on the DSM, the requirements burden has been calculated for each subsystem, with a total burden of 47 for the project. Assuming a target development cost of $2.5M, the estimated budget / staffing for each subsystem team can be derived as shown in Figure 2-9 below.

Subsystems

Requirements	A	B	C	D	E	F	G	H	I	….	*m*	Multiplier (Mi)
1	1	0	0	0	0	0	0	0	3		0	1
2	0	1	1	0	1	0	0	0	3		0	1.2
3	0	3	0	0	0	0	0	0	0		0	1
4	0	1	0	0	3	0	1	0	1		0	1.3
5	0	0	0	3	1	0	1	0	1		0	1
6	0	0	3	0	0	0	0	0	1		0	1
7	0	0	0	0	0	3	0	0	0		0	1
8	0	0	0	0	0	0	0	3	1		0	1.5
….												
n	1	0	0	1	1	0	0	0	0		3	1

Subsystems

Requ	A	B	C	D	E	F	G	H	I	….	*m*	Total
1	1								3			4
2		1.2	1.2		1.2				3.6			7.2
3		3										3
4		1.3			3.9		1.3		1.3			7.8
5				3	1		1		1			6
6			3						1			4
7						3						3
8								4.5	1.5			6
….												
n	1			1	1						3	6
Requirements Burden	2	5.5	4.2	4	7.1	3	2.3	4.5	11.4		3	47

	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**….**	***m***	Total
Requirements Burden	2	5.5	4.2	4	7.1	3	2.3	4.5	11.4		3	47
% of Total Burden	4%	12%	9%	9%	15%	6%	5%	10%	24%		6%	100%
Estimated Budget	106,383	292,553	223,404	212,766	377,660	159,574	122,340	239,362	606,383		159,574	$2,500,000
Estimated Hours (@ $50 / hr)	2,128	5,851	4,468	4,255	7,553	3,191	2,447	4,787	12,128		3,191	50,000
Engineers Needed (@ 2,000 hrs / yr)	1.1	2.9	2.2	2.1	3.8	1.6	1.2	2.4	6.1		1.6	25.0
Engineers at 85% Utilization	1.3	3.4	2.6	2.5	4.4	1.9	1.4	2.8	7.1		1.9	29.4

Engineering hourly rate ($): $50

Figure 2-9: Staffing Projections Based on Complexity
(Initial budget allocations established from complexity assessment)

Comparing these projected staffing needs to the current organizational resources (across each subsystem group) will provide a staffing plan that is based on projected PD complexity.

IV. Case Study Analysis

To demonstrate the utility of the proposed methodology, the results of its application are reported on a recent DoD project (military vehicle). To maintain confidentiality, the name of the project and organization remain undisclosed.

Existing Process:

As is typical of most DoD projects, the PD activity begins with a well-defined set of customer requirements. The project included over 1,350 singularized requirements which were assigned to ten separate engineering subsystems (column one, Table 2-2 below). Based on the organization's current process, the percentage of labor hours for each subsystem team were estimated as shown in column 2, Table 2-2 below (planned budget). Column 3 lists the actual labor hours spent throughout development (based on the original budget).

Subsystem Team	Planned Labor Hours	Actual Labor Hours	Planning Error (% Deviation)
Body	21%	24%	-15%
Telemetry	23%	22%	3%
Auxiliaries	2%	1%	45%
Electrical	19%	15%	29%
Survivability	1%	4%	-69%
Powertrain	5%	9%	-42%
Chassis	6%	7%	-12%
Reliability	2%	2%	20%
Systems	16%	12%	35%
Supportability	5%	4%	16%
Totals	100%	100%	$R^2 = .8903$[21]

Table 2-1: Planned vs. Actual Labor Hours
(Budget performance over two year development phase)

The *Planning Error* (or % deviation) between the *actual* vs. *planned* labor for each subsystem using the current state process is shown in column 4. The results show deltas ranging from -69% to 45%. A negative number indicates the planned labor was under-estimated by the given percentage. Although the data in Table 2-2 suggests budget issues were experienced on the project, the management team had the flexibility to re-allocate funding[22] as subsystem teams showed signs of deviations from their budgets. Unfortunately, even in the most ideal cases where the re-allocations can be effectively tracked, it results in significant coordination effort by management to overcome the initial budget misalignments. It also presents considerable risk to the program of going over budget.

A linear model correlating the original planned labor (from the current process) to the actual hours spent yields an overall R^2 of .8903, indicating a strong predictive relationship. However,

[21] Based on the correlation analysis the relationship between the planned vs. actual labor hours spent for the existing process was R^2=.8903

[22] Having the flexibility to re-allocate budgets across subsystem teams will tend to perpetuate inaccurate bids, as there is little consequence for poor planning. With increased competition and growing financial oversight, the pressure for more accurate bids and detailed planning up-front is increasing.

the unexplained 11% variation can also lead to significant budget deltas as reflected in Table 2-2 above.

Requirements Analysis and Allocation Matrix (RAM): For the newly proposed process the requirements allocation was performed by assigning each of the requirements to their primary and secondary owners. A preliminary P:S ratio of 3:1 was applied to the RAM and the results were translated into the simplified subsystem DSM (as described in Figure 2-7 of the methodology). The resulting matrix is shown in Figure 2-10 below:

Providers (Output)

Receivers (Input)	Body	Telemetry	Auxiliaries	Electrical	Survivability	Powertrain	Chassis	Reliability	Systems	Supportability
Body	1038	8	9	41	22	9	31	0	138	16
Telemetry	10	672	0	47	3	2	2	0	22	2
Auxiliaries	12	0	234	12	1	13	3	0	17	5
Electrical	43	62	4	711	3	23	11	0	28	14
Survivability	30	1	3	7	204	0	4	0	22	4
Powertrain	26	4	5	14	1	381	11	0	28	6
Chassis	10	14	2	13	0	5	324	0	16	10
Reliability	5	8	5	8	3	8	8	30	5	2
Systems	91	36	4	34	29	31	44	0	426	6
Supportability	23	14	17	18	6	32	29	0	24	144
P:S Ratio (3)	1285	819	282	903	272	502	465	30	720	202

Figure 2-10: Requirements DSM – Case Study Example
(RAM estimate based on P:S ratio = 3)

The sum at the bottom of each column indicates the total effort needed for each subsystem to meet all primary and secondary owned requirements (using a P:S ratio equal to ‘3’).

Difficulty Multiplier Calculation (M_i)

Due to the large number of requirements for this project (>1,350), we opted to derive the difficulty multipliers at the subsystem level for efficiency[23]. Employing the difficulty multiplier construct from Figure 2-5, the calculations were performed as follows.

Novelty (N): Scored as a single measure using a Likert-scale ranging from 1 (carry-over products needing little to no improvement) to 5 (new technology).[24] For our dataset, requirements novelty ranged from 2 to 5.

Flexibility (F): Table 2-3 below shows the assessment for design flexibility (F) based on allocations of each requirement tier. Scores of 1, 3, or 5 were assigned to each subsystem based on the percentage of the total Tier 1 ($T1$) and Tier 2 ($T2$) requirements they have been allocated. The scoring is based on the following banding: 1 = < 1%, 3 = 1– 5%, 5 = > 5%.

The final design flexibility score (F) is calculated using the equation: $F = \frac{2T1+T2}{3}$ Since Tier 1 requirements allowed for no flexibility in threshold performance, they were weighted with twice the difficulty versus the Tier 2 requirements. The final scores for design flexibility (F) range from 1.7 to 5.0 for our project.

---

[23] Difficulty multipliers can be generated for each individual requirement, or for entire subsystems based on the cumulative requirements assigned. This approach allows for tailoring based on the number of requirements for the program.

[24] This classification is consistent with guidelines established by the Canadian government for measuring risk and complexity of IT projects.

	Body	Telemetry	Auxiliaries	Electrical	Survivability	Powertrain	Chassis	Reliability	Systems	Supportability	
Tier 1 Requirements (T1)	8	13	0	0	1	5	1	9	19	0	Total: 56
Tier 2 Requirements (T2)	43	69	6	52	11	29	11	0	42	12	Total: 275
Tier 3 Requirements (T3)	295	142	72	185	56	93	96	1	81	36	
% of all T1	14	23	0	0	2	9	2	16	34	0	
% of all T2	77	123	11	93	20	52	20	0	75	21	
SCORE (T1)*	5	5	1	1	3	5	3	5	5	1	
SCORE (T2)**	5	5	3	5	3	3	3	1	5	3	
OVERALL SCORE (F)***	5	5	2	2	3	4	3	4	5	2	

* Scoring Criteria: 1 = 0%, 3 = 1-5%, 5 = >5%

** Scoring Criteria: 1 = 0%, 3 = 1-14%, 5 = 15%+

*** Weighted overall score (T1 given double the importance of T2)

Table 2-2: Scoring for Performance Flexibility
(Flexibility defined as function of requirements tolerance using tiering assessment)

Capability (E): Scored by years of experience[25] in the commodity and industry.[26] Commodity experience (denoted Ec) includes exposure to the specific functional area and/or related technologies.[27] Industry experience (Ei) assesses how well the team understands the customer's needs and the development processes.[28] Capability is calculated using the equation: $E = (Ec + Ei)/2$. In the example each of the variables is set to equal weighting based on SME input.

Per the scoring criteria (reference Appendix 1), the industry and commodity experience *(Ei and Ec)* was scored from 1 – 5, with a 1 indicating the highest level of experience, and a 5 the least. This relationship reflects the fact that design difficulty decreases as experience increases and vice-versa.

25 To score the capability variable *(E)*, only the engineering leads were assessed for years of experience, as they provide design guidance for the team. This approach minimized the amount of analysis required due to the small number of technical leaders in each area (generally three or less).
26 Capability is considered an element of organizational complexity because it resides in the workforce, and not in the product itself.
27 In cases of new technologies being developed, individuals with broad experience in related technologies and legacy systems are expected to become more proficient sooner, and require less training.
28 Actions that may impact the experience / capability variable include employee training to improve technical / industry knowledge, hiring individuals with related experience, employee turn-over, launching products in new industries, etc

	Body	Telemetry	Auxiliaries	Electrical	Survivability	Powertrain	Chassis	Reliability	Systems	Supportability
Industry Experience (Ei)*	1.0	4.0	4.0	4.0	1.0	4.0	4.0	4.0	3.0	1.0
Commodity Experience (Ec)*	1.0	3.0	1.0	2.0	1.0	3.0	3.0	2.0	3.0	1.0
SCORE (E)**	1.0	3.5	2.5	3.0	1.0	3.5	3.5	3.0	3.0	1.0

* Scale: 5 = Least Experience; 1 = Most Experience
** Aveage for Industry Experience and Commodity Experience scores

Table 2-3: Experience of Technical Team
(Experience defined as a function of both commodity and industry knowledge)

Table 2-4 results indicate that the most experienced teams received a score of 1.0, while the least experienced teams received scores of 3.5.

Coordination (Lt)*:* Defined as the efficiency of the organization's structure and processes. The more interaction that is required across teams, the more time / effort will be needed to achieve the goals—particularly if the available project time is constrained. For this reason, coordination challenge has been quantified as a function of two variables including: (1) the percentage of assigned requirements needing secondary coordination (Sr), and (2) the available lead-time slack for design development (Lt).

Sr is calculated by determining each team's percentage of allocated requirements that they are secondary owners of. As the percentage of secondary responsibility increases, the level of coordination will also increase for each team. Calculating this ratio across the other engineering groups indicates a range of 0 – 67% exists. By analyzing the data groupings and reviewing the results with the SMEs, the calculated percentages were translated into a 1-5 scale with 5 representing the highest level of coordination, and a 1 representing the least amount of coordination difficulty.[29]

	Body	Telemetry	Auxiliaries	Electrical	Survivability	Powertrain	Chassis	Reliability	Systems	Supportability
% Secondary Requirements	42%	40%	38%	45%	50%	49%	57%	0%	67%	55%
Coordination of Secondary Requirements (Sr)*	3.0	3.0	2.0	3.0	4.0	3.0	4.0	1.0	5.0	4.0
Lead-time Challenge (Lt)**	5.0	5.0	3.0	3.0	4.0	4.0	3.0	3.0	3.0	5.0
SCORE (L)***	4.0	4.0	2.5	3.0	4.0	3.5	3.5	2.0	4.0	4.5

* Scoring Criteria: 1:0-9%, 2:10-39%, 3:40-49%, 4:50-59%, 5:60%+ (5 = Most Coordination; 1 = Least Coordination)
** Scale: 5 = Least Slack time; 1 = Most Slack time
*** Overall score (average of Coordination and Lead-time Challenge scores)

Table 2-4: Coordination and Logistics Assessment
(Coordination and logistics measures scored as a function of secondary requirements responsibility and available lead time)

The development lead time (Lt) element is calculated based on the amount of slack time that is projected in the development schedule for a given set of requirements. Based on the customer delivery date, the SME's from each subsystem team assessed their requirements and determined if their work must be performed under a compressed schedule (indicating negative slack), under normal scheduling with the critical path (indicating 0 slack), or could be scheduled with some level of flexibility. Based on their assessments of allocated requirements, the subsystem teams scored their lead time difficulty from 1 – 5 as shown in Table 2-5. The initial criterion for scoring 1-5 was selected based on experience from prior programs and can be tailored as appropriate.

The final scores for coordination / logistics challenge (L) range from 2.0 to 4.5 using the equation: $L = (Sr + Lt)/2$. Although equal weighting is applied to Sr and Lt in our case, historical data may indicate that one of these measures will contribute more significantly to development effort. In such cases, a modified weighting can be employed.

[29] Because this step generates a *relative* measure of complexity between groups (rather than an absolute measure) the scoring table may vary between projects

Consolidating all four variables into the difficulty multiplier (Mi) yields the following equation:

$$Mi = \frac{average\ (L+F+E+N)}{unity}$$

where: Unity = 3.0; based on concept of Likert scale 1-5, 3 indicating neutral / baseline score [30]

$$L = \frac{(Sr+Lt)}{2}$$

$$F = \frac{2T1+T2}{3}$$

$$E = \frac{Ec+Ei}{2}$$

$$N = Novelty\ score$$

The difficulty multipliers (*Mi*) have been calculated for each subsystem as shown in Table 2-6. The range of values for this data set is .73 – 1.38 (for Auxiliaries and C4ISR, respectively). The difficulty multiplier is derived from the equation: $Mi = average\ (L + F + E + N)/3$. The value represents "unity" because it's the middle range of the 1-5 scoring and reflects a neutral or baseline assessment for each variable. For example, any of the seven measures scored above a '3' would indicate an increase in development difficulty is needed. Likewise, a score below 3 would indicate less than normal difficulty is present. This approach is consistent with the Likert scale approach, using the center of the scoring range to indicate a nominal assessment.

[30] Reference notes in prior section on anticipated range of difficulty multiplier.

	Body	Telemetry	Auxiliaries	Electrical	Survivability	Powertrain	Chassis	Reliability	Systems	Supportability
Experience / Capability (E)*	1.0	3.5	2.5	3.0	1.0	3.5	3.5	3.0	3.0	1.0
Flexibility (F)**	5.0	5.0	1.7	2.3	3.0	4.3	3.0	3.7	5.0	1.7
Coordination / Logistics (L)*	4.0	4.0	2.5	3.0	4.0	3.5	3.5	2.0	4.0	4.5
Novelty (N)***	4.0	4.0	2.0	5.0	5.0	5.0	4.0	5.0	4.0	3.0
Overall Difficulty Multiplier (Mi)****	1.17	1.38	0.73	1.11	1.08	1.36	1.17	1.14	1.33	0.85

* Scale: 5 = Most Difficult; 1 = Least Difficult
** Scale: 5 = Least Flexible; 1 = Most Flexible
*** Scale: 5 = Most Novelty; 1 = Least Novelty
*** Overall score: (E+F+L+N) / 3 unity

Table 2-5: Difficulty Multiplier Calculation
(Multiplier scores in excess of 1.0 indicate above nominal effort is required)

V. Final Results

The derived difficulty multipliers are applied to the Requirements Effort Matrix to complete the complexity DSM and calculate the total requirements burden (complexity) for each subsystem below.

Providers (Output)

Receivers (Input)	Body	Telemetry	Auxiliaries	Electrical	Survivability	Powertrain	Chassis	Reliability	Systems	Supportability
Body	1038	8	9	41	22	9	31	0	138	16
Telemetry	10	672	0	47	3	2	2	0	22	2
Auxiliaries	12	0	234	12	1	13	3	0	17	5
Electrical	43	62	4	711	3	23	11	0	28	14
Survivability	30	1	3	7	204	0	4	0	22	4
Powertrain	26	4	5	14	1	381	11	0	28	6
Chassis	10	14	2	13	0	5	324	0	16	10
Reliability	5	8	5	8	3	8	8	30	5	2
Systems	91	36	4	34	29	31	44	0	426	6
Supportability	23	14	17	18	6	32	29	0	24	144
P:S Ratio (3)	1285	819	282	903	272	502	465	30	720	202
Difficulty Multiplier	**1.17**	**1.38**	**0.73**	**1.11**	**1.08**	**1.36**	**1.17**	**1.14**	**1.33**	**0.85**
Requirements Burden	1499	1126	204	1001	295	682	543	34	960	172

Table 2-6: Complexity DSM
(Requirements burden is calculated as a function of the allocated requirements and difficulty multipliers)

Results from Table 2-7 show that the multipliers reduced the requirements burden scores for some subsystems, and increased them for others significantly. To validate the methodology a correlation test is run between the complexity assessment scores and actual budget[31] spent in developing the different sub-systems. The analysis shows that the methodology provides a comparable, and slightly improved predictability of over 4% based on the new R^2 = .9319 versus the R^2 of the existing process (.8903). This suggests the unexplained variation can be reduced by accounting for: (1) requirements allocation, (2) product novelty, (3) design flexibility, (4) coordination challenges, and (5) experience

A comparison of estimate methods (by subsystem) is shown in Table 2-8 below. The results indicate predictions from the new method were fairly consistent across each of the subsystem groups.

Subsystem Team	Current Process Estimate	Actual Labor Cost	New Process Estimate
Body	21%	24%	23%
Telemetry	23%	22%	17%
Auxiliaries	2%	1%	3%
Electrical	19%	15%	15%
Survivability	1%	4%	5%
Power train	5%	9%	10%
Chassis	6%	7%	8%
Reliability	2%	2%	1%
Systems	16%	12%	15%
Supportability	5%	4%	3%
Totals	100%	100%	100%
Predictability	R^2=.8903		R^2 = .9319

Table 2-7: Comparison of Labor Hour Estimates (New Method vs. Current)
(New process estimate results in improved budget prediction R^2)

Referring to Table 2-7 we find that the multipliers (Mi) revised the requirements burdens from -28% to +38%, with the most significant increases seen in the Telemetry, Power train, and Systems Engineering areas. Conversely, the largest reductions were shown in the Auxiliaries and

[31] The actual budget costs include engineering labor only, and do not include expenses for material and components for prototyping and evaluation. For an estimate of these costs, historical data from similar programs can be used.

Supportability groups indicating development effort in those areas is impacted significantly by the complexity elements (i.e., novelty, flexibility, team experience, coordination difficulty). These represent areas that can be explored to find opportunities for tailoring complexity.

The Body Engineering group had the highest estimated requirements burden (1,499) per Table 2-7. This appears to be the result of: (1) large number of requirements assigned, (2) high degree of novelty and coordination needed, and (3) minimal design flexibility (see Tables 2-6 & 2-7). In order to reduce this burden, the team may consider modifications to the product architecture to re-allocate requirements out of the Body area and into such areas as Auxiliaries. They may also consider splitting the Body group into smaller subsystem areas such as structures, armor, etc., and re-allocating requirements accordingly. If additional opportunities cannot be found to reduce the technological complexity (due to the lack of requirements flexibility), the team may attempt to increase organizational resources or reduce coordination difficulty by increasing lead time.

Results for the Telemetry and Systems Engineering teams indicate they also have high requirements novelty and low flexibility, but have the added challenge of operating with a less experienced team as shown in Table 2-6. This situation may be improved though added training and/or employing more experienced staff.

Referring to Table 2-6, the most experienced teams were found to be Supportability and Survivability teams (scores of 1.0), while the least experienced teams were Powertrain, Chassis, and Telemetry teams (scores of 3.5). It is interesting to note that the teams reflecting the least amount of experience were also staffed heavily with contract engineers from related industries, which revealed some risk in the current personnel outsourcing strategy.

Coordination and logistics challenges were high for 5 of the 10 subsystem teams, suggesting that program lead-time may be universally difficult to achieve. Relaxing the lead-time for the program would result in a reduction of requirements burden (and uncertainty) for the areas of Body, Telemetry, Survivability, Systems, and Supportability. Collectively, these areas account for over 63% of the total requirements burden for the system.

By analyzing the results from Tables 2-6 and 2-7 for each subsystem, specific opportunities for reducing / tailoring complexity can be identified, and guide management to improve the alignment between the organization and the product, thereby reducing development risk.

VI. Discussion and Implications

The model presented provides a method for assessing and tailoring elements of both PD complexity and organizational resources to improve their alignment and overall launch success. Having the ability to make adjustments to both of these areas simultaneously will provide significant planning flexibility for PD organizations.

The case study results demonstrate that a quantitative assessment of PD complexity can be performed during early concept development to provide an accurate estimate of design effort and cost. The results of the new process yielded slightly improved predictability, with the benefit that it can be re-iterated throughout development as requirements and other information is updated.

The method was validated on a CoPS project example, however, it is designed to be adapted to other requirements-based PD projects as well, regardless of size. Appropriate complexity variables should be selected based on previous research, applicability to the product and the developing organization (see Table 2-1).

The methodology is designed to be implemented without difficulty by extending traditional systems engineering processes related to: (1) requirements allocation and (2) functional architecture development. The process leverages the use of *existing* data to support the analysis without creating the need for costly new data-collection activities—this will enable the calculations to be updated/tracked with minimal additional effort.

The method can be applied across multiple programs simultaneously and aggregated for use in resource planning at the *portfolio* level. Utilizing the model for portfolio analysis will not require all projects to be at the same phase of development maturity. Rather, the method can be applied and re-iterated throughout the lifecycle of any program. Employing the method at the portfolio level will provide an enterprise view and highlight areas that are at risk of exceeding, or under-utilizing available resources—this significantly aids in enterprise-level resource planning.

Because the proposed method can quantify complexity as a function of labor hours (or cost) per functional group, it is also useful in supporting the early bid and proposal processes. In this way, it provides a quantifiable justification of cost for a set of assumptions. It also enables the team to have a significant level of 'system level' understanding at the early proposal stage, which provides benefits in developing a winning bid.

Several of the complexity variables selected can aid in providing a more tailored, cost-effective design that can still meet customer requirements. By enabling the analysis to be performed at the early concept stage, the organization can address life-cycle costs where it will have the most significant impact.

The difficulty multiplier construct provides a broad measure of PD complexity by including elements of both technological and organizational complexity. Six of the seven measures used in the case study example were derived from quantitative data readily available to the organization, including: requirements tiers, project lead time, and employee experience. Applying such suitable metrics will ensure that the assessment is efficient to perform and ideal to incorporate.

The model illustrates that *complexity* can be reduced through a: (1) simplification/trading of requirements, (2) increase in tolerances of requirements, (3) improvement of modularity through re-allocation of requirements, (4) increase in maturity of components and technologies through design strategy decisions, etc.

Organizational tailoring can also be pursued to improve resource alignment by such methods as: (1) increasing staffing in selected subsystems, (2) co-locating teams, (3) assigning more experienced members to complex product areas, and (4) increasing the available development time. By providing opportunities to manage the complexity through adjustments to both the product and organization, the model facilitates improved resource allocations and alignment.

VII. Limitations

The process has been developed for application across many industries, however, further studies are needed to demonstrate the robustness of the process, and its adaptability. Opportunities for tailoring can be achieved with the selection of complexity variables that are appropriate to the product, and based on experience and available data.

The model requires a large amount of available data and input from key SMEs. Often, in the early stages of concept/proposal development, there is limited time and information available for planning. To be successful then, the complexity construct should include relevant variables that heavily leverage existing data from the organization to minimize the assessment burden. Also, the number of SMEs involved should also be managed to ensure most detail can be collected with minimal commitment of resources/time. Finding the optimal level of information vs. predictive accuracy may take several iterations, so, it is recommended that historical datasets be used to validate preliminary modeling. It is understood that CoPS projects by nature have unique characteristics, so, care is needed in applying historical results.

Due to the nature of the defense industry there are a limited number of new programs available to validate the model's performance and robustness. Although the program selected for evaluation of the method is ideal due to its high complexity and large size it represents a single study that needs to be supported with additional cases in the future.

Because a portion of the data was obtained from SME input that was collected in parallel with the existing development process, there is some potential for bias. However, the bias is estimated to be minimal as SME input is based on the collective experience of over twenty years in a given field(s). The evaluation project represents only a fraction of that experience.

VIII. Conclusion

Products and processes are becoming obsolete more quickly, which is driving PD complexity (Cooper, 2000). In the last century we have seen the time it takes for new technology to go from prototype to 25% market penetration reduced by almost 80% (from 50 years to less than 12) (Group, 2010). In this environment there is an ongoing need for complexity management and process tailoring (Kim and Wilemon, 2003). This paper addresses this need by integrating several streams of research including complexity management, organizational alignment, new product development, and process tailoring to establish a model for early project planning and resource allocation.

In the work of Tatikonda and Rosenthal (2000) he suggests that PD organizations assess the novelty of their projects and adjust them accordingly (and explicitly) in the 'front-end' of development. Until now, no single method was available to accomplish this, although several effective methods for calculating complexity after the fact have been proposed. As this research demonstrates, the true value of quantifying complexity is to provide guidance for future design actions.

As PD projects becomes more complex, it is essential to understand the key variables that need to be managed to provide the most benefit to project success. This research demonstrates the importance of modeling the system to identify these variables, and understand how to control them.

Over-commitment of company resources is an important problem in product development that can ultimately lead to launch failure (Yu, Figueiredo, 2010). Effectively quantifying product complexity and ensuring that it is properly aligned to planned organizational resources can help organizations avoid this problem (Gokpinar, Hopp, 2010).

This research provides a novel and effective framework for quantifying complexity at the earliest possible PD stage, receipt of customer requirements. The research extends the work of Gokpinar, et al. (2010) by providing a means for early detection of coordination deficiency. By identifying these challenges at the start of the PD process, organizations will be better able to align their resources before costly development begins.

The methodology also extends the work of Yu, et al. (2010) by providing a detailed resource allocation model for early capacity planning. As his research confirmed, the number of new products an organization can successfully launch is constrained by the degree of their complexities (Yu, Figueiredo, 2010). By using this model to quantify program complexities, detailed capacity planning activities can be accomplished and greater PD success can be achieved.

CHAPTER 3: Framework for Managing Risk Identification and Mitigation in Complex Products and Systems (CoPS)

I. Introduction

One of the most significant barriers to product development (PD) success is a failure to understand complexity and risk in projects (Canada, 2010; Smith, 1992). This is in large part due to the uncertainty that is present in these projects (Browning, Deyst, 2002; Institute, 2008). Today, more than ever, new product development (NPD) is being challenged to acquire technical knowledge quicker in order to manage uncertainty and minimize the risk of failure (Cooper, 2003).

A primary goal of risk management is to reduce uncertainty at the earliest point in the PD process (Browning, Deyst, 2002; Institute, 2008). Risk management practices are aimed at reducing the uncertainty of achieving project goals for cost, schedule and product performance (Simpleman, 2006). Risk management practices have been growing in maturity and are now routinely practiced across many Complex Products and Systems (CoPS) industries including defense, IT, construction, etc (Akintoye and MacLeod, 1997; Chapman, 2001; Kutsch and Hall, 2010; Ren and Yeo, 2004; Simpleman, 2006). When properly implemented, risk management can become a major part of the organizational business activities capable of improving operations in all areas (Akintoye and MacLeod, 1997; Tchankova, 2002; Thompson and Perry, 1992). Today, risk management practices are constantly being updated to improve their techniques and consistency (Chapman, 2001; INST, 2002).

Unfortunately, despite the need for risk management and its clear benefits, there still remains significant disparity in terms of organizational resources being applied to the discipline (Akintoye and MacLeod, 1997; Kutsch and Hall, 2010). Literature suggests this disparity is in part to organization's inability to consistently capture and resolve risks, which prevents them from experiencing the full benefit of risk management (Akintoye and MacLeod, 1997; Kutsch and Hall, 2010). This situation has caused differences in the way risk management is practiced as well. While some project managers work to identify and mitigate risk in advance, others choose to address risks only after they've been realized and become problems for the organization (Yang and Burns, 2004). This later mentality is based on a reluctance to commit resources to events that may not occur (i.e. risks), choosing instead to wait until risks fully materialize even if they do become more costly to address at that point (Kutsch and Hall, 2010).

In an ideal project all risks and uncertainties would be identified proactively, with mitigation activities established to resolve the issues before they impact performance (Cooper, 2003; Institute, 2008). In practice, PD teams operating under condensed timelines and budgets are forced to prioritize the uncertainties they deem as most detrimental to project success (Cooper, 2003; McDonough, Kahn, 1999). This results in an incomplete list of risks being identified and acted upon for PD projects. Unfortunately, it may be the case that these unidentified and unmanaged risks ultimately result in the most significant detrimental impact to the program's cost, performance, and schedule (Chapman, 2001; Tchankova, 2002).

To increase the effectiveness of risk management in PD, literature has indicated the need for greater emphasis to be placed on the *identification* of risks, rather than improving the formality of the process and techniques (Chapman, 2001; Tchankova, 2002). Many consider the identification step to be the most important in risk management (Chapman, 2001). Unfortunately, risk identification has been a challenge (Kutsch and Hall, 2010). For organizations already struggling to manage projects with constrained resources, how can they ensure that proper risks are being identified, and in sufficient quantities to drive success?

Research has suggested that a practical target for risk identification is to document 5 to 10 primary risks per project based on its development complexity (Thompson and Perry, 1992). However, this suggestion is problematic if there is no established method for quantifying PD complexity[32], and no measurable relationship between the number of identified risks and PD complexity. Furthermore, to generate sufficient risks of the wrong type would provide little benefit to the program as well. From a practical perspective then, it remains unclear what the risk identification process should yield to ensure PD success (Kutsch and Hall, 2010). This paper addresses these issues by proposing the use of a complexity construct to provide a preliminary guide for the number of risks that should be identified for each subsystem in the PD project. This research extends the work of Kim and Wilemon (2003) by validating the relationship of complexity to risk, to aid in early risk planning and identification. Data from a major CoPS project are analyzed to determine if *complexity* measures can be used to predict *risk*

The paper also addresses the need for improved risk identification strategies by reviewing several taxonomies to determine the *types* of risks that should be considered in complex development projects. Risk data from previous programs is evaluated to identify the areas of most concern to development teams historically. Finally, a novel method of quantifying risk *effectiveness* is proposed for use in continuous improvement activities and coordinating risk management. Collectively this research provides insights for the improved identification, measurement, and mitigation of risk in CoPS development projects.

The balance of the paper is organized as follows: Section II provides a review of current literature on risk management, its process steps and common frameworks in order to better understand the context of the research. Section III outlines the testing approach and hypotheses

---

[32] For this research PD complexity is understood to encompass all the difficulties and uncertainties posed by the technology during the development, including consideration of the organization's tasks and people Kim, J. and D. Wilemon. 2003. Sources and assessment of complexity in NPD projects. *R&D Management*, 33(1): 15-30.

studied in this research. A CoPS case study example is then presented in Section IV to validate the relationship between measured complexity and risk. PD performance is also reviewed for groups performing minimal risk management activity to begin to understand the value of risk management. Sections V and VI summarize the research results with a review of the insights / limitations and final conclusion respectively.

II. Literature Review

In recent years, risk management literature has put considerable emphasis on the mechanics of risk handling and mitigation[33] rather than the identification of risks (Chapman, 2001). This is likely motivated by the perceived need for increased training in risk management to achieve better results (Akintoye and MacLeod, 1997). However, research indicates some of the greatest benefits can be realized by improving risk identification (Chapman, 2001).

A. Risk Management

In March of 1998, the Department of Defense (DoD) published a guide for risk management to assist defense contractors in administering risk in acquisition programs. The guide was the output of a working group tasked by the undersecretary of Defense in 1996 to support recent acquisition reform by documenting the way the DoD conducts risk management. The Risk Management Guide, now in its sixth edition, has become a standard by which many defense contractors establish their risk process and execute their programs.

[33] Risk mitigation is defined as the approach the organization takes to address potential unfavorable consequence(s) to project cost, schedule or performance (Simpleman, 2006). Mitigation actions include steps to reduce the either the consequences of an unfavorable event (such as installing airbags to increase vehicle crash survivability), or the probability of the event occurring (such as installing a traffic light to reduce vehicle accidents). Reductions in either the likelihood or consequence of a risk will reduce the overall severity of the risk (Simpleman, 2006)

The RM Guide for DoD Acquisition defines risk as a continuous process employing five primary steps including: identification, analysis, planning, implementation, and tracking (Simpleman, 2006). The description of each step is shown in Table 3-1 below:

Step	Description
Identification	Document potential events that will impact the performance, cost or schedule of a product/program
Analysis	Assess the magnitude of each risk in terms of its probability of occurrence and consequence to the product/program
Planning	Identify all activities necessary to reduce the likelihood and/or consequence of a risk event including the: timeline, lead, projected benefit, and required funding of each step
Implementation	Execute the approved mitigation steps aimed at reducing the probability and/or consequences of the risk
Tracking	Monitor the progress of the mitigation activities to ensure success

Table 3-1: Risk Steps from the DoD Risk Management Guide (Simpleman, 2006)
(Five primary steps of the risk management process as recognized by the DoD)

Identification involves answering the question "what can go wrong?" (Simpleman, 2006). For organizations that practice 'opportunity' management in parallel to risk, the identification stage will also include a consideration of the possible gains the program may experience (Tchankova, 2002). In this context, a failure to take advantage of an *opportunity* to reduce cost, shorten the schedule, or increase performance is equivalent to taking a loss (Dickson and Haystings, 1989).

Although there is generally agreement across professional organizations as to what is included in the PD risk management process, there does remain considerable variation as to how the steps are delineated. The Project Management Body of Knowledge (PMBOK) defined process aligns closely with the DoD Guide, but consolidates the last two steps (implementation and tracking) into one step called monitoring and control (Institute, 2008). The Capability Maturity Model Integration (CMMI) model recognizes only three steps in the risk management process including: identification, analysis, and handling—where handling includes all activities related to planning and implementation, but excludes risk monitoring (INST, 2002). While the CMMI model acknowledges risk monitoring as necessary for the process, it is formally captured as a part of the project management function, and not explicitly a step assigned as a risk management function (INST, 2002). The system engineering handbook published by International Counsel on Systems Engineering (INCOSE) also shares the common first two steps, but consolidates planning and implementation as part of a single step called planning (Group, 2010).

The most basic breakdown of the process is that of Thompson and Perry (1992), which recognizes just two steps including: risk analysis and risk management. Here risk analysis includes the activities related to identification and assessment (involving both qualitative and quantitative methods) and risk management includes all the policies and responses related to planning, controlling, and monitoring the risk (Thompson and Perry, 1992).

Each of the five risk process structures presented (DoD, PMBOK, CMMI, INCOSE, and T&P) have been adopted in industry practice and referenced in the literature. A comparison table highlighting the differences between the process breakdown/terminology is shown in Figure 3-1 below.

<table>
<tr><th>Description</th><th>DoD Guide</th><th>PMBOK</th><th>CMMI</th><th>INCOSE</th><th>Thompson & Perry</th></tr>
<tr><td>Document potential events that will impact the performance, cost or schedule of a product/program</td><td>Identification</td><td>Identification</td><td>Identification</td><td>Identification</td><td rowspan="2">Analysis</td></tr>
<tr><td>Assess the magnitude of each risk in terms of its probability of occurrence and consequence to the product/program</td><td>Analysis</td><td>Analysis</td><td>Analysis</td><td>Analysis</td></tr>
<tr><td>Identify all activities necessary to reduce the likelihood and/or consequence of a risk event including the: timeline, lead, projected benefit, and required funding of each step</td><td>Planning</td><td>Response Planning</td><td rowspan="2">Handling</td><td rowspan="2">Planning</td><td rowspan="3">Management</td></tr>
<tr><td>Execute the approved mitigation steps aimed at reducing the probability and/or consequences of the risk</td><td>Implementation</td><td rowspan="2">Monitoring & Control</td></tr>
<tr><td>Provide monitoring and feedback on the progress of the mitigation activities to ensure success</td><td>Tracking</td><td></td><td>Monitoring</td></tr>
</table>

Figure 3-1: Comparison of Risk Process Steps from Prominent Sources
(Risk process content is consistent between leading sources; variations exist in process step definitions)

Risk identification is considered by many to be the most important step in risk management because only after a risk is identified can it be addressed (Chapman, 2001). Risk identification is a continuous process that should begin at the start of a project. It should be carried out across the entire organization and at all levels (Simpleman, 2006; Tchankova, 2002; Thompson and Perry, 1992). Rather than emphasizing individual, isolated risks, the goal of risk identification is to ultimately determine where the organization/project is most susceptible to risk, and what conditions will encourage/discourage these events from happening (both internally and externally) (Tchankova, 2002).

B. Risk Classifications Frameworks

Throughout the literature several risk taxonomies have been developed to help facilitate a methodical approach to identifying risks in PD (McManus and Hastings, 2006). Because taxonomies are established at a high level, and technical development projects share multiple design phases, it is not uncommon for taxonomies to be utilized successfully across many product types and industries, with moderate tailoring. To realize these efficiencies it is useful to consider some of the more significant cross-industry contributions to risk classification frameworks.

In 1993 the Software Engineering Institute (SEI) conducted a comprehensive study to identify repeating risk data within software development projects (Carr, 1993). The study involved the administration of a comprehensive questionnaire to SMEs across numerous government and civilian programs. Based on the questionnaire results, a taxonomy was established that organized risks into three major classes as shown in Table 3-2 below:

Risk Class	Description
Product Engineering	The technical aspects of the work to be accomplished
Development Environment	The methods, procedures, and tools used to produce the product
Program Constraints	The contractual, organizational, and operational factors within which the software is developed but which are generally outside of the direct control of the local management.

Table 3-2: SEI Risk Classification Summary (Carr, 1993)
(Classifications are based on the origin of the risks identified)

In this approach risks are categorized based on their origin. The product engineering class includes risks that originate from the specific work to be performed, including requirements analysis, design, product integration, test, etc. Development environment risks are a result of the process or methods being employed such as development process, management methods, work environment, etc. Program constraints include those risks originating from resources, contracts, or program interfaces (Carr, 1993). Results from the study indicate that the framework provides a thorough list of risks incorporating all functional areas of a program (Carr, 1993). This taxonomy has subsequently been used by the product development community as a template for identifying risks.

Taxonomies such as SEI's that are based on risk *origin* have the benefit of being intuitive because they align with process steps, development phases, organizational structures, and/or company practices (Carr, 1993). Grouping risks by *origin* is also flexible and can be adapted based on the needs of the organization and project.

Following the work of SEI, TRW consolidated several DoD software risk studies spanning nearly a decade and found that over 150 common risk issues had been identified (Conrow and Shishido, 1997). Organizing the risks into similar categories revealed that natural groupings occurred in the areas of: project level, project attribute, management, engineering, and work environment risks. Descriptions are shown in Table 3-3 below:

Risk Group	Risk Issue Details
Project level	Excessive, unrealistic, or unstable requirements, lack of user involvement, or underestimation of PD complexity
Project attribute	Performance shortfalls, unrealistic cost or schedule
Management	Ineffective project management
Engineering	Ineffective integration, assembly, test, quality control, engineering, etc. Unanticipated difficulties associated with the user interface
Work environment	Immature or untried design, process, or technologies selected Inadequate work plans, configuration control, methods, or poor training

Table 3-3: TRW DoD Software Risk Summary (Conrow and Shishido, 1997)
(Classifications of common risks experienced in major software development programs)

While these five categories effectively capture the majority of issues encountered, the method was criticized as being overly broad, making it difficult to assess risk impacts and establish mitigation plans[34] (Conrow and Shishido, 1997). Rather than focus on the operational areas that the risks originated from, Sarbacker et al. (1997) proposed a framework based on the engineering timeline. Using this model, risks were categorized along the three major phases of development including: envisioning, design, and execution (Sarbacker and Ishii, 1997). This classification scheme organizes risks in terms of *when* they will occur in PD. Sarbacker defines *envisioning risk* as the likelihood the product will not meet customer wants, despite meeting the specifications in the design vision. *Design risk* relates to the product not demonstrating the attribute(s) specified in the design vision. *Execution risk* is the concern of not being able to deliver a 'realized' product as designed. Per their approach, after assessment of risks in each area through team discussion, the total program risk is summarized graphically along three-dimensions (x, y, z) to provide a visual interpretation of the total risk impact. Because the assessment is purely qualitative, no numerical scoring is provided along each axis. However, the process does provide a structured method for early risk assessments for innovative products to guide decision makers through the concept approval process (Sarbacker and Ishii, 1997).

[34] Mitigation plans include a list of all actions, stakeholders, budget impacts, timing, and goals of each step planned for reducing the risk (Simpleman, 2006).

Williams (1998) proposed a risk taxonomy based on the environment within which the risks occur, such as the physical, social, political, operational, economic, legal, or cognitive (Williams, Smith, 1998). While this framework is similar to SEI and TRW in focusing on the operational sources of risks, it provides much broader descriptions of risk categories—allowing risks to be captured outside of the immediate project environment. This framework is summarized in Table 3-4 below.

Environment	Description
Physical	Acts of nature, the environment and weather, real estate, etc.
Social	Changes in human behavior, social structures, people's values, culture, etc.
Political	Governments, policy, lections, laws, taxation, etc.
Operational	Organizational activities which impact people, equipment, or property of the company
Economics	Impacts related to the global monetary environment, availability of resources and spending, market conditions, etc.
Legal	Relates to the formalized controls and constraints that exist between states and countries. Includes protections of rights and intellectual property
Cognitive	Relates to the organization's ability to accurately perceive and understand the risk threats. Perception vs. reality

Table 3-4: Environment based Risk Classification of Sources (Williams, Smith, 1998)
(Classification strategy facilitates evaluation of internal and external forces impacting risk)

This classification supports comprehensive analysis of risks facing the organization, both internally and externally (Tchankova, 2002).

Tchankova (2002) suggested a more broad risk identification process that considered the four key elements of risk source, hazard factors, perils, and exposure area. Risk sources include the internal or external areas that are the potential root causes of the risk such as such as market conditions, production materials, customer needs, etc. Hazard factors include the situations that may increase the chance of a risk such as a bad decision, or over sight of a key issue. Perils are un-predictable events such as a fire, industrial accident, natural disaster, etc. Perils always result in negative impacts (Tchankova, 2002). Finally, the exposure areas include those areas impacted by the risks. While Tchankova's framework may initially seem generic, it provides the benefit of being able to assess risks across several contexts (Tchankova, 2002).

In 1999 a study was conducted by the Standish Group International (SGI) to analyze performance results from 7,400 IT projects. The study revealed that only 24 percent were complete within time and budget (Baccarini, Salm, 2004). Motivated by these results, Baccarini et al. (2004) conducted research to determine the most common risks experienced in IT programs based on historical data. After identifying 27 of most common risks from the literature, he conducted a survey of IT project managers to rate each risk category in terms of importance. Table 3-5 below shows the list of risks organized into seven primary categories.

Category	Potential Root Cause
Commercial and legal relationships	Third party performance, IP litigation, friction between clients and contractors
Economic circumstances	Market conditions, competitive actions, software not needed
Human behavior	Staff quality, *insufficient staff*
Political circumstances	Corporate culture, executive support, unrelated requirements
Technology issues	Inadequate documentation, software unfit, poor production system, technology limits, incomplete requirements, poor user interface
Management activities	*Unrealistic project schedule*, requirements changes, user testing, daily progress reviews, accountability, poor leadership, wrong functionality, change management system
Individual activities	Over specification, *unrealistic expectations*

Table 3-5: Common IT Project Risks (Baccarini, Salm, 2004)
(Significant number of risks relate to management and behavioral issues rather than technology)

The categories established are similar to those used by Williams, which include broad classifications of risks based on environmental origins. Potential root cause information has also been provided in each category to facilitate risk identification. The results indicated that the top three risks are a result of: insufficient staff, unrealistic project schedule, and unrealistic expectations (*italicized* in Table 3-5). Baccarini (2004) confirmed that the survey results were consistent with the literature, indicating most problems stemmed from management or behavioral issues, rather than technical. The consolidated research findings were used to establish preliminary checklists for IT project teams to use in identifying risks (Baccarini, Salm, 2004).

A summary of the risk taxonomies presented is shown in Table 3-6 below.

SEI (1993)	TRW (1994)	Sarbacker (1997)	Williams (1998)	Standish Group (1999)	Tchankova (2002)
Operation	*Group*	*PD Phase*	*Environment*	*Business Area*	*Cause*
Product engineering	Project level	Envisioning	Physical	Commercial and legal relationships	Risk source
Development environment	Project attribute	Design	Social	Economic circumstances	Hazard factors
Program constraints	Management	Execution	Political	Human behavior	Perils
	Engineering		Operational	Political circumstances	Exposure area
	Work environment		Economic	Technology issues	
			Cognitive	Management activities	
			Legal	Individual activities	

Table 3-6: Risk Taxonomy Summary
(Risk classification strategy should be selected to support business actions)

While the literature indicates there are many approaches to categorizing project risks, the method selected should be considered carefully as it will provide insights into areas of vulnerability, and possible risk controlling strategies (Tchankova, 2002). Selecting a risk classification strategy that is consistent with operational metrics, departments, or development phases will provide more meaningful and actionable data for program teams (Institute, 2008).

C. Risk Elicitation Techniques

Regardless of the framework used to categorize risks, the process of identifying risks is the first step (Chapman, 2001). Literature has suggested a number of techniques to be used to facilitate risk identification, including: brain-storming, nominal group technique, Delphi method, expert interviews, checklists, and individual assessments (Thompson and Perry, 1992). Although each of these techniques have been recommended in generalized risk literature, there are significant benefits and disadvantages to each (Chapman, 1998).

Chapman (1998) compared three common working-group methods of brainstorming, nominal group, and Delphi technique to determine the merits and drawbacks of each. It is understood that the context of each project plays a key role in determining the effectiveness of each method, so a generalized model was established for comparing methods with considered group size, member characteristics, environment, leadership, etc. A discussion of each technique is provided below.

Brainstorming is a group problem solving technique aimed at spontaneously eliciting creative ideas from all members (Holt, 1996). The method is attributed to Alex Osborn (1938) as a way to quickly generate a large set of data/options without fear of judgment or criticism from the team (Chapman, 1998). Guidelines for brainstorming include: suspending criticism, encouraging creativity, and building on ideas through combination and improvement (Chapman, 1998). The method encourages power-balance between participants, suspension of judgment, the absence of personal agendas, etc. however, this is often difficult to achieve in practice due to common inter-group dynamics (Holt, 1996). Because of the social challenges involved in brainstorming, the technique has limitations (Chapman, 1998). Isaksen (2005) noted the three key barriers to brainstorming include the emergence of judgments during ideation, members giving up on the group, and inadequate structure of the interaction. As such, brainstorming may be unsuitable for initiatives involving high degrees of technical expertise, subject to manipulation of the people involved, or requiring high degree of documentation (Rickards, 1974).

The Nominal Group Technique (NGT) was developed in the late 1960’s by Andre Delbeeq and Andrew Van de ven as a more formalized method of generating, assessing, and consolidating group input (Chapman, 1998; Scott, 1983). The technique provides a quick decision while ensuring input from all participants has been considered. Using this method all participants are asked to document their ideas and submit them to the facilitator for group evaluation (anonymously) and rank-ordered. Because this technique supports balanced participation, its value increases as group size increases (Chapman, 1998; Scott, 1983). Research has also shown NGT provides better results in terms of the number, uniqueness, and quality of ideas generated.

Documentation is improved with NGT with the increased formalization of the process. This results in a more direct approach to disagreements and a decrease in extraneous conflict between participants (Chapman, 1998).

The Delphi method was developed in the mid 1950's by Rand Corporation as a means of achieving group consensus based on collective intelligence (Armstrong, Green, 2007). The process collects input from individual respondents (separately and anonymously) using questionnaires. The results are then consolidated and summarized by a facilitator and distributed to the team. Additional iterations can be performed based on the consolidated data (Chapman, 1998). The Delphi method provides several benefits including accommodating unlimited participants, minimizing pressure to conform, and eliminating in-process criticism. Issues related to the Delphi technique include the time required to complete the analysis, the inability to resolve participant conflicts, difficulty in clarifying questions/responses among participants, and the feeling of detachment from the problem solving effort (Chapman, 1998). The Delphi approach is appropriate for decisions involving differing opinions, a need to correlate informed judgments, and a need to educate participants about diverse options (Hasson, Keeney, 2000). Literature has shown that it provides a more accurate result than unstructured problem solving methods (Chapman, 1998).

The success of risk identification depends heavily on the in-depth knowledge and experience of the stakeholders (Bajaj, Oluwoye, 1997). Because the collective knowledge of a group exceeds that of an individual, pursuing identification strategies that rely solely on the risk analyst's knowledge may not always be optimal (Bajaj, Oluwoye, 1997). There are instances, however, when an individual assessment may be the preferred approach such as with SME's operating under strict time constraints.

In a study of risk analysis approaches employed by construction firms, Bajaj et al. (1999) found that informal risk reviews by senior staff was the preferred method of identifying risks during the initial proposal stage (Bajaj, Oluwoye, 1997). The number of staff members depended on the size of the project. Although these results would seem to contradict the studies that concluded large formalized ‘working groups’ as being best for identifying risks, in some cases, the issues of timing and convenience outweigh their benefits.

In research conducted in the UK by Project Risk Analysis and Management (PRAM) involving a wide range of industries, a simple checklist was identified as the preferred method of risk identification, and used heavily by most participants (Bajaj, Oluwoye, 1997). Checklists can be employed effectively by participants with varying levels of experience, and often provide an excellent summary of historical data based on past experience (Bajaj, Oluwoye, 1997). Checklists are appropriate for both traditional and complex programs sharing similar requirements.

In Bajaj’s research, five risk review strategies were identified by the surveyors as being appropriate to use. The techniques listed in order of popularity include: opinion of 1 or 2 experienced persons (85%), circulating info to the team (79%), judgment of the estimator alone (63%), review in department meetings (52%), external consultant (47%), and brainstorming (42%). The results indicate organizations prefer assessments by 1 or 2 people significantly more than group analysis such as departmental meetings and brainstorming. However, the authors still recommended that each of these methods be conducted as group exercises as much as possible, as the experience of the individual can be limited (Bajaj, Oluwoye, 1997). It was also noted that every technique was employed to some degree by at least 40% of the companies based on the circumstances of their programs. In situations of limited time and resources a simplified method is often preferred, and its practical value should not be under-estimated.

Understanding when to apply each elicitation method has been a challenge for the risk identification process (Chapman, 1998). When implementing one of the working group assessments, selecting representatives of the core design team is critical to ensuring that risk data is collected thoroughly (Chapman, 2001). Research has shown that group input provides more diverse and in-depth data based on the cumulative experience of the participants (Chapman, 1998). Unfortunately, as group size increases, these techniques become less efficient due to the decreasing cohesiveness of the group caused by personal conflicts (Harrison, 1975). This supports the conclusion that no single method is ideal in all cases (Bajaj, Oluwoye, 1997).

Perhaps the most significant aspect of risk identification involves the decision of which concerns to accept as risks and pursue mitigation (Cooper, 2003; McDonough, Kahn, 1999). Although little research has been done in this area, the organization’s approach to this question will have major implications on the effectiveness of their risk process (Kutsch and Hall, 2010). Its impact cannot be understated. In a study conducted by Kutsch and Hall (2008), they consider the case of risks being deliberately ignored by project managers because they are deemed ‘irrelevant,’ or perceived to have an overly negative reflection on the program. The study defined irrelevant risks in three ways including: 1) Untopical – Information deemed ‘off-topic’ and not pertinent to the project; 2) Taboo – Risks deemed inappropriate because their exposure creates anxiety or puts the program at risk of being viewed poorly or cancelled, and 3) Undecidability – Risks unclear in terms of their accuracy. The study determined that it was common for project manager to practice *‘deliberate ignorance’* towards risks, resulting in adverse affects to the risk process. Project teams observed several negative behaviors manifested because of the social/cognitive tendencies of deliberate ignorance. These behaviors included tendencies to:

- Accept only easily identifiable risks, regardless of the severity of consequence
- Accept only risks that could easily be mitigated
- Accept only risks to areas they are knowledgeable about
- Accept only risks that are near certain to occur
- Avoid risks involving human and managerial elements
- Avoid risks that are perceived as too negative

The research concluded that many projects are impacted by deliberately ignoring certain types of risks. In some cases these risks may be the most damaging to project success (Institute, 2008). In implementing the risk identification process, steps should be taken to avoid instances of deliberate ignorance including increasing awareness of these tendencies and training for identification of appropriate risks. If left unaddressed, these behaviors can result in the risk process becoming ineffective, or even counter-productive in some projects (Kutsch and Hall, 2010).

III. Conceptual Framework and Method

Literature has indicated that risk is proportional to PD complexity, and a lack of risk management will negatively impact project success (Institute, 2008). Unfortunately, to date there have been no known studies that sufficiently quantify these relationships beyond proportionalities. In addition there have been no methods developed for using these relationships to predict and plan in PD. Given that the research on *quantifying* complexity and risk in PD is still in its infancy, this is not surprising. To address this gap this research aims to demonstrate that PD complexity can be used as a predictor of risk in CoPS products, and validate that a lack of risk management will have a negative impact on PD success. In the next section these questions are formalized as hypotheses and tested using data that is commonly available in CoPS projects.

H1: The amount of risk in a project increases with the amount of PD complexity

To test the first hypothesis, a correlation analysis is run comparing estimates of PD complexity (Ci) to risk activity performed (ri). Figure 3-2 below illustrates the proposed conceptual framework.

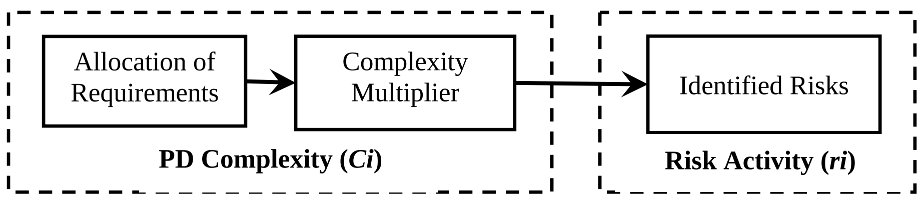

Figure 3-2: Framework for Proactive Assessment of Necessary Risk Management Activity
(Complexity multiplier consists of elements of technology and organizational complexity)

In this framework *PD complexity* is estimated using the method presented in chapter one. Here nine separate data points are generated that represent the complexity of each subsystem (i). The complexity estimates are calculated as a product of the allocated customer requirements (per subsystem) and a complexity multiplier consisting of several variables impacting development effort (including design flexibility, technology novelty, coordination, and experience) .

The amount of *risk activity performed* (per subsystem) is estimated based on the historical number of risks that were identified and managed by each subsystem team through development. Within the two year development phase a total of eighty risks were documented across nine functional areas. Risk identification was performed by all subsystem team using multiple methods including: brainstorming, individual assessment, expert interview, and checklist(s). All subsystem teams were proficient in risk management practices and had equivalent access to risk process tools, support, and materials ensuring the opportunity to identify and manage risks was consistent.

Only risks that were formally reviewed and approved by the program team were included in the list. Risk approval requires the input and consensus of six risk-board members comprising the core management team. Multiple functional areas were represented in the risk board including: program management, engineering, manufacturing, system integration, supply chain, quality, contracts, and finance.

Approved risks were documented and tracked electronically from inception through closure using established risk management software to ensure accurate reporting and status. Any concerns that were deemed to have already occurred were classified as problems and addressed separately from risks. The risk list only includes those items that could be pro-actively resolved before they occurred.

H2: A lack of risk management in complex projects will negatively impact project success

To test the second hypothesis (H2) it is necessary to evaluate the performance of subsystems that employed *low levels* of risk mitigation activity versus those performing *higher levels* of risk mitigation. Using the same historical risk data as above, we assume the nominal amount of risk activity required for each group is equivalent to the percentage of total complexity. This approach will ensure that risk activity is consistently applied across the program for all subsystem groups. In cases where the percentage of risks identified by the subsystem (ri) was less than its percentage of estimated complexity (Ci), a negative project performance is expected according to hypothesis 2. Thus, the relationship between Ci and ri can be described as follows:

$Ci < ri$ for subsystems performing the highest level of risk management activity
$Ci = ri$ for subsystems performing a nominal level of risk management activity
$Ci > ri$ for subsystems exhibiting a lack of risk management activity

The subsystems (i) with the largest negative delta between identified risks (ri) and estimated complexity (Ci) were deemed to have a lack of risk activity, such that:

Subsystems with lack of risk activity = $Maximum\ Di, where\ Di = ri - Ci$ *(for each subsystem i)*

To validate H2 the subsystems reflecting a lack risk management activity are evaluated for negative impacts to their PD performance. PD performance can be measured in many ways including requirements compliance, cost, schedule, etc. In order to provide a robust evaluation of subsystem performance several metrics were included in this research including:

- Engineering development cost – measured in engineering labor hours used throughout the development phase
- Non-compliant requirements (NCRs) – Requirements that do not meet minimum threshold performance
- Test failure modes - Significant issues found after the vehicle was complete and was being evaluated for overall system performance capability

The metrics were selected based on the availability of information to the researcher, and their ability to address multiple elements of performance (including both cost and requirements compliance).

IV. Case Study Analysis

A. Complexity vs. Risk Identification

Hypothesis 1: Based on the results of Chapter 1 the total complexity of each subsystem is shown in Figure 3-3 below. The complexity scores (Ci) have been normalized to reflect the percentage of total complexity for each subsystem area. The risk results (ri) have also been normalized and added to Figure 3-3 for comparison purposes. The data reflects the total number of risks identified by each group through the 20 month period of development.

		Telemetry	Auxiliaries	Electrical	Survivability	Powertrain	Chassis	Reliability	Systems	Supportability
REQUIREMENTS	Telemetry	672	0	47	3	2	2	0	22	2
	Auxiliaries	0	234	12	1	13	3	0	17	5
	Electrical	62	4	711	3	23	11	0	28	14
	Survivability	1	3	7	204	0	4	0	22	4
	Powertrain	4	5	14	1	381	11	0	28	6
	Chassis	14	2	13	0	5	324	0	16	10
	Reliability	8	5	8	3	8	8	30	5	2
	Systems	36	4	34	29	31	44	0	426	6
	Supportability	14	17	18	6	32	29	0	24	144
COMPLEXITY	Requirements Allocation (w/ P:S ratio = 3)	819	282	903	272	502	465	30	720	202
	x Difficulty Multiplier	*1.38*	*0.73*	*1.11*	*1.08*	*1.36*	*1.17*	*1.14*	*1.33*	*0.85*
	Total Complexity Score	1126	204	1001	295	682	543	34	960	172
	Percentage of Total Complexity (*Ci*)	**22%**	**4%**	**20%**	**6%**	**13%**	**11%**	**1%**	**20%**	**3%**
RISK	Number of Risks Identified	21	1	25	0	15	5	2	11	2
	Percentage of Total Risks Identified (*ri*)	**26%**	**1%**	**30%**	**0%**	**18%**	**6%**	**2%**	**13%**	**2%**

Figure 3-3: PD Complexity and Risk Estimates for Sample CoPS Project[35]
(Risk and complexity data have been normalized for comparison)

The correlation analysis confirms that there is a strong relationship between development complexity and identified risks with R^2 = 0.781[36] (reference Figure 3-4 below). The data supports the hypothesis that risk activity can be estimated based on complexity.

---

[35] Results for the body subsystem have been omitted from the analysis due to their risks being deemed classified.
[36] The results are contingent on having an accurate and reliable predictor of subsystem complexity such as presented in section 1

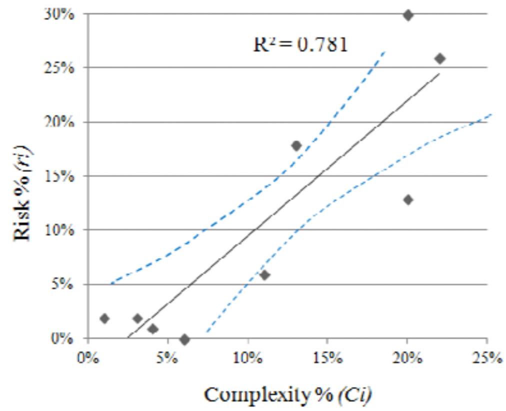

Figure 3-4: Correlation of Estimated Complexity vs. Risk for Sample CoPS Project
(A strong relationship is indicated between the complexity and risk measures)

Because this correlation assessment would be used to facilitate risk identification and tracking, the primary objective is not to maintain a 100% correlation, but rather to ascertain where additional risk activity may be needed. Data points near the extremes of the confidence interval represent areas where additional focus should be placed. The correlation plot for this data set indicate the standard deviation increases as the points move further to the right (into the 3rd and 4th quartiles of the graph), as complexity is increasing. This suggests that a greater emphasis should be placed on risk actions for subsystems with higher complexity --particularly as there are limited resources available for risk mitigation.

Hypothesis 2: In order to validate the importance of risk management activity on PD performance, H2 is evaluated to determine if subsystems exhibiting a lack of risk management activity realized any negative impact on PD performance.

Figure 3-4 (above) indicates the subsystems with the largest deltas ( ) include survivability, chassis, and systems engineering as shown by the three data points furthest below the correlation line. These points represent the subsystems performing a lack of risk management.

The plotted data points $(Ci,\ ri)$ of these subsystems are (*6%, 0%*), (*11%, 6%*), and (*20%, 13%*), respectively. The performance metrics for each of these subsystems are shown in Table 3-7 below, including: test failures, non-compliant requirements (NCRs), and development cost. Column 4 titled *'Secondary Responsibility Requirements'* shows the percentage of requirements that subsystems provide secondary input for, but are not lead responsible. This is relevant because poor performance by subsystems with a high percentage of secondary responsibility may be manifested in other areas—those with the primary responsibility.

	Complexity *(Ci)*		Identified Risks *(ri)*		Delta *(Di)*	Secondary Responsibility Requirements	Test Failures	NCR's	Cost
		%		%		%			% Over
Telemetry	1126	22%	21	26%	4%	18%	0%	12%	16%
Auxiliaries	204	4%	1	1%	-3%	17%	0%	11%	-18%
Electrical	1001	20%	25	30%	10%	21%	8%	28%	-8%
Powertrain	682	13%	15	18%	5%	24%	**14%**	15%	**106%**
Chassis	**543**	11%	**5**	6%	**-5%**	30%	**73%**	**7%**	**35%**
Reliability	34	1%	2	2%	1%	0%	0%	7%	-1%
Systems	**1013**	20%	**11**	13%	**-7%**	**42%**	5%	9%	-12%
Supportability	172	3%	2	2%	-1%	29%	0%	12%	3%
Survivability	**295**	6%	0	0%	**-6%**	25%	0%	0%	**278%**

Table 3-7: Subsystem Performance Metric Summary for Sample CoPS Project
(Cost and performance metrics have been included to provide a thorough assessment of project performance)

Although no *single* metric was found to correlate directly with Di, it is understood that subsystem performance may be impacted in a number of ways by unidentified and unmitigated risks. Referring to the data of Table 3-7, there are several observations that can be made about the subsystems performances. Key metric data has been placed in bold.

The Survivability group identified 0 risks, despite having responsibility for an estimated 6% of the overall development complexity (Di = -6%). Cost over-run in this group was the highest of any subsystem team at 278% of planned budget. Fortunately, the relative development costs for this subsystem were small in comparison to the overall budget, and accounted for just 1.4% of the total. Therefore, despite the major cost over-run, the impact to the program was minimal. However, the negative cost results suggest an increase in risk management activity was needed, and may have improved performance through improved planning and early mitigation.

The Chassis group identified 6% of the total technical risks throughout development, yet had responsibility for over 11% of the estimated development complexity ($Di =$ -5%). The performance metrics indicate there were significant performance issues realized in this area, which accounted for 73% of all test failures identified. A review of cost data shows that the Chassis team also experienced the third highest cost over-run of the nine subsystems evaluated at 35%. Underperforming so significantly in both performance and cost suggests there were major challenges that needed to be overcome. The data suggests that additional risk planning may have been beneficial in proactively mitigating, or reducing the shortfalls in performance and/or cost.

Systems engineering (SE) identified 13% of all risks throughout development, and had been assessed with an estimated 20% of the complexity ($Di =$ -7%). Although the data does not reveal a significantly negative impact to any one of the metrics, it does indicate that they had impacts on 5% of the testing failures found, and 9% of the requirement’s non-compliances. These numbers were not considered extremes compared with the other subsystems groups. Although the results do not seem consistent with the results from the survivability and chassis groups, further investigation reveals that 42% of the requirements allocated to the SE required their secondary input only which is the highest percentage of any of the nine subsystems. This suggests that subpar performance within the Systems engineering group may have been manifested in the metrics of other groups. After reviewing the details of non-compliant requirements (NCRs), and having discussion with affected SMEs, it was determined that the most significant non-compliances related to vehicle weight, an area that the SE had considerable secondary involvement[37]. The impact of these non-compliances is a significant manifestation in both the powertrain and chassis areas as the increased weight resulted in reduced vehicle performance and maneuverability, which were identified during testing. This detail is supported in the metric summary table as the chassis and powertrain areas realized the highest test failures, accounting for a combined 87% of all failures identified. The groups also experienced the second and third highest cost over-runs of 106% and 35% over budget (for Powertrain and Chassis, respectively). Survivability was the only other subsystem with a higher cost over-run, which was a team that identified zero risks.

The data supports hypothesis 2 and suggests that project teams exhibiting a lack of risk activity can experience negative performance impacts in terms of cost and/or performance. Although no single metric was found to correlate to Di directly, indications of negative impacts could be seen across various metrics as Di increases.

Other observations include the fact that two of the most complex subsystems, Electrical and Telemetry, accounted for over 42% of the complexity, yet identified 56% of the program risks, resulting in a positive Di (14%). This indicates that these groups performed a *higher* degree of risk management compared to their portion of complexity. The metric results for these subsystems show they were responsible for only 8% of the total test failures, and over-ran their budget cost by 4.8%. Their combined NCR's percentage was 40%, which is slightly lower than the 42% of requirements they had been allocated. In total, the metric results indicate their performance resulted in no significantly negative impact to performance or cost—despite having responsibility for 42% of the development complexity.

[37] The details related to specific requirements and their non-compliances are being maintained as confidential.

B. Risk Sources Identified

To analyze the data further for trends in risk identification, the eighty risks are grouped in terms of their impact(s) and source(s). Table 3-8 summarizes the risks in terms of their impact to cost, schedule, or performance based on the risk taxonomy employed in several CoPS industries.

	Performance	Cost	Schedule
Telemetry	8	0	13
Auxiliaries	1	0	0
Electrical	17	0	8
Survivability	0	0	0
Powertrain	11	1	1
Chassis	3	2	0
Reliability	2	0	0
Systems	6	3	2
Supportability	1	1	0
TOTAL	49	7	24
	61%	**9%**	**30%**

Table 3-8: Summary of Risk Impacts for Sample CoPS Project
(Risk taxonomy consistent with the DoD Guide for Risk Management)

The results indicate the majority of risks (61%) are performance related, while only a fraction (9%) were found to be cost risks. These percentages are not unusual for an early development program that is focused on establishing the current limits of technology and system capability. As projects progress through their lifecycle, the frequency and types of risks identified will evolve (Institute, 2008). In the observed data, schedule risks comprised 30% of the risks as a result of longer lead times being anticipated for developing the high-tech requirements. The data indicates that technology innovation and complexity were responsible for over 90% of the technical risks identified.

Development cost[38] is often less of a concern at the early PD stages where design strategies are expected to change and flexibility is valued higher than such elements as quality and initial performance. Understanding that the early development phases of CoPS projects often follow a

[38] All the risks in this project were specific to the early development phase, therefore cost risks do not include the cost of production.

similar allocation of risk *types* (i.e., technical, schedule, or cost) can provide guidance in planning for future risks and vulnerabilities. This detail can aid in improving risk identification effectiveness.

Analyzing the risk data across multiple taxonomies can provide further insight into effective identification strategies for the future. Having organized the risks by *type*, they are next analyzed by *source*. After reviewing the risk details with SMEs, it was determined that 95% of them could be categorized into one of the six areas below. A category of 'other' was provided for the remaining (5%) miscellaneous risk sources.

Packaging:	Risks related to the physically coupling components together within available space and dimensional constraints
Requirements difficulty:	Risks related to achieving the threshold (minimum) performance as defined in the requirements documents
Changes:	Risk related to unexpected changes in the design or requirements strategy
Process execution:	Risk that the execution of work will not progress as quickly as needed to support the project timeline due to process inefficiencies, interruptions, or initial lack of lead time
Information/decisions:	Risk that formal information or milestone decisions will not be available/completed in time to initiate key processes, or confirm design strategy
Interaction:	Risk that approved design strategies between subsystems are in conflict with one another and will cause performance or cost impacts when integrated into the larger system
Other:	All other miscellaneous risk sources not addressed in the other 6 areas

Table 3-9 below reflects the detailed count and percentage of risk sources[39] by subsystem/IPT area.

[39] Although more detailed sub-sourcing categories could be established from the data, it was determined that the current fidelity supports effective analysis.

.	Packaging	Requirements Difficulty	Changes	Process Execution	Info / Decision	Integration	Other	
Telemetry	1	4		9	7			21
Auxiliaries		1						1
Electrical	4	8		4	6	1	2	25
Survivability								0
Power train	3	7	1			1	1	13
Chassis	1	2		2				5
Reliability		2						2
Systems		4	2	2	2	1		11
Supportability				1		1		2
TOTAL	9	28	3	18	15	4	3	
	11%	**35%**	**4%**	**23%**	**19%**	**5%**	**4%**	

Table 3-9: Summary of Risk Sources for Sample CoPS Project
(95% of risks identified in the development phase could be attributed to six categories)

The data reveals that *requirements difficulty* was the cause of most risks at 35%, with the electrical and power train subsystems having the largest number of risks in this area (at over 50% combined). The table also shows that Reliability had two requirements difficulty risks, which accounted for 100% of their documented risks. These results are consistent with the complexity analysis from Section 1, which indicated that Electrical, Power train, and Reliability had a '5' for requirements novelty [40]. This data supports the concept that complexity assessments can provide guidance into risk areas that should be identified and tracked.

The next largest risk sources were from process execution (23%) and information/decision making (19%), which combined accounted for over 40% of the total risks identified. These risk categories relate to planning and execution, which are key functions of project leadership. Ensuring that proper documentation and decision-making is occurring can reduce this risk significantly. Because the program was operating under a compressed timeline, there was additional risk in these areas. Understanding these coordination challenges up front will help facilitate effective risk identification and mitigation strategies.

[40] Per Table 2-5,requirements novelty was assessed on a scale from 1 to 5, with 5 indicating the highest degree of difficulty.

Packaging concerns were the next highest contributor of risks accounting for 11% of those identified. For programs involving the integration of multiple subsystems such as this one, a significant number of packaging risks are anticipated. Early design and modeling activities can be used to mitigate these risk areas. The data indicates that electrical and powertrain were the subsystems with the most risks in packaging, having identified 7 of the 9 risks in the category. However, these results are deemed reasonable given the large number of electrical modules required, and the limited space available for packaging in the engine compartment. A focus on early modeling and integration in these areas with applicable software tools can help reduce the risk.

C. Measuring Effectiveness

The risk management activity for the program was generally considered to be highly effective as 80 technical risks were identified, and nearly 70% of those risks were mitigated or avoided [41]. To determine a more specific measure of risk management effectiveness, a process of summarizing mitigation progress is proposed. Figure 3-5 shows the initial and final summary matrices that track the reduction of risk severities from initial identification to final mitigation/closure.

---

[41] Risks mitigation is defined as the actions (steps) that reduce a risk to an acceptable level. Risk avoidance is achieved when the root cause and/or consequence is completely eliminated such as with alternative design decisions or concepts (Simpleman, 2006).

Consequence (c_i)

$A_{i,j}$	1	2	3	4	5
5		3	3		1
4		3	5	6	2
3		2	7	2	3
2	1	6	5	1	3
1		1	1	3	3

Likelihood (k_j)

Initial Assessment

Consequence (ci)

$B_{i,j}$	1	2	3	4	5
5					
4					
3					
2		5	2	1	
1	25	5	4		

Likelihood (k_j)

Final Assessment

Figure 3-5: Risk Effectiveness Matrices for Sample CoPS Project[42]
(Reductions in risk severity assessments are indicated by numbers moving diagonally from the upper right to the lower-left corner)

The numbers in the matrices represent the quantity of risks assessed at each severity level (likelihood and consequence) at a given point in time. The initial severity scores (time *t=0*) are shown in the matrix to the left ($A_{i,j}$). As mitigation actions are completed for each risk, the severity scores decrease as indicated by the numbers moving diagonally from the upper right-hand corner (high severity) to the lower-left (low severity)[43]. The final risk severities shown in matrix $B_{i,j}$ reflect the final residual risk remaining after mitigation actions have been complete. Due to budget and timing constraints, it is often not possible to eliminate all likelihood and consequences of a risk (Simpleman, 2006). However, in many cases risks can be reduced to a more acceptable level as indicated in the final summary matrix, $B_{i,j}$.

Building off this approach of summarizing risk severities, we propose a method of quantifying risk effectiveness into a single measure. The process is accomplished in the following four steps:

1. Create the *initial state* matrix ($A_{i,j}$) summarizing all risks severities at time *t=0*,
2. Determine the *final state* matrix ($B_{i,j}$) summarizing all severities after mitigation actions have been complete,
3. Calculate the total risk severity of each matrix by summing the product of all likelihood scores x consequence scores for each risk, and

[42] Avoided risks will have no residual severity, therefore will not be reflected in Matrix $B_{i,j}$.
[43] Risk severity is a function of the likelihood and consequence of the risk. The lower the probability of occurrence and impact to the program, the lower the severity will be.

4. Determine the delta between the total risk severities (R_0 and R_f) to quantify the overall effectiveness of the mitigation actions.

The method can be expressed in the following general form:

Risk effectiveness *(E_{ff})* = Initial risk severity *(R_0)* – Final risk severity *(R_f)*

$$R_0 = \sum_{j=1}^{m} \sum_{i=1}^{n} (k_j)\,(C_i)\,A_{i,j}$$

$$R_f = \sum_{j=1}^{m} \sum_{i=1}^{n} (k_j)\,(C_i)\,B_{i,j}$$

$$E_{ff} = \sum_{j=1}^{m} \sum_{i=1}^{n} (k_j)\,(C_i)\,A_{i,j} - \sum_{j=1}^{m} \sum_{i=1}^{n} (k_j)\,(C_i)\,B_{i,j}$$

where:

$A_{i,j}$ $i \in [1,2,3,4,5]$, $j \in [1,2,3,4,5]$ = Initial risk severity matrix (at time $t=0$)
$B_{i,j}$ $i \in [1,2,3,4,5]$, $j \in [1,2,3,4,5]$ = Final risk severity matrix
C_i, $i \in [1,2,3,4,5]$ = Column constants for consequence scores
k_j, $j \in [1,2,3,4,5]$ = Row constants for likelihood scores

Applying the formulas above, the initial and final risk severities (R_0 , R_f respectively) and effectiveness (E_{ff}) scores are calculated for the sample CoPS project as:

$$E_{ff} = R_0 - R_f = 590 - 87 = 503$$

The results indicate that 85% of the initial risk severity (R_0) was mitigated through the development phase. These numbers support management's assessment that the risk management process was successful on the program. The method also provides an opportunity to conduct quantitative comparisons of risk effectiveness across other programs.

V. Insights and Limitations

Literature indicates that project performance will improve with risk management (Conrow and Shishido, 1997; Yeo and Yingtao, 2009). Unfortunately, it is unclear how much risk management is needed to ensure success. This research takes a first step in addressing this question by demonstrating that risk management activity can be successfully estimated from PD complexity. The primary goal in using complexity estimates to forecast risk is not to achieve a 100% correlation, but rather provide guidance as to where risk management should be applied most aggressively--such as the subsystems with the highest degree of complexity, residing in the upper end (i.e. third and fourth quadrant) of all subsystem complexity scores.

While the results are preliminary, they are intended to lay the groundwork for future, more extensive studies in managing complexity and risk in PD.

The method presented requires an established risk process to be in place for risk identification and mitigation. Today, many organizations are working to improve their risk process through training from CMMI, MPI and other process standards organizations. Therefore, the effectiveness of the existing risk processes will constrain the results of the proposed process.

The proposed process relies on complexity scoring to estimate the level of risk activity needed. Literature has indicated the need for more quantitative and accurate assessments of complexity to be available (Gokpinar, Hopp, 2010; Sosa, Eppinger, 2004). Until more widespread methods for estimating PD complexity are established and accepted, the universal application of this approach may be limited.

To be successful, the method requires a culture of embracing risk management and risk identification to ensure sufficient reporting of risks is being done. In organizations plagued by such tendencies as risk avoidance and deliberate ignorance, the true benefits of this process, and risk management in general will never be realized (Kutsch and Hall, 2010).

The data used in this research originated from a major DoD project employing nine separate subsystems tracked over a 20 months period, which provides high confidence that the results are reflective of common practice. Due to the broad nature of available data, several interviews were conducted with subject matter experts (SMEs) to provide additional details where necessary. As a result of the researcher's availability to SMEs, and the completeness of the data tracked in the risk system, the quality of the data is believed to be high.

No research has been done to standardize risk classification frameworks for CoPS projects. This study serves as a starting point by summarizing the most common sources of risks experienced in complex development programs. It is recommended that the risk sources identified be considered in future projects to help guide and improve risk identification. Referencing historical data can aid significantly in identifying key risk sources and mitigation strategies.

The risk effectiveness metric (E_{ff}) provides a novel method for measuring the success of risk management by providing quantitative evidence of mitigation success. However, because risk severities are qualitative estimates containing elements of probability (i.e., likelihood) they are not additive. Therefore, the proposed method is not intended to provide an *absolute* assessment of project risk, but a relative assessment of overall risk severity.

Applying this method across multiple projects can provide a means of comparing/benchmarking the effectiveness of risk programs for continuous improvement activities. The metric also provides an accurate assessment of risk performance because it is a function of both the quantity of risks identified, as well as the reduction in risk severities (likelihood and/or consequence).

Like any metric, the risk effectiveness measurement is susceptible to gaming by individuals that are not focused on the goals of continuous improvement. Attempting to increase risk effectiveness scores by exaggerating risk severities or mitigation efforts should be strongly discouraged. The inclusion of a formal risk review board as was used in the company that was studied can protect against such issues, and ensure all assessments are accurate and relevant.

Risk identification is thought by many researchers to be the most important aspect of risk management, and the most significant contributor to its success (Tchankova, 2002). Risk literature proposed several taxonomies/frameworks that could be used to guide the risk identification process. Analyzing the risk data across multiple taxonomies (including origin and impact) will help to highlight areas of vulnerability to future projects, and suggest mitigation strategies that can be employed going forward.

VI. Conclusion

The inability to manage complexity is cited as one of the primary reasons for product development failure (Smith, 1992). Complexity is a function of uncertainty and risk management is a process used to manage uncertainty (Browning, Deyst, 2002; Institute, 2008; Simpleman, 2006). Effective risk management has been shown to improve PD success (Cooper, 2003). Unfortunately, despite the importance of risk management there remains a significant disparity of resources applied to risk between organizations and industries (Akintoye and MacLeod, 1997; Kutsch and Hall, 2010). Research suggests it has been troublesome to determine the proper amount of risk activity needed to support PD success (Kutsch and Hall, 2010; Tchankova, 2002; Thompson and Perry, 1992). This research addresses the issue by presenting a methodology for estimating risk activity based on PD complexity.

Risk identification is perhaps the most important step in risk management, as undocumented concerns have little chance of being mitigated or controlled (Tchankova, 2002). To guide the identification of risks the literature has produced several taxonomies that can be used to highlight common areas of vulnerability in PD. Unfortunately, these tools have primarily been developed from software and IT projects with no unique taxonomies presented for vehicle development programs specifically. To address the needs of CoPS projects, this paper considers risk data from a major complex vehicle development project to determine which risk categories are appropriate. The results provide a preliminary framework of common risk sources that can be expanded with future, more extensive studies of risk data.

This research provides guidance in terms of both the quantity and types of risks that are appropriate to identify to support effective risk management in complex vehicle development projects.

Although several risk taxonomies are examined in this paper including by source, time frame, environment, etc., it is recommended that organizations take care to implement a framework that aligns with their program metrics or departmental responsibilities to ensure the data is most useful and actionable (Institute, 2008).

Historically risk literature has focused heavily on the mechanics of the risk process, rather than emphasizing the identification and mitigation of risks (Chapman, 2001). Effective risk identification requires more than mature and well-defined processes (Institute, 2008). It requires that appropriate elicitation techniques be employed throughout development, and implemented in a culture that is committed to documenting and resolving risks. Achieving the full benefits of risk management also requires support from top management to encourage risk identification strategies throughout the organization, and avoid such negative behaviors as deliberate ignorance and risk avoidance (Kutsch and Hall, 2010).

Although significant effort has been applied to improving risk management processes, maturity, and training, little research has focused on quantifying overall risk management effectiveness. This paper addresses the issue by proposing a method for measuring risk management effectiveness as a function of the number of risks identified, and the cumulative reduction of their risk severities. The method provides a means of performing evaluation and continuous improvement of the risk management process across projects in an organization.

CHAPTER 4: Managing Development Complexity for Complex Product Systems by Tailoring Risk Profiles of Design Concepts

I. Introduction

Today's advanced products are marked by increasingly complex subsystems and greater functionality (Kim and Wilemon, 2003; MacCormack, Verganti, 2001; Mihm, Loch, 2003). As a result, product development (PD) organizations have been struggling to develop these sophisticated products due to the uncertainty and risk they possess (Eppinger, Whitney, 1994; Kim and Wilemon, 2003; Morelli, Eppinger, 1995; Tani and Cimatti, 2008; Williams, 1999). Risk is present in any project that exceeds current capabilities and is compounded when these systems must be developed for less cost or with compressed schedules (Engineers, 2010; Harned, 2003). Studies indicate that nearly 85% of lifecycle[44] costs are locked-in after only 15% of detailed design is complete, which underscores the need for early coordination of risks (Kahn, 2005). Unfortunately, information during the fuzzy front end of PD is often unclear, chaotic, and highly uncertain (Kahn, 2005).

Organizations manage risk through information processing (Krishnan and Ulrich, 2001). Throughout the PD process, information is generated about design performance through prototyping, analysis, and measurement (AT&T, 1993; Tatikonda and Rosenthal, 2000; Thomke, 2003). Effective PD organizations leverage these activities throughout their process to aid in data collection and learning (Thomke, 2003). As the design progresses, more information becomes available, and the amount of uncertainty decreases, resulting in a reduction in risk (Browning, Deyst, 2002).

---

[44] Lifecycle costs include such elements as design, engineering, production, assembly, deployment, maintenance, and end-of-life collection and disposal activities (ATT, 1993)

The primary goal of PD is to identify a “recipe” that conforms to the requirements of the customer (Browning, Deyst, 2002). A critical aspect of PD success is the proper understanding of customer requirements and selection of the design solution (AT&T, 1993). Typically, there are several designs that can be developed as possible solutions and analyzed in terms of their comparative benefits and burdens (AT&T, 1993; Carr, 1993). Each design concept is considered for its level of compliance to customer requirements versus its overall cost (AT&T, 1993; Browning and Eppinger, 2002; Carr, 1993). By considering alternative design strategies, the PD organization is able to adjust the benefits vs. burdens of each option to find the highest value solution. In this process, the primary focus is on maximizing the performance vs. cost of the design without full consideration of the total risk and uncertainty of each alternative. This approach can result in organizations over committing resources into design concepts that are too complex or difficult to achieve. The concept yielding the highest return may also present the highest risk.

In this paper a methodology for establishing an early risk profile for design alternatives is presented in order to identify the optimal mix of design elements that will minimize development risk. The method extends the work of Browning, et al. (2002) that developed a method of quantifying requirements into performance risk values. Although Browning’s work addressed performance risk independently, this research extends his model to include assessments of performance, cost, and schedule risk simultaneously and provide a more robust risk profile. The assessment is also conducted at an earlier point in the development process to support early concept selection.

,The proposed method allows the PD team to minimize concept risk by selecting design elements with reduced risk profiles and thereby maximize the chance of PD.

The risk profile model draws from three separate areas of PD research, including: complexity, risk management, and product strategy decisions. The methodology is initiated based on customer requirements in order to evaluate risk at the earliest point in the development process, and provide a means for tying design decisions directly to risk metrics. Establishing a link between program requirements and risk allows the development team to directly manage the trade-offs that must occur between customer needs and performance uncertainty.

The proposed method is applied to a CoPS project to demonstrate its robustness in dealing with a high level of complexity. The definition of CoPS projects is consistent with Hobday's research which defines it as projects having limited-volume, a high degree of complexity and customization, and heavy focus on systems engineering and integration (Hobday, 1998).

The balance of this paper is organized as follows: Section II provides a review of current literature on risk management, PD complexity[45], and product requirements to better understand the context of the research. Section III outlines the method used for risk tailoring. A CoPS example is then presented in Section IV to demonstrate the process steps and results. Sections V and VI summarize the insights and limitations of the research, followed by the final conclusion(s) in Section VII.

II. Literature Review

In this section the relationship between program complexity and risk is explored to gain greater insight into how these elements can be managed throughout development. To establish a robust model, it is necessary to consider the areas of risk process management, complexity, risk assessments and technical decisions in PD. Understanding these areas will provide key insight

[45] For this research PD complexity is understood to encompass all the difficulties and uncertainties posed by the technology during the development, including consideration of the organization's tasks and people Kim, J. and D. Wilemon. 2003. Sources and assessment of complexity in NPD projects. *R&D Management*, 33(1): 15-30.

into quantifying program risk, tailor design concepts, and applying the method in the context of existing risk management processes.

A. *Risk Management Process*

Today's risk management processes employ many useful tools to facilitate the identification of risks including formalized processes and assessment methods involving mathematical probabilities, confidence intervals, and impact functions (Covello and Mumpower, 1985). Simple qualitative tools include SWOT analysis, influence diagrams, and cause-effect diagrams (Hulett, 2001). Still more quantitative tools exist to provide sophisticated risk analysis using decision trees, simulations, statistical analysis, and failure mode effects analysis (FMEA). These tools have proven to be effective in many industries where used consistently (Hulett, 2001).

Organizations with an aversion to risk tend to avoid uncertainty by emphasizing early controls of development activities (Nakata and Sivakumar, 1996). Based on studies by Johne (1984) the most experienced innovators use formal mechanism to track and control uncertainty in PD (Nakata and Sivakumar, 1996). However, this is not to say that complete risk avoidance should be the goal to ensure successful product development. In fact, survey results taken across multiple industries conclude that risk taking is actually a primary attribute for successful innovation (Nakata and Sivakumar, 1996). The key is found in Myerson and Hamilton's (1986) work which shows "proactiveness" and risk taking are correlated to successful PD (Nakata and Sivakumar, 1996). Suggesting the identification and planning for uncertainty in the early stages is the key (Ahmed, 2007).

An integral part of identifying and assessing project risks relates to the specific *risk taxonomy* being employed (Carr, 1993; Sarbacker and Ishii, 1997; Simpleman, 2006). Taxonomies based on risk origin (i.e. where was the risk generated from) have the benefit of being intuitive to the team based on their alignment with the organization, process, development phases, or department / specialties (Carr, 1993). However, organizing risks based their area of impact (i.e. cost, schedule, or performance) provides the benefit of being aligned with project metrics. For product-related DoD projects, risks are categorized based on their potential impact (Browning, Deyst, 2002; Browning and Eppinger, 2002; Simpleman, 2006). Chapter 3 of this dissertation provides a more extensive discussion of risk taxonomies.

Risk Assessment and Scoring: When adopting a taxonomy based on the DoD model, risks are scored using a five-point Likert scale based on their likelihood of occurrence, and consequences. Table 4-1 below shows a description of the common risk assessment criteria employed. Each risk is assigned a score for likelihood and consequence, such as 1-5, 2-4, 3-2, etc.

Likelihood			**Consequences**			
Score	**Prob**	**Severity**	**Score**	**Technical Performance**	**Schedule**	**Cost**
1	10%	Very little	1	Minimal or no impact	Minimal or no impact	Minimal or no impact
2	30%	Little	2	Minor reduction in performance	Able to meet key dates	< 5% over budget
3	50%	Moderate	3	Moderate reduction with limited impact	Able to meet key milestones with no float	5 – 7% over budget
4	70%	Significant	4	Significant degradation which may jeopardize program success	Program critical path affected	>7 – 10% over budget
5	90%	Severe	5	Key technical threshold will may jeopardize program success	Cannot meet key program milestones	>10% over budget

Table 4-1 - Risk Evaluation Criteria (Choi and Ahn, 2010; Simpleman, 2006)
(5 point Likert evaluation criteria is consistent with the DoD risk model and used extensively across industry)

A major criticism of this approach has been its use of an overly generalized scale for assessing risks (Choi and Ahn, 2010). Furthermore, the single-point measures for likelihood and consequence scoring would be more accurately represented by a probability distribution functions (PDF's) (Browning and Eppinger, 2002). Although the method depicted in Table 4-1 is simplistic, it has been deployed extensively and is the foundation of many risk management processes across industries—including defense, aerospace, and software (Carr, 1993; Simpleman, 2006). Choi and Ahn (2010) argue this method is limited because it offers only five classifications of scoring and cannot discriminate between small differences in factors (Choi and Ahn, 2010).

A major challenge of adopting more sophisticated analysis techniques for risk assessment is the availability of information (Johnson, 1997). In the early stages of PD when little information is known about an event, a triangular PDF estimating the best, worst, and most likely outcomes is often the most detailed prediction that can be provided (Johnson, 1997; Kotz and René van Dorp, 2004). In recent years, the triangular probability distribution has become standard for calculating likelihood assessments due to its simplicity and intuitiveness (Johnson, 1997; Ren é van Dorp and Kotz, 2002). It is currently employed extensively in Monte Carlo simulation modeling and various risk / uncertainty software such as @Risk and Crystal Ball (Kotz and René van Dorp, 2004). The triangular distribution has also been shown to provide comparable results for estimating accuracy when used as a proxy to the beta distribution (Johnson, 1997; Ren é van Dorp and Kotz, 2002). For this research, the common form of the asymmetric triangular density function as presented by Kotz and René van Dorp (2004) is referenced.

$$f(z|a,m,b) = \begin{cases} \frac{2}{b-a}\frac{z-a}{m-a}, & \text{for } a \le z \le m, \\ \frac{2}{b-a}\frac{b-z}{b-m}, & \text{for } m \le z \le b, \\ 0, & \text{elsewhere} \end{cases}$$

<u>Given</u>
a: worst case value
m: most likely value (mode)
b: best case value
z: actual value

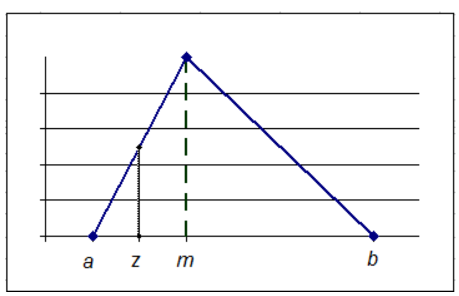

Figure 4-1: Triangular Distribution Function (Kotz and René van Dorp, 2004)
(Commonly used risk assessment technique to identify best case, worst case, and most likely outcomes)

Risk Mitigation: Although risks may be assessed with a high degree of severity initially, mitigation actions can be identified to decrease the severities to an acceptable level over time (McManus and Hastings, 2006). Mitigation strategies[46] focus on reducing the likelihood and/or consequence of the risk with actions that provide the best balance of cost vs. results (Simpleman, 2006). As the impact of the risk or its probability are reduced, the severity will also be reduced (Ahmed, 2007).

Risks mitigation[47] typically requires the investment of additional time and/or resources in the project, therefore it is important to identify these challenges during initial project planning (AT&T, 1993; Carr, 1993).

Early risk identification ensures mitigation activities are properly planned for and the maximum time and resources are available for mitigation. For acquisition organizations it is recommended to begin at the concept definition phase, to allow for handling through requirements modifications (Carr, 1993). In order to significantly affect lifecycle costs, risks must be identified and addressed in the earliest stages of design and development and continue throughout development as new situations arise (AT&T, 1993; Kayis, Arndt, 2006; Kim and Wilemon, 2003; MacDonell, 2002; Raz and Hillson, 2005).

B. Requirements and Customer Needs

Requirements create risk due to uncertainties associated with achieving design goals (AT&T, 1993). Understanding customer requirements is essential to assessing risk as the more challenging the performance threshold(s) are, the higher the risk of achievement will be (AT&T,

[46] Common risk handling strategies include: risk avoidance, risk transferring, and risk reduction. Risk avoidance may include such actions as product redesign, supplier resourcing, or decisions to not proceed a project or investment. Risk transfer is commonly practiced with insurance policies, fixed exchange rate negotiations, and general contract terms which transfer responsibility for a risk event to another entity Simpleman, L.M., Paul ; Bahnmaier, Bill ; Evans, Ken ; Lloyd, Jim. 2006. *Risk Management Guide for DoD Acquisition*. 148..

[47] Mitigation activities are specific actions targeted toward reducing the likelihood and/or consequence of a risk. Mitigation actions seek to minimize or potentially eliminate a risk's root cause or impact.

1993). Technical risk assessment should begin with the allocation of system requirements to functional areas. The decomposition of requirements addresses system complexity by establishing a preliminary functional architecture (AT&T, 1993; Browning and Eppinger, 2002; Group, 2010). System level requirements which are decomposed and allocated to subsystem teams are stated in increasingly more detail so they can be measured and verified at the subsystem level (AT&T, 1993). The requirements allocation process provides an indication of technical risk areas based on overall PD complexity (AT&T, 1993).

Interaction is a primary component of complexity and risk (Kayis, Arndt, 2006; Kim and Wilemon, 2003; MacCormack, Verganti, 2001; Mihm, Loch, 2003). Coupled requirements needing coordination between several groups often require increased effort and pose greater risks to system performance than decoupled requirements (AT&T, 1993; Suh, 1999). In some cases performance in one area can negatively impact requirements in other areas, calling for trade studies to be completed to find the optimal design balance (AT&T, 1993). Making design decisions which can decouple components and subsystems will simplify the design and reduce complexity as demonstrated in such methods as axiomatic design (Suh, 1999). Such design decisions are classified as product architecture or modularity decisions.

PD literature recognizes the need to establish requirements priorities due the limited resources available to achieve them (Karlsson, 1996). In highly complex projects there are typically a vast number of requirements and several performance targets that are in direct competition with one another (Curtis, Krasner, 1988). The situation necessitates a method of prioritization in order to resolve conflicts and focus limited resources. Unfortunately, despite the clear need for requirements priorities, a consistent and universal method has not yet been identified (Karlsson, 1996).

A common approach to establishing requirements priorities in complex projects is based on the importance of the function to system performance (Firesmith, 2004). Such prioritization helps to guide the system designers to ensure the most essential capabilities are maintained (Karlsson, 1996). In some cases where performance thresholds are in direct opposition to one another (such as power vs. fuel economy) it is sometimes necessary to forego (trade) one requirement in support of the higher priority requirement.

In DoD projects a common method of defining requirements priorities is through a generalized three-tier rating scale including:

- Tier 1: Requirements deemed “essential” to system performance. These represent the highest priority and are non-tradeable, allowing zero flexibility in achieving the threshold performance levels.
- Tier 2: Requirements with limited flexibility in threshold performance, and may be traded-off (i.e. not met) in order to meet higher Tier 1 priority goals when necessary.
- Tier 3: Requirements with the most flexibility. Defined as tradeable against Tier 1, Tier 2, and other Tier 3 requirements in order to optimize the overall system performance.

The DoD model is consistent with much of the research which suggests categorizing requirements based on how well they satisfy customer needs. One of the most recognized models for classifying customer preferences is the Kano Model (Figure 4-2), developed in 1984 (Chen and Chuang, 2008; Xu, Jiao, 2009). Using the Kano approach requirements are organized based on the three different levels of satisfaction they provide the customer including: (1) ‘must be’ requirements, (2) ‘one-dimensional’ requirements, and (3) ‘attractive’ requirements (Chen and Chuang, 2008; Sauerwein, Bailom, 1996). The ‘must be’ requirements are defined as ‘prerequisites’, and must be present or the customer will be extremely dissatisfied. However, because they fulfill a basic need, customer satisfaction will not increase as a result of them being there. Achieving ‘must be’ requirements can only result in the customer being ‘not dissatisfied’ (Matzler and Hinterhuber, 1998). The ‘one-dimensional’ requirements are defined as having a linear relationship with customer satisfaction. As these requirements increase in performance they provide increasing customer fulfillment. The ‘attractive’ requirements have the greatest influence on customer satisfaction but are not explicitly requested. The absence of ‘attractive’ attributes does not dissatisfy the customer, yet their presence in the product will delight (Sauerwein, Bailom, 1996).

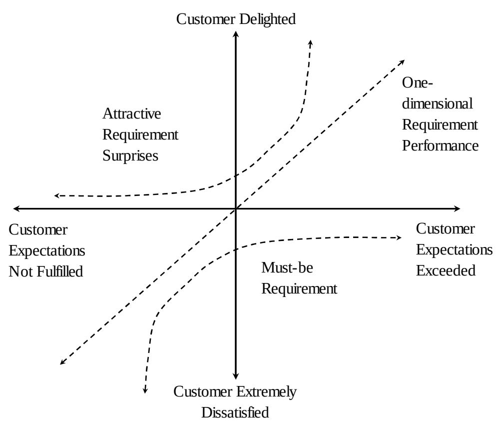

Figure 4-2: Kano Model (Berger, 1993)
(Several customer satisfaction models are consistent with the Kano model)

The Kano model helps PD organizations to prioritize requirements by determining which capabilities should be developed further in order to maximize customer satisfaction (Chen and Chuang, 2008; Sauerwein, Bailom, 1996). Such information provides an effective method for guiding requirements trades in the design (Chen and Chuang, 2008). A significant contribution of the Kano model is its generalized use of ‘*utility*’ curves. The Kano model extended the concept of simple classifications / grouping by providing a visual indicator of how customer satisfaction is generally impacted along the entire range of performance for each attribute.

Several studies conducted after the Kano model employ similar strategies for grouping requirements. A common practice for complex software projects has been to group requirements in terms of: (1) essential capabilities, (2) useful capabilities, and (3) desirable capabilities, respectively (Firesmith, 2004). This classification scheme is also consistent with the defense industry's practice of tiering requirements as critical, major, and minor (1, 2 and 3 respectively). In practice, establishing the type of requirements categories to be used has been far less difficult than determining the actual rank ordering of the individual requirements (Firesmith, 2004). While smaller projects have successfully used traditional methods for rank ordering requirements such as Analytical Hierarchy Process (AHP), Multi-attribute Utility Theory (MAUT), Quality Function Deployment (QFD), etc. (Firesmith, 2004; Karlsson, 1996), it is often infeasible for highly complex systems to employ these more sophisticated methods due to the large number of requirements to be addressed. In such cases the simpler method of grouping requirements of like priorities has been used, based on the consensus of key stakeholders (Firesmith, 2004; Sauerwein, Bailom, 1996).

C. Managing Uncertainty

Uncertainty in the achievement of program goals creates *risk*, and is closely associated with complexity (Kim and Wilemon, 2003) reference Figure 4-3. Complexity is a key contributor to task uncertainty[48] which can negatively impact project execution. New technologies create ongoing challenges in PD which have regularly led to launch delays and cost overruns (Tatikonda and Rosenthal, 2000). By developing methods to reduce program risk, organizations will be better able to manage complexity and vice-versa (AT&T, 1993; Kim and Wilemon, 2003; Tatikonda and Rosenthal, 2000).

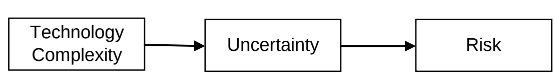

Figure 4-3: Relationship of Technology Complexity to Risk
(Complexity contributes to task uncertainty which creates risk)

[48] Task uncertainty is the difference between the required amount of information needed to complete a task, and the amount of information possessed by the organization (Tatikonda, 2000)

Webster defines risk as “the possibility of loss or damage” which highlights its two key elements of *uncertainty* and *consequence*. Integrating the components of complexity and risk reveals that uncertainty is a shared element as shown in Figure 4-4 below (Kim and Wilemon, 2003).

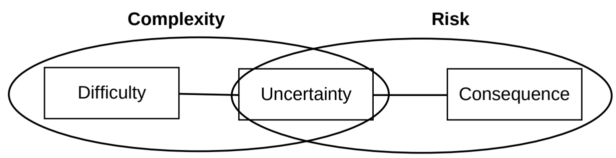

Figure 4-4: Complexity vs. Risk
(Uncertainty is a shared element between risk and complexity)

The relationship indicates that a reduction in uncertainty will result in a reduction of both the complexity and risk of the project.

A primary goal of risk management is to determine how much risk an enterprise is willing to accept (Steinberg, Martens, 2004). To accomplish this, a method for framing and consolidating risks into one summary is needed. Research has attempted to address this by showing that total performance risk (R_p) or cost risks (R_c) can be represented as a function of the individual risks as shown by the equation:

$$R_p = \sum_{0}^{n} r_n \quad where\ n = individual\ performance\ risks$$

$$R_c = \sum_{0}^{n} r_n \quad where\ n = individual\ performance\ risks$$

Although most modern assessments of project risk are based on subjective evaluations due to uncertainty, there is still a need to quantify risk for effective PD planning and execution (Browning, Deyst, 2002; McManus and Hastings, 2006). Browning (2002) proposed a quantitative method for assessing and tracking program risks using Technical Performance

Measures[49] (TPM's) (Browning, Deyst, 2002). In Browning's model the program's total performance risk (R) is determined by summing together the individual risk assessments for each TPM (R_{tpm}). $Risk\ (R)$ is defined as $Likelihood\ (L) * Consequence\ (C)$ (Ahmed, 2007). To generate the risk probabilities a triangular distribution function (PDF) is generated for each TPM by identifying the lowest, highest, and most likely performance values for the requirement--an approach that is consistent with contemporary practices for risk assessment (Johnson, 1997). The method then applies a utility curve for each TPM to determine its value (utility) at each performance level (AT&T, 1993; Group, 2010).

To calculate risk as a function of $L * C$, the area of the PDF curve falling below the minimum acceptable performance level is multiplied by the utility curve (C) to represent the risk (R_{tpm}). A weighting criteria (W_i) is then used to prioritize each TPM so the importance of each requirement is reflected in the overall risk assessment.

Risk assessments are based on predictions that change over time as new information becomes available (Browning, Deyst, 2002; Simpleman, 2006). Browning's model is designed to be iterated throughout the development process to update TPM risk status. The method provides an effective means for quantifying and tracking risk of to an established design with well defined performance targets. However, to be effective for use during concept selection the model should provide a means of assessing all types of risks concurrently (i.e. performance, schedule, and cost) to address the trade-offs that are necessary for the design. This paper extends Browning's research by adapting the scoring approach so it can be applied to all types or risks concurrently, and be utilized for concept selection and early risk tailoring.

---

[49] Technical Performance Requirements (TPM's) are the key performance requirements for a program and collectively provide an indication of the customer's overall system performance needs such as payload, top speed, weight, etc.

Risk Trading: In the subsequent work of Browning and Eppinger (2002) they explore the concept of trading-off risk types (cost vs. schedule) using a *process* architecture example. In their research, alternative process architectures are defined and evaluated (via simulation) to determine which of them provides the most acceptable risk profile in terms of cost and schedule (Browning and Eppinger, 2002). The model provides a novel way to predict process efficiency upfront, while demonstrating an effective method for trading between two risk types (Browning and Eppinger, 2002). Plotting the total cost risk vs. schedule risk for each alternative provided a 'trade-off frontier' (curve) that could be used in determining an acceptable range of alternatives for the two dimensions (Browning and Eppinger, 2002) – reference Figure 4-5.

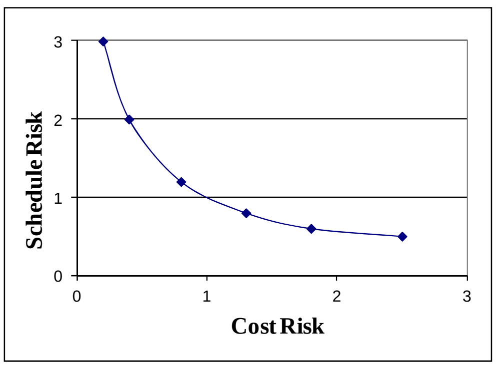

Figure 4-5: Cost vs. Schedule Risk (Browning and Eppinger, 2002)
(Risk types can be traded based on project needs)

Although the method presented by Eppinger and Browning is applied to an engineering *processes* example, the idea of comparing multiple design concepts (through process architectures) to trade-off risk types (cost vs. schedule) is demonstrated. The methodology is also consistent with contemporary risk management methods which employ the use of triangular PDF's to reflect cost and schedule uncertainties rather than a lesser accurate single point estimate for risk likelihood (Browning and Eppinger, 2002). Using this model, the total cost risk -vs- schedule risk can be calculated as a function (curve) along the entire range of outcomes (Browning and Eppinger, 2002). This method provides a more comprehensive assessment of the total risk than previous studies by including both *cost* and *schedule* risk impacts, however, it does not provide a means of incorporating *performance* risk into the same assessment. It is here that additional research is needed, as all three risk types are significant and provide input to concept selection decision. To be effective the method must also be applicable to *product* architecture assessments, as these are often the primary drivers for generating design concepts in CoPS projects (AT&T, 1993).

PD Decisions & Risk: Throughout product development, decisions are made which increase or decrease program risk (Browning, Deyst, 2002). Kim and Wilemon (2003) examined such cases in their study of PD complexity and its negative impact on development projects (including late delivery, over budget, under performance, etc). (Kim and Wilemon, 2003). Since design decisions affect the risk levels of a project (increasing or decreasing) we recognize that they present a significant opportunity to exercise ‘control’ of complexity and risk.

Technical risk originates from customer requirements so it is necessary to determine a method of handling them through design strategies (Kim and Wilemon, 2003). In 2001 Krishnan and Ulrich completed an extensive review of product development literature which included an analysis of over 400 articles recommended by 50 scholars across the field of PD (Krishnan and Ulrich, 2001). After reducing the literature to a working list of 200 papers the authors identified a recurring set of key decisions that are routinely made within PD projects (reference Appendix 2). The decisions involved a collection of issues from such areas as concept development, supply chain, product design, testing, and production (Krishnan and Ulrich, 2001). Although the research was conducted across a broad range of industries employing different PD processes, the authors observed that the *type* of decisions made remained fairly consistent (Krishnan and Ulrich, 2001). Using these decisions as *levers* to manage PD complexity can reduce risk.

Product Architecture and Coupling: Product architecture has an impact on complexity and defines how the function of the product is carried-out by its components (Ulrich, 1995). Because a product's functionality is separate from its physical make-up, there are several ways in which a product can be structured and still maintain necessary operation. Ulrich (1995) provides an example of three common architectural topologies that can be applied to products to organize / define their essential functions. In his research he applies the topologies to simple products including a desk, computer, and trailer which could employ any of the three PA's (slot, bus, and sectional) yet still meet the essential functions (Ulrich, 1995). While Ulrich asserts that no single product architecture is optimal in all cases, he suggests that organizations take care to choose the best PA strategy for their needs, particularly when trying to minimize technical risk (Ulrich, 1995).

To develop the optimal product architecture the goal is to group components to maximize the interaction between related / internal elements, and minimize the links (or coupling) required to other (external) elements (Campagnolo and Camuffo, 2010; Gokpinar, Hopp, 2010; Ulrich, 1995). Ulrich defines a coupled component as one that cannot be changed without changing the component(s) it's attached to. He suggests coupling is something to be avoided. Sosa, et. al (2003) builds off of this research by considering the level of modularity vs. integration in a product architecture and how it affects PD performance (Sosa, Eppinger, 2003; Sosa, Eppinger, 2004).

Modularity can be measured by the number of physically coupled or interacting elements. Modular systems are the opposite of *integrated* systems, which contain many design interfaces across many systems elements, forming a functionally distributed model [26]. The concept behind modularity is to "break-up complex systems into fewer discrete pieces that can then communicate with one another through standardized interfaces within a standardized structure" (Langlois, 2002). Sosa, et al. (2003) suggest organizational coordination across modular systems (for example between separate departments) requires more management effort than integrated coordination (which occurs within a department) (Sosa, Eppinger, 2003). Therefore, in managing coordination risk it is essential to establish the most appropriate modularity which will influence process architecture and drive the organizational interactions (Browning and Eppinger, 2002).

Technology Novelty and Maturity: Novelty is one of the primary contributors to complexity (Hobday, 1998; Novak and Eppinger, 2001). Novelty / newness can be measured in a number of ways including: unique capability, design approach, components material / technology, or integration of elements (Kim and Wilemon, 2003; Novak and Eppinger, 2001). New and innovative technologies introduce requirements risk because there is uncertainty in their development and performance (Smith, 2005).

Ensuring the successful incorporation of innovative products requires a process for managing the maturity levels of the technologies being developed (Mankins, 2002). To address this need NASA established a formal method for the assessing technology readiness levels (TRL) within complex projects in the 1980's (Mankins, 2002; Sauser, Verma, 2006). The TRL scales have been adopted by many organizations including both government and commercial and include nine levels of maturity as shown in Figure 4-6 below:

<table>
<tr><th colspan="2">TRL Definition</th><th colspan="5">Maturation Process (NASA)</th></tr>
<tr><td>1</td><td>Basic principles observed and reported</td><td rowspan="2">Basic R&D</td><td></td><td rowspan="2"></td><td rowspan="5"></td><td rowspan="6"></td></tr>
<tr><td>2</td><td>Technology concept and application formulated</td><td rowspan="4">Establish Technology Base</td></tr>
<tr><td>3</td><td>Analytical experimental critical function and/or characteristic proof-of-concept</td><td rowspan="7"></td><td rowspan="5">Capability Focus</td></tr>
<tr><td>4</td><td>Component and or breadboard validation in lab environment</td></tr>
<tr><td>5</td><td>Component and or breadboard validation in relevant environment</td></tr>
<tr><td>6</td><td>System/subsystem Model of prototype demonstration in relevant environment</td><td rowspan="4"></td><td rowspan="3">Advanced Development</td></tr>
<tr><td>7</td><td>System prototype demonstrated in relevant environment</td><td rowspan="3">Flight Projects</td></tr>
<tr><td>8</td><td>Actual system completed and qualified through test and demonstration</td><td rowspan="2"></td></tr>
<tr><td>9</td><td>Actual systems proven through successful mission operations</td><td></td></tr>
</table>

Figure 4-6: TRL Definition and Maturation Process
(adapted from Mankins (2002) and Sauser (2006))

Although the TRL process can only provide a measure of individual maturities (not system level), and does not reflect integration difficulty, it does provide an indication of development risk (Sauser, Verma, 2006). However, its application as a risk assessment tool is limited because of the broad classifications of each TRL level. Because each subsystem / component will have unique risks associated with moving from one TRL level to the next, there is little utility in the classification scheme as detailed risk quantifying tool (Sauser, Verma, 2006). However, the TRL process has been used successfully for many years to provide a common language and generalized understanding of technical development maturity between customer and developer (Smith, 2005).

Requirements challenging the state-of-the-art with high performance thresholds, or new technologies should be identified as high risk areas, and will likely require more effort to coordinate, develop, and validate (AT&T, 1993). Conversely, decisions made to employ established technology will reduce risk because the development and performance capabilities are known. Selecting available components which are already in production is a common way to minimize development risk through maturity (Mankins, 2002). Unfortunately, CoPS projects often require highly tailored components and/or unique materials for their applications to achieve the needed performance. These items can often become long-lead items (Hobday, 1998) and create immediate schedule risk to the program. In such cases additional coordination effort is needed to ensure appropriate suppliers have been selected, timely communication with the manufacturers is occurring, and the materials are being fabricated on schedule (as they are often on the critical path with little to no room for delay) (Chan and Kumar, 2007).

Sourcing Risk: In today's competitive climate, coordination with key suppliers is essential for success (Chan and Kumar, 2007). Supplier and outsourcing decisions generate risk for organizations in terms of higher costs, diminished performance, and longer lead times. (Benoit, Patry, 2001). For aggressive projects already limited in time and resources any supplier disruption can create significant program risk (Chan and Kumar, 2007). Over the last several decades considerable research has been devoted to supplier selection techniques. In reviewing key decision criteria, Chan and Kumar (2005) suggest that supplier profiling should play a primary role in the source selection by including such factors as: financial status, performance history, and facility capacity (Chan and Kumar, 2007). Product strategy should also be considered a key determinate in supplier selection since innovative technologies (as used in CoPS projects), require additional flexibility (Fisher, 1997). Often early development programs require quick iterations of design changes, and expedited deliveries to meet aggressive program schedules. Choosing the appropriate supplier can significantly impact the risk profile of a these projects.

Based on the literature several PD variables have been found that impact risk. Table 4-2 (below) provides a list of the more common variables with the risk measurements shown at each extreme. When risk variables are described by the measures listed in 'low risk' column (i.e.

established technology, short lead time, high team competency, etc.), the overall development risk will be reduced. Conversely, as risk variables begin taking on the values to the right such as with new technology, long lead times, and high integration, the development risk will be increased. Product development teams seeking low-risk solutions should generate concepts maintain risk variables in the low-risk range.

Risk Variables	Low Risk	High Risk	Reference
Technology (novelty & maturity)	Established	New	(AT&T, 1993; Kim and Wilemon, 2003; Smith, 2005)
Lead time	Short	Long	(Kahn, 2005; Meyer and Utterback, 1995)
Coupling / integration	Uncoupled	Highly coupled	(Kahn, 2005; Meyer and Utterback, 1995)
Requirements priority	Tier 3	Tier 1 / TPM	(AT&T, 1993; Browning, Deyst, 2002)
Material (cost & geometry)	Low	High	(Kahn, 2005)
Team competency / effectivness	High	Low	(Baccarini, 1996; Carr, 1993; Kahn, 2005)
Early modeling capability	Extensive	Limited	(Thomke, 2003),(Kahn, 2005)
Commonality	Low	High	(Krishnan and Ulrich, 2001)
Sourcing	External	Internal	(Kahn, 2005; Krishnan and Ulrich, 2001)
Process steps/ hand-offs	Fewer	Many	(Krishnan and Ulrich, 2001)

Table 4-2: Product Development Risk Variables
(Projects possessing variables with higher risk have greater uncertainty)

The review of the literature found that complexity and risk share a common element of uncertainty (Kim and Wilemon, 2003). Eliminating uncertainty through early risk assessment is essential for product development success (Nakata and Sivakumar, 1996). While several risk process and maturity models have been proposed, there remains no universal method for how risk should be executed (Conrow and Shishido, 1997). The next section outlines a method that integrates several of the promising techniques reviewed in the literature for early identification, quantification, and tailoring of design risk.

III. Proposed Methodology

The method developed extends the work of Eppinger and Browning (2002), which demonstrated that risk types can be traded-off (i.e., cost vs. schedule) to generate a risk profile curve for alternative process architectures. This paper expands that concept by including all three types of risk impacts (cost, schedule, and performance) in the trade model concurrently to provide a complete evaluation of risk. Maintaining this taxonomy provides consistency with current industry practices as outlined by the DoD and consistent with CoPS projects (Simpleman, 2006).

The process for generating the complete risk profile model is shown in Figure 4-7 below.

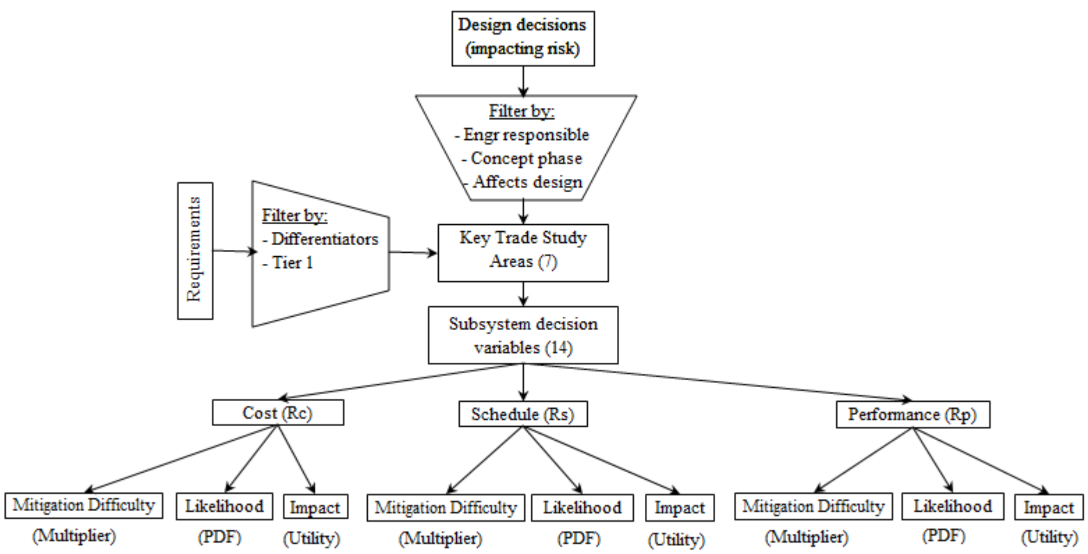

Figure 4-7: Risk Profiling Model
(The risk profiling model evaluates risk to schedule, cost, and performance concurrently)

The process begins by filtering and prioritizing the list of customer requirements to determine the performance goals that need to be emphasized in the concept. The requirements' filtering serves to reduce the number of design concepts generated so the solution space remains at a manageable size.

In practice, the specific filters may vary based on the organization's strategy, and the needs of the program. After identifying the key requirements, the subsystem team(s) filter the list of design decisions to determine where opportunities exist to generate competing concepts.

For this study, the three criteria used to filter the requirements list include:

1. Requirements that are mandatory and non-tradeable to the customer.
2. Requirement focusing on the company's differentiating capabilities vs. competitors.
3. Requirements that encouraged multiple unique design solutions.

The literature indicated that several key drivers of technical risk are common in product development activities (reference Table 4-2). Based on research by Krishnan and Ulrich (2001) thirty-four design decisions are commonly made throughout the PD process (reference Appendix 2). If these decisions are confirmed to impact risk, they can be used as a basis for assessing and generating an early risk profile.

Risk PDF Functions*:* Leveraging established techniques for risk scoring, the total program risk is calculated using triangular PDF functions and utility curves for each of the three risk areas concurrently (i.e. performance, cost, and schedule). This is consistent with Browning's method for calculating risk for tracking TPM's (Browning, Deyst, 2002).

Risk is defined as the probability of an adverse outcome, or:

$$Risk\ (R) = Likelihood\ (L) * Consequence (C)$$

For a requirements-based assessment, the risk would be the *likelihood* of not meeting the requirement multiplied by the *consequence* of not meeting the requirement. To illustrate, based on the scoring criteria shown in Figure 4-8, a value of '4' indicates an "at threshold" condition, which is the minimum acceptable value for the requirement. Any performance below that threshold value (4) represents an adverse outcome, and therefore risk--this area is shown in gray shading in Figure 4-8.

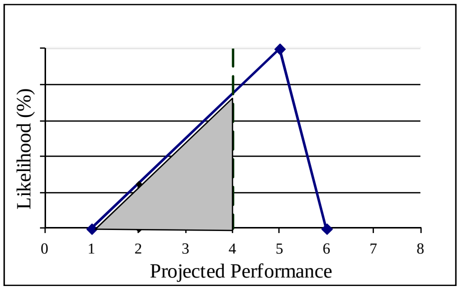

Figure 4-8: Requirements Performance Distribution Function
(Performance risk is represented by the gray shaded area--the probability of below target performance)

The *consequence* (or impact) of having a less than the threshold performance will depend on the customer's needs. For requirements that are deemed '*must haves*', any value below the threshold will be unacceptable, and return zero utility. For requirements that are '*one-dimensional,*' increases in performance (beyond threshold) will increase customer satisfaction.

In order to quantify project risk, a utility curve must be generated for each requirement to indicate the *impact* of performance levels. Unfortunately, in the concept development stage, generating this level of detail is often infeasible due to the limited time and information available. An alternative method would be to develop a finite number of generic utility curves based on the classifications of requirements. For example, unique utility curves can be generated based on tier ratings (e.g. 1, 2, or 3), or related customer preference models such as the Kano model (e.g. must-have, one-dimensional, critical, etc.). Once the product team decides on the requirements classifications, an appropriate utility curve can be developed for each group.

A common practice in design trade analysis is to measure requirements compliance *utilities* based on four performance levels as indicated in Figure 4-9.

Description	Utility
Above threshold	1.0
At threshold	.8
Slightly below threshold	.5
Well below threshold	.2

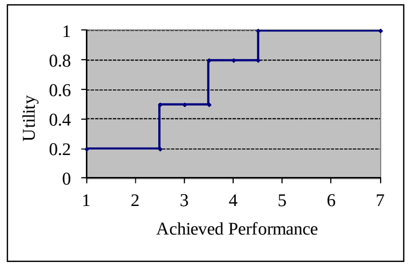

Figure 4-9: Common Requirements Utility Curve
(Utility curve indicates the customer values below threshold performance)

The measurement scale for these situations would be non-linear, and assume a penalty of .3 if performance slips from 'threshold' to 'slightly below threshold'. This utility function is indicative of a customer providing the opportunity for partial credit on performance close (but below) the threshold. Performance that exceeds the threshold would earn additional utility of 0.2 (i.e. 0.8 to 1.0). In trade study analysis, such a utility function guides the decision makers to focus first on threshold performance, and secondarily on exceeding the threshold.

***Remediation Difficulty**:* Although the initial risk of a design alternative may be high, PD organizations recognize that various mitigation strategies may be possible to reduce the risk to lower levels. In some cases, mitigation may be applied to some risks with minimal program effort or cost, even when the initial risk assessment is high. In such cases, PD teams may be more apt to pursue the higher risk design with plans to mitigate the risk in the future. Therefore, the anticipated difficulty of mitigating a risk is a key variable PD teams consider in their design decisions. To capture this element of the risk assessment, a multiplier called *'remediation difficulty'* (D_i) has been included in the model for each design alternative and risk type (performance, cost, and schedule).

Total Risk Calculation: Integrating the risk *likelihood*, *consequence* and *remediation difficulty multiplier* yields the following equation for total performance risk[50]:

$$R_p = D_p * \int_{-\infty}^{T_h} f_{L_p}\ (X_0) * \left[U_{T_h}(T_h) - U_{T_h}(X_0)\right] dX_0$$

where:

$L_p = Likelihood\ of\ performance$
$T_h = Threshold\ of\ performance\ as\ defined\ by\ the\ requirement$
$X_0 = Actual\ performance\ levels$
$U = Utility\ function$
$D_p = Difficulty\ in\ mitigating\ risk$

An equivalent formula is derived for both the cost risk *R(c)* and schedule risk *R(s)*, which are summed to provide a total risk profile (R_T) equal to:

$$R_T = \sum[W_p * R_p + W_c * R_c + W_s * R_s]$$

where W_p, W_S, *and* W_c *equal weighting criteria for each risk type*

While it is understood that probabilistic functions such as risk are not additive by nature, the assessment of total risk (R_T) in this research is intended to provide a *relative* measure of riskiness between design concepts to provide the PD team with a broad perspective of the challenges and uncertainty that exist with different options. The method is not intended to provide an *absolute* value of project risk, but a relative assessment between design concepts.

[50] This form of the risk equation was suggested by Browning, et al. (2002) for TPM risk calculations.

IV. Case Analysis Example

The process began with a validation of the method proposed, followed by a filtering of requirements and filtering of key requirements to highlight those most essential to success. A total of seven requirements areas were selected as shown in Table 4-3 (below).

Validating Decisions Affecting Risk: Working with SME's to cross-reference Krishnan and Ulrich's research with the list of risk variables from this research (Table 3-2) revealed that each of the ten variables are impacted by one or more of the design questions. Therefore, by guiding the design to reference the design questions for each of their alternatives, the proposed risk variables will be considered.

Validating Risk Variables for CoPS Proejcts: To provide confidence that the risk variables were relevant for CoPS products, historical data of design trade studies from a large-scale DoD project were analyzed. The historical data covered a 20-month development cycle that yielded 87 separate trade studies ranging in complexity from component-level to system-level trades. The SMEs and trade study analysts reviewed each of the trade studies for impact on the ten design risk variables (Table 4-2). Based on the analysis, each of the design decisions was found to impact at least one of the risk variables identified, indicating the risk variables are relevant to CoPS projects (reference Appendix 1). In order to maintain the confidentiality of the development program the design trade study data has been withheld.

Filtering: To expedite the risk analysis Krishnan and Ulrich's was filtered for PD decisions impacting: (1) the concept development phase, (2) the physical design of the system or components, and (3) those allowing engineering to have lead responsibility for decisioning. The finished list was reduced to ten decisions.

Key Trade Studies: After considering the relevant design decisions for each of the seven requirements areas, the team generated alternatives for each of the trade decisions. The resulting design concept affected five of the ten risk areas (reference Table 4-3).

The seven trade decisions were used as the foundation for the risk concept generator and included the two most promising alternatives for each subsystem, yielding 14 separate decisions variables. A total of 128 vehicle configurations were possible[51] $(2 \text{ alternatives})^{7\ (\text{subsystems})}$ given the alternatives. Once identified, each of the fourteen design alternatives was analyzed in terms of its cost, schedule, and performance risk by the appropriate subject matter expert(s).

The trade study list is shown in Table 4-3 below which includes: affected subsystem area, key requirements focus, relevant design decision made, and risk variable(s) impacted. A description of each of the 14 decision variables is also included in the last column.

[51] Not all possible configurations were feasible due to incompatibilities of design elements. To address these issues compatibility constraints were added to the model.

Subsystem Area	Requirement Focus	Design Decision and Risk Variable Impacted	No.	Decision variable
Powertrain	*Mobility requirements including engine power and fuel economy*	*Technology (Novelty): Alternatives employing different technologies with significant performance capabilities and development maturity*	*1*	*Hybrid*
			2	*Gas*
Electrical	*Electrical (man-to-machine) interfacing and reliability requirements*	*Coupling / integration: Alternatives performing the same function with varying levels of integration / component interaction*	*3*	*Integrated Software features*
			4	*Hardwired features*
Interior	*Mobility requirements including engine power and fuel economy*	*Technology (Maturity): Alternatives using same basic technologies but with varying levels of performance and fielding*	*5*	*Available displays*
			6	*New displays*
Chassis	*Mobility requirements including engine power and fuel economy*	*Material: Similar solutions employing different material types (e.g. high performance material vs. standard material)*	*7*	*Steel alloy*
			8	*Titanium*
Structure	*Mobility requirements including engine power and fuel economy*	*Material: Alternatives employing the same material but with unique geometries to address performance requirements.*	*9*	*Angled*
			10	*Flat*
Structure	*Transportion weight requirements vs. other performance features*	*Requirements priorities: Alternatives are targeted toward achieving different performance needs (transport vs. additional capability)*	*11*	*Weight reduction*
			12	*Additional functionality*
Electrical	*Mobility requirements including engine power and fuel economy*	*Sourcing: Alternative suppliers for parts / subsystems performing the same basic function, or targeting the same performance goals*	*13*	*New source*
			14	*Existing*

Table 4-3: Trade Study List

(Fourteen decision variables representing the seven affected subsystem areas)

An analysis of the decision variables for each subsystem area indicated that compatibility issues existed for certain combinations of elements. For example, the hybrid engine geometry would not allow it to be packaged with the angled structure without compromising structural performance and the available displays could not accommodate the integrated software features due to technology limitations. Therefore, design constraints were established to ensure these alternatives were not selected together.

Employing a 7-point Likert scale for each of the 14 alternatives, the responsible subsystem team(s) provided an assessment of risk in the three areas of performance, cost, and schedule for each design alternative. The scoring was done using a triangular PDF function indicating the best-case, worst-case, and most likely outcomes for each risk area. A description of the performance scoring metrics is shown in Table 4-4 below.

Score	Description	Projected Mean Performance
7	Well ABOVE Threshold	~20% or more
6	Moderately above Threshold	~10%
5	Slightly above Threshold	~5%
4	At Threshold	0
3	Slightly BELOW Threshold	~5%
2	Moderately BELOW Threshold	~10%
1	Well BELOW Threshold	~20% or more

Table 4-4: Performance Scoring Metrics for Design Alternatives
(Team derived 7-point scale for performance assessments of each design alternative)

Utility Function: The design team determined that all seven design decisions affect non-tradeable (Tier 1) requirements, with no anticipated partial credit for performance falling below the threshold level, and no extra credit for performance exceeding the threshold. This is not unusual for competitive down-select contracts, where contractors are scored in terms of how many requirements are met, and over-achievement of requirements can be viewed as over-designing which adds cost and timing to the contract. The function used is as shown in Figure 4-10.

Description	Utility
Above threshold	1.0
At threshold	1.0
Slightly below threshold	0
Well below threshold	0

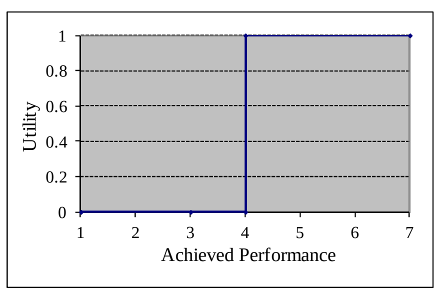

Figure 4-10: Tier 1 Requirement Utility Curve
(Utility curve indicates zero customer value for below threshold performance)

Remediation Difficulty Multiplier: The remediation difficulty multiplier was assessed by the SMEs based on their design knowledge and experience. The scoring outline in Table 4-5 indicates the range of values used for the remediation variable. Note that the values have been normalized with a 1.0 as the highest possible score—which would indicate no remediation is anticipated. The scoring is based on the concept that an absence of risk mitigation will not cause the risk likelihood or consequence to increase. The highest a risk value can be without mitigation is the same value, so it is therefore multiplied by 1.

Score	Description
1.0	Extremely difficult to mitigate, if at all. Anticipated high cost / time of mitigation. Clear mitigation actions may not be known
0.8	Difficult to mitigate, however, some mediation can be accomplished. May incur medium to high mitigation cost / time
0.5	Moderate effort to mitigate, with low to medium projected cost associated. General mitigation approach is known
0.2	Easy to mitigate, with minimal to no cost for mitigation. Needed mitigation steps are known
0	No significant risk identified

Table 4-5: Remediation Difficulty Multiplier Scoring Descriptions
(Highest score of 1.0 indicates no mitigation is likely and the entire risk assessment will be carried forward)

Applying the above scoring criteria to the CoPS project example resulted in the risk *consequence* data shown in Table 4-6 below (for each of the fourteen options). All input data for performance risk, cost risk, schedule risk, and their associated difficulty multipliers have been included.

The total risk profile R_T has also been calculated for each design option, based the individual risk scores ($R_p,\ R_c, and\ R_s$) and their assigned weightings. Risk scores at the extremes of 0.0 or 1.0 indicate all projected outcomes (best, worst, most likely) fall above or below the performance threshold target(s). A score of 1.0 indicates the team expects with near certainty that the threshold will not be met, while a score of 0 indicates a negligible amount of risk is present in meeting the goal.

Applying the Tier 1 utility curve and weighting values from Figure 4-10 to the likelihood data results in the R_T scores shown on the far right column of Table 4-6. The R_T has been normalized with a 0 indicating no risk, and a 1.0 indicating the highest risk level. The remediation multiplier has also been normalized with a maximum value of 1.0 (100%) as the worst case would result in no mitigation action being achieved, and the entire risk value remaining. In the absence of risk mitigation, the highest the risk would be is the same value.

		Performance			*Dp*	*Rp*	Cost			*Dc*	*Rc*	Schedule			*Ds*	*Rs*	R_T
Options	Weight	Likely	Worst	Best		**0.4**	Likely	Worst	Best		**0.3**	Likely	Worst	Best		**0.3**	1.0
1 - Hybrid	0.2	7	6	7	0.0	0.0	1	1	2	0.8	0.8	2	1	4	1.0	1.0	0.540
2 - Gas		4	3	5	1.0	0.5	6	4	7	0.0	0.0	5	4	7	0.0	0.0	0.200
3 - SW switches	0.05	4	4	5	0.0	0.0	2	1	4	1.0	1.0	5	4	5	0.0	0.0	0.300
4 - HW switches		4	4	5	0.0	0.0	4	3	5	0.5	0.3	4	3	4	0.2	0.2	0.135
5 - Existing Display	0.1	3	2	4	0.8	0.8	4	3	5	1.0	0.5	4	4	6	0.0	0.0	0.470
6 - New Display		6	4	7	0.0	0.0	2	1	4	0.8	0.8	3	1	4	1.0	1.0	0.540
7 - Titanium	0.1	6	4	7	0.0	0.0	1	1	1	1.0	1.0	3	2	4	1.0	1.0	0.600
8 - Steel		4	3	4	0.8	0.8	4	2	4	0.5	0.5	4	4	5	0.0	0.0	0.470
9 - Angled geometry	0.25	5	4	6	0.0	0.0	3	2	4	0.8	0.8	3	3	4	1.0	1.0	0.540
10- Flat geometry		3	2	4	1.0	1.0	5	4	6	0.0	0.0	5	4	6	0.0	0.0	0.400
11- Weight reduction	0.1	4	4	5	0.0	0.0	4	3	5	0.8	0.4	4	4	5	0.0	0.0	0.120
12- Add'l functionality		1	1	1	1.0	1.0	7	7	7	0.0	0.0	7	7	7	0.0	0.0	0.400
13- New source	0.2	4	3	4	0.8	0.8	5	5	6	0.0	0.0	3	3	4	1.0	1.0	0.620
14- Current source		4	4	5	0.0	0.0	3	2	4	0.8	0.8	4	4	5	0.0	0.0	0.240

Table 4-6: Risk Consequence Scoring
(Cost, schedule, and performance risk estimates for fourteen decision variables)

Using this data, several low-risk design concepts employing a linear combinatorial optimization model are generated.

$$Minimize: R_T = \sum k_i * [W_p R_p i + W_c R_c i + W_s R_s i] + k_j * [W_p R_p j + W_c R_c j + W_s R_s j]$$
$$for\ n\ decisions$$

__Where:__
R_T = *Total Project Risk*
R_p = *Performance Risk*
R_c = *Cost Risk*
R_s = *Schedule Risk*
$k_{i,j}$ = *Binary (decision variables for alternatives 'i' &' j')*
W_p = *Weighting for Performance Risk*
W_c = *Weighting for Cost Risk*
W_s = *Weighting for Schedule Risk*

$Lp_{i,j}$ = *Likelihood of performance risk i, j*
$Ip_{i,j}$ = *Impact of performance risk i, j*
$Lc_{i,j}$ = *Likelihood of cost risk i, j*
$Ic_{i,j}$ = *Impact of cost risk i, j*
$Ls_{i,j}$ = *Likelihood of schedule risk i, j*
$Is_{i,j}$ = *Impact of schedule risk i, j*
C_n = *Constraint of decision 'n'*

Subject to:
$k_i + k_j = 1$, *for n = 1- m decisions*
$W_p + W_c + W_s = 1$

$$Rp_{i,j} = \sum \int Lp_{i,j} * \int Ip_{i,j}$$
$$Rc_{i,j} = \sum \int Lc_{i,j} * \int Ic_{i,j}$$
$$Rs_{i,j} = \sum \int Ls_{i,j} * \int Is_{i,j}$$

The mathematical formulation can be readily implemented using any commercial optimization software. In our case, we employed the Premium Solver available as an add-on for Microsoft Excel. The modeling output is shown in Table 4-7 for the minimum risk (R_T) solution. The optimal solution is based on individual weighting factors for each of the design decisions and each of the three risk types (schedule, cost, and performance). Design decision weighting was determined by the development team and was based on the number of requirements impacted, and their importance to overall design performance.

	Decision Matrix	Dec	Wt	Risk (Perf)	Risk (Cost)	Risk (Sched)	Risk (Total)
			Wt	0.4	0.3	0.3	1
1	Hybrid	0	0.2	0	80	100	54
2	Gas	1		50	0	0	20
3	SW switches	0	0.05	0	100	0	30
4	HW switches	1		0	25	20	13
5	Existing display	1	0.1	80	50	0	47
6	New display	0		0	80	100	54
7	Titanium	0	0.1	0	100	100	60
8	Steel	1		80	50	0	47
9	Angled geometry	0	0.25	0	80	100	54
10	Flat geometry	1		100	0	0	40
11	Weight reduction	1	0.1	0	40	0	12
12	Additional functionality	0		100	0	0	40
13	New source	0	0.2	80	0	100	62
14	Current source	1		0	80	0	24
		Risk Score		310	245	20	203
		Subject to		<= 500	<= 500	<= 500	

Table 4-7: Risk Solver Results for Minimum Total Risk (R(T)) Solution
(Optimization results for balanced concept solution; defined as all risk types below 500)

Modifying the constraints and re-optimizing to find low-risk solutions for schedule risk (R_s), performance risk (R_p) and cost risk (R_c) yields the results shown in Table 4-8 below.

	Risk (Performance)	Risk (Cost)	Risk (Schedule)	Risk (Total)	Selected Elements (Table 6 for description)
Performance Solution	50	405	320	237	2, 4, 6, 7, 9, 11, 14
Cost Solution	490	125	120	269	2, 4, 5, 8, 10, 12, 13
Schedule Solution	410	205	20	231	2, 4, 5, 8, 10, 12, 14
Balanced Solution	310	245	20	203	2, 4, 5, 8, 10, 11, 14

Table 4-8: Risk Consequence Scoring
(Design concepts for lowest performance risk, lowest cost risk, lowest schedule risk, vs. balanced solution)

Based on the output the total risk (R_T) is found to increase when performance risk (R_p) is reduced to 0. In a similar vein, minimizing the cost (R_c) or schedule risk (R_s) to their lowest values also results in an increase to total program risk. Often as one type of risk is decreased for a given concept it results in an increase to another risk type due to various design trade-offs.

This dynamic illustrates that design concepts will not always achieve minimal risk values for all risk areas simultaneously. Hence, a balanced solution must be found that will avoid delivering excessive levels of risk in any one of the areas (cost, schedule, or performance).

Finding an optimal *balanced solution* is an iterative process. It begins by establishing a *preferred* maximum acceptable value for each risk type based on input from the SMEs. Based on the initial results, the constraints for each risk type can be increased or decreased to find a feasible then optimal solution.

In practice, organizations categorize risks as either high, medium, or low based on an assessment of their likelihood and consequence. The acceptable risk level will be based on the amount of risk aversion a company possesses. Ideally, the *balanced solution* would not possess high risk for any of the risk types.

In the example above, the balanced solution yields the best overall risk assessment (R_T) despite none of the individual risk types being at their minimum level.

V. Insights

The model demonstrates that design concepts can be successfully tailored based on risk profiles. The results also confirm that alternatives analyzed along one or two dimensions of risk may not provide an optimal solution. Even for programs focused on minimizing a single risk type (such as performance or schedule alone), a comprehensive analysis is needed to understand the trade-offs that will occur across the other risk areas.

The risk tailoring method is a robust process capable of supporting analysis at both the program level and the enterprise-level. By analyzing risk profiles across the entire portfolio of products and summing the results, a cumulative risk profile can be generated for the organization. The portfolio risk would calculated as:

$$R_{T_p} = \sum_{1}^{n}[W_1 * R_{T_1} + W_2 * R_{T_2} + \cdots + W_n * R_{T_n}]$$

Where:

R_{T_p} *equals the total risk of the portfolio*

R_{T_n} *equals the total risk of project 'n'*

W_n *equals the weighted risk impact of project 'n' on the entire risk portfolio.*

In addition to the initial risk magnitude, consideration must be given to how easily the risk can be reduced or mitigated. Experience may show that a high-risk area can be reduced significantly by applying some targeted mitigations with minimal cost of program effort. The model has captured this aspect by including a remediation difficulty multiplier that considers the anticipated effort and success of mitigation actions for each option and risk area.

In the case of portfolio analysis, larger projects requiring more resources may be weighted higher in terms of their contribution to overall portfolio risk. If the portfolio analysis determines there is too much risk residing in a single area (cumulatively), it may be appropriate to make design decisions for specific projects that would shift the risk into other areas as demonstrated in the example. Having such insight into risk projections could greatly enhance product planning and allow tailoring of the entire product line to occur. The method would also help to provide quantitative justification of the forecasted product plans.

As product complexity increases, risk in design requirements also increases due to uncertainty (Browning, Deyst, 2002). Effective management of risk requires a mature risk process and an organizational culture that supports risk from the top management through the working level. It should be understood that managing risk is not synonymous with eliminating risk. Risk at a manageable level has been shown to be good for innovation, and has led to increased performance, and state-of-the-art development (Nakata and Sivakumar, 1996). Uncertainty represents both risk and opportunity (Steinberg, Martens, 2004). As such, the goal of the risk tailoring approach is to provide uncertainty in the proper amount to reduce operational surprises and losses, yet expand capability (Steinberg, Martens, 2004). It is not necessarily the goal to eliminate uncertainty.

In order to assess risk levels, several taxonomies have been developed that are based on either risk origin or risk impact. A common taxonomy employed for CoPS projects is based on cost, schedule and performance impacts (Group, 2010; Simpleman, 2006). The tailoring method employs this taxonomy.

Technical requirements drive risk across all areas. Understanding requirements and their priorities ensure the concept solution will meet customer expectations (AT&T, 1993). Several methods of prioritizing requirements have been developed and are employed in industry including the Kano model and standard tiering model (Chen and Chuang, 2008; Sauerwein, Bailom, 1996). To utilize the risk tailoring method PD teams need to develop utility curves associated with each performance level (Browning, Deyst, 2002). While Browning (2002) rightly suggests employing utility curves to quantify customer *impact* at different performance levels, it is often not feasible to generate highly detailed curves for each requirement at the early concept development stage. In these circumstances, a generic curve could be used for common groups of requirements.

The methodology presented provides a novel process for quantifying and tailoring risk across all three areas concurrently. The scoring is based directly on requirements, and can be initiated early on during concept development. The process also offers significant benefits to cost management by facilitating early design decisions, before significant investment is made.

As the quality of information improves throughout the PD lifecycle, the method can be reiterated to maintain a current assessment of program risk. Based on the work of Browning (2002) the magnitude of risk should decrease over time as uncertainty is reduced. The model can be used to track the effectiveness of risk management by tracking the rate at which risk is decreasing over time.

The risk tailoring method was designed to be consistent with established practices in systems engineering, risk management, and requirements analysis. It is expected that this will facilitate a trouble-free adoption into existing processes for CoPS projects. Although the research was applied to a military project, it is anticipated that it can be adapted to other projects and industries provided they have sufficient maturity in their risk process (Ren and Yeo, 2004).

VI. Limitations

The risk tailoring process requires a large amount of subjective assessment from the product team in areas of technical performance, cost, and schedule measures. As with many decisions made at the fuzzy-front end, the amount of available information is often limited (Kahn, 2005). As such, the experience of SMEs will be relied on heavily, particularly with CoPS projects. However, this is not unlike the existing process which relies on subjective evaluations to make product strategy decisions (Browning, Deyst, 2002). In order to properly identify risks, the organization must have a reasonable level of risk process maturity (Carr, 1993). SMEs must also be proficient in risk assessment techniques to understand how concepts must be scored in terms of both likelihood and consequence (Simpleman, 2006). This may require training in some instances.

Although the risk tailoring process was designed to be accomplished in a timely manner, it will often be executed under tight time constraints of early concept development. Care should be taken to ensure that a thorough, unbiased assessment of each concept is provided, as it will guide the final selection process. A clear understanding of customer requirements is needed to evaluate different design concepts.

The risk tailoring assessment yields a quantitative number for each risk type. These derived numbers must be clearly correlated to real-world impacts. For example, the organization must understand what the tangible difference is between a risk assessment score of ‘x’ versus ‘y’, and understand what a reasonable trade-off would be between schedule risk versus cost risk. These calculated scores need to be meaningful to the organization in order to drive appropriate decisions. The tailoring process will also require the organization to understand their own risk profile, and how much risk they are willing to accept in the given program, or portfolio of products. With experience applying the method, this understanding will come.

Because risks are probabilistic in nature they are not additive. For this reason, the method described cannot provide an *absolute* assessment of project risk in mathematical terms, and this may never be possible. However, to effectively assess the feasibility of a design concept it is often necessary to have a comprehensive view of its risk/uncertainty profile that the method does provide. This is done as a relative assessment of risk for one concept versus another.

Scoring risk for schedule and cost requires threshold targets to be established and understood by each subsystem team. Often at the fuzzy front end this information is incomplete (AT&T, 1993). Although the customer will typically provide the list of performance thresholds by requirement, the equivalent information for subsystem cost and schedule is often derived by the developer based on detailed design decisions and supplier input. Without clear targets, the PD team will be unable to establish the *unacceptable* consequence levels. Probability functions indicating the likelihood of outcomes for cost, schedule, and performance will also be necessary.

In the current risk process, after key risks are identified, detailed mitigation plans are developed to determine what actions can reduce the risk and at what cost (Group, 2010; Simpleman, 2006). Although the risk tailoring method provides a clear assessment of *initial* risk, there is still a need to implement mitigation activities into the risk assessment. Projecting mitigation difficulty and overall success can be challenging.

VII. Conclusion

Risk is a measure of uncertainties in achieving program goals in cost, schedule, or performance. Value is maximized when product strategies effectively balance risk versus expected return. Based on the growing number of unsuccessful launches in PD, such balance has been difficult to achieve. In this paper, we present a novel method for tailoring risk for early design concepts of CoPS projects. The method integrates literature related to complexity variables, PD design decisions, risk taxonomies, and risk analysis techniques. Care has been taken to ensure that the process is consistent with industry practices so it can be implemented with little disruption to the organization. The result is a robust method for early risk tailoring used to identify the optimal low-risk design strategy at concept development.

Complexity in the form of uncertainty generates risk; therefore, controlling complexity will reduce risk. Failure to control risk during PD ultimately leads to negative impacts on performance, cost, or schedule. The ability to mitigate risks depends on the available resources of the program, so the earlier that risks can be identified, the greater the likelihood that they can be planned for and mitigated. The proposed risk tailoring approach provides an effective framework for analyzing risk before significant commitments of funds are made. Since the majority of lifecycle costs are locked-in after concept development, it is here that we can achieve the most significant results.

Although previous research demonstrated methods for trading-off risk types, no single model had addressed all risk types concurrently. The results demonstrate that risk analysis should be performed along all three dimensions of risk (i.e., performance, cost, and schedule) concurrently to ensure an optimal solution is found. The risk tailoring method can be applied at both the project level and the enterprise level. Because this method provides a clear connection between *project-level decisions* and enterprise-level *product strategies,* it facilitates significant alignment in the organization. Furthermore, having this portfolio risk information up front will drive many relevant planning decisions and maximize the chance for development success.

The risk tailoring method is focused on achieving customer satisfaction because it is initiated from key customer requirements. Is has the flexibility to maintain updated status even as requirements evolve and customer preferences change. Using this method for early concept development can greatly improve complexity management and PD launch success.

Managing PD complexity has becoming increasingly difficult due to the rapid advances in technology and global competition. In the desire to remain competitive, organizations need to be careful not to over-commit their resources on overly risky projects that could accomplish their goals through a more calculated approach. Often the drive to develop the highest performing product or subsystems can overshadow the realities of what is achievable by an organization.

CHAPTER 5: Summary

I. Research Conclusion and Implications

This research presents methods for utilizing complexity and risk constructs to improve product development for CoPS projects. The results from applying the proposed methods on a defense industry project demonstrates that it is feasible to assess complexity and risk in the early stages of PD to guide resource planning and design decisions. Several variables for aligning complexity to the organization were presented including the: (1) product requirements, (2) modularity changes, (3) technology maturity, etc. Opportunities for tailoring organizational capabilities included: (1) increasing staffing, (2) co-locating teams, (3) assigning more experienced members, and (4) increasing development time. The key elements for tailoring should be selected based on the industry and product attributes.

The models presented were designed to be employed at the earliest point in development—receipt of customer requirements. Early analysis provides the design team with a significant level of understanding from the onset of the project which provides benefits in developing a winning bid and achieving launch success. Early tailoring also facilitates the effective management of life-cycle costs by influencing design before significant investment has been made.

Although the models were developed and validated on a CoPS project, it is anticipated they can be applied to other requirements-based projects as well with minor modifications. Further studies are needed to verify the effectiveness of the models in each context.

The tailoring methods can be easily integrated into PD organizations, as they are consistent with traditional systems engineering (SE) processes. The primary SE processes leveraged in this work included: (1) requirements allocation, (2) risk management, and (3) trade studies. The research also leverages the use of existing data rather than employing new data-collection activities. Implementing these models using existing processes and information allows the analysis to be updated and maintained with minimal effort.

Although the methods were demonstrated on a single project, they were designed to be applied across multiple projects concurrently to aid in planning at the portfolio level. Providing a consolidated view of the risk and complexity of the entire portfolio will reveal significant opportunities for aligning resources within the organization. It is also essential when evaluating new programs to understand for their impact to the current product line to avoid over-commitment of resources.

In an effort to improve risk management effectiveness recent literature has focused on the mechanics of the process rather than the identification and mitigation of risks. Unfortunately, successful risk identification requires more than a well-defined process. It requires effective facilitation techniques be implemented in a culture that is committed to documenting and resolving risks. To improve risk identification several risk taxonomies were reviewed that highlight common areas of vulnerability in PD. This research extends that work by evaluating risks that are specific to CoPS projects to help guide in early risk identification.

Achieving the full benefits of risk management requires support from top management to encourage risk identification strategies throughout the organization, and avoid such negative behaviors as deliberate ignorance and risk avoidance (Kutsch and Hall, 2010)(Kutsch and Hall, 2010).

Managing risk is not synonymous with eliminating risk. Risk at a manageable level can facilitate innovation and lead to increased performance, and technology development. The goal of risk profiling is to provide uncertainty in the proper amount to reduce operational losses, yet expand capability--it is not to eliminate risk entirely.

Managing PD complexity has been increasingly difficult due to technology advancement and global competition. To remain competitive organizations should avoid over-commitment of resources on overly risky or ambitious projects. Achieving all performance thresholds is the goal, by understanding the expectations of the customer, and that which is achievable by your organization.

II. Recommendations for Future

The complexity estimation model requires significant data input from the organization and subject matter experts. To be successful the assessment should utilize variables that leverage available data and minimize the burden of assessment. Integrating this research with future studies related to business analytics models or organizational planning tools will provide opportunities to further refine the process and allow the most detail to be collected with the least commitment of resources / time. Moving toward a method of seamless, automated assessments would expand the use of the model, and allow more extensive validation of the techniques and refinement of the data.

This research takes a major step in quantifying the amount of risk management activity needed to ensure success. However, the results are preliminary and are based on data from one major CoPS project. The research lays the groundwork for more extensive studies in the future which should involve multiple programs to improve the estimation methods and make the process more robust. Conducting extended studies will expand the use of these methods across multiple projects, in multiple industries, and improve their prediction accuracy.

In developing a risk profile, a proper understanding of requirements and their priorities is necessary to meet customer expectations. In the method presented, utility curves were utilized to establish specific values of various performance levels. Due to the limited time available in early development it is not always feasible to generate customized, highly detailed utility curves for each requirement. In such instance, future research to define common utility curves would be beneficial in reducing the time to generate risk profiles, and improve the accuracy of customer valuation.

Estimating needed risk activity requires an accurate complexity assessment. Until more quantitative and accurate assessments of complexity are available and accepted, the use of this approach may be limited. Future research in alternative complexity estimation techniques should be pursued.

In organizations plagued by such tendencies as risk avoidance and deliberate ignorance, the true benefits of risk management can never be realized. Risk management requires a culture of embracing risk identification to ensure sufficient reporting is being accomplished. To extend the research in this paper, methods for changing the culture related to perception of risk management, and practicing risk management should be pursued. Organizations that approach risk management as a form of pro-active problem resolution rather than an admission of failure will realize far more success in their programs.

Effective risk identification is one of the most important aspects of risk management and a key contributor to its success. To assist in risk identification several generalized taxonomies have been proposed to guide the identification process. Unfortunately, this is only a starting point, as many risks are specific to project type and industry. To better understand risks among different products and industries future studies should be performed using historical risk data from relevant programs. Establishing risk taxonomies based on actual data by project / industry will help to highlight key areas of vulnerability in early development. This information may also suggest mitigation strategies that could be planned in the early phases to reduce risk initially.

Despite the importance of risk management there remains a significant disparity of resources applied to risk between organizations and industries. Research suggests it has been difficult to determine the proper amount of risk activity needed to support PD success. This research provides a starting point for estimating risk activity needed to achieve success in CoPS projects. However, future, more extensive studies can provide additional fidelity in terms of the absolute quantity and types of risks that are appropriate to support effective risk management in complex vehicle development projects.

In this dissertation, we present novel methods for assessing risk and complexity of early design concepts for CoPS projects. When applying these methods to other industries, care must be taken to ensure the process is aligned with the organizational practices so it can be implemented with little disruption to the organization. Future studies are needed to find the best approach to adapting these techniques across multiple industries, using relevant variables, and available information.

The risk profiling process quantifies risk in a single measure (by type). In order to effectively support design decisions the derived numbers must clearly correlate to real-world impacts. Organizations must understand the tangible difference between risk assessment scores and the trade-off that occurs between a given risk level vs. performance. Future research indicating what an acceptable risk vs. performance trade-off may be, and how it is influenced by the organization's overall risk tolerance is beneficial. The scores should be meaningful enough to drive design decisions. Additional research is also needed to understand how the process may be applied to the entire portfolio of products.

APPENDIX 1 – Complexity Assessment Scoring Criteria

Variable Description	Measure	Scoring Criteria (5 = most difficult, 1 = least difficult)	Variable	Calculation	Difficulty Multiplier
Coordination / Logistics Challenges - 'L'	% of requirements needing secondary coordination with other groups	5 - > 60% or more of requirements effort requires coordination 4 - 50 to 59% of requirements effort requires coordination 3 - 40 to 49% of requirements effort requires coordination 2 - 25 to 40% of requirements effort requires coordination 1 - less than 25% of requirements effort requires coordination	Sr	L = (Sr + Lt) / 2	Mi = Average (L+F+E+N) / unity Note: Unity = 3.0 in order to scale the scoring from .33 to 1.67
	Available development lead time	5 - Design, integration, or deliverables needed under compressed development timing 4 - Design, integration, or deliverables on or near critical path with little to no lead time slack (0 - 5% slack) 3 - Design, integration or deliverables within development lead time with moderate slack available (5% - 15% slack)	Lt		
Performance Flexibility - 'F'	% of Tier 1 requirements assigned	5 - > 5% of Tier 1 requirements 3 - 1 to 5% of Tier 1 requirements 1 - 0% of Tier 1 requirements	T1	F = (2T1 + T2) / 3	
	% of Tier 2 requirements assigned	5 - 15% or more of Tier 2 requirements 3 - 1 to 14% of Tier 2 requirements 1 - 0% of Tier 2 requirements	T2		
Experience / Capability - 'E'	Commodity experience	5 - No exposure or working knowledge (0 yrs) 4 - Limited exposure with limited working knowledge (1 - 2 yrs) 3 - Moderate exposure and working knowledge (3 - 4 yrs) 2 - Good exposure with solid working knowledge (5 - 10 yrs) 1 - Extensive exposure and extensive working knowledge (10- 20 yrs)	Ec	E = (Ec+ Ei) / 2	
	Industry experience	5 - No exposure or working knowledge (0 yrs) 4 - Limited exposure with limited working knowledge (1- 2 yrs) 3 - Moderate exposure and working knowledge (3 - 4 yrs) 2 - Good exposure with solid working knowledge (5 - 10 yrs) 1 - Extensive exposure and extensive working knowledge (10- 20 yrs)	Ei		
Novelty / Newness of Technology - 'N'	Novelty of technology / requirements	5 - New technology, and/or extensive performance requirements and integration. Never before achieved in the Industry or application) 4 - New design with challenging to extensive improvements or integration required 3 - New and carry-over design with moderate to challenging improvements 2 - Carry-over design with moderate upgrades and integration required 1 - Carry-over products with little to no performance improvements or integration	N	N	

APPENDIX 2 – List of Common Questions in PD

Risk Variable(s) Impacted

Area	PD Decisions	Technology	Leadtime	Coupling / Integration	Requirements	Material / Cost	Team competency	Early Modeling Capability	Commonality	Sourcing	Process steps
Concept Development	What are the target values of the product attributes, including price?				x						
	What is the core product concept?	x		x	x				x		
	What is the product architecture?			x							
	What variants of the product will be offered?			x					x		
	Which components will be shared across which variants of the product?			x					x		
	What will be the overall physical form and industrial design of the product?	x			x						
Supply Chain	Which components will be designed and which will be selected? Who will design the components?									x	
	Who will produce the components and assemble the product?									x	x
	What is the configuration of the physical supply chain, including the location of the decouple point?									x	
	What type of process will be used to assemble the product?										x
	Who will develop and supply technology and equipment?									x	
Product Design	What are the values of the key design parameters?				x						
	What is the configuration of the components and assembly			x							
	What is the detailed design of the components, including material and process selection?				x	x	x				x
Performance Testing	What is the prototyping plan?	x			x	x		x			x
	What technologies should be used for prototyping?	x						x			
Production	What is the plan for market launch and testing?		x					x			
	What is the plan for ramp-up?		x							x	
	What is the market and product strategy to maximize probability of economic success?	x	x		x				x		x
Product Strategy and Planning	What portfolio of product opportunities will be pursued?	x			x				x		
	What is the timing of product developmentprojects?		x								
	What, of any assets will be shared across which products?			x					x		
	Which technologies will be employed in the product(s)?	x									
Product Development	Will a functional, project, or matrix organization be used?						x				
	How will teams be staffed?						x				
	How will project performance be measured?						x				
	What will be the physical arrangement and location of the team?						x				
	What investments in infrastructure, tools and training be made?						x				
	What type of development process will be employed (e.g. stage-gate)?						x				x
Project Management	What is the relative priority of development objectives?				x						
	What is the planned timing and sequence of development activities / major milestones?		x								x
	What will be the communication mechanisms among team members?						x				
	How will the project be monitored and controlled?						x				x

Krishnan and Ulrich (2001), List of Common Questions in Product Development

REFERENCES

Abdelgawad, M. and A.R. Fayek. 2012. Comprehensive Hybrid Framework for Risk Analysis in the Construction Industry Using Combined Failure Mode and Effect Analysis, Fault Trees, Event Trees, and Fuzzy Logic. Journal of Construction Engineering and Management-Asce, 138(5): 642-651.

Accountability Office, U.G. 2008. DEFENSE ACQUISITIONS: A Knowledge-Based Funding Approach Could Improve Major Weapon System Program Outcomes, U.S.G.A. Office, Editor Committee on Armed Services, U.S. Senate:

Adler, T.R., J.G. Leonard, and R.K. Nordgren. 1999. Improving risk management: moving from risk elimination to risk avoidance. Information and Software Technology, 41(1): 29-34.

Ahmed, A. 2007. A review of techniques for risk management in projects. Benchmarking : an international journal, 14(1): 22-36.

Akintoye, A.S. and M.J. MacLeod. 1997. Risk analysis and management in construction. International Journal of Project Management, 15(1): 31-38.

Alberts, C.J. and A.J. Dorofee. 2010. Risk Management Framework: United States. 72p.

Anonymous. 2010. Defense Cost and Resource Center - Enhancing DoD Cost Analysis. [cited 2010; Available from: http://dcarc.pae.osd.mil/.

Antonio, K.W.L., C.M.Y. Richard, and E. Tang. 2009. The complementarity of internal integration and product modularity: An empirical study of their interaction effect on competitive capabilities. Journal of Engineering and Technology Management, 26(4): 305-326.

Armstrong, J.S., K.C. Green, and A. Graefe. 2007. Methods to Elicit Forecasts from Groups: Delphi and Prediction Markets Compared. Foresight: The International Journal of Applied Forecasting, (Fall 2007).

AT&T. 1993. Design to reduce Technical Risk: McGraw-Hill. 752.

Atkinson, R., L. Crawford, and S. Ward. 2006. Fundamental uncertainties in projects and the scope of project management. International Journal of Project Management, 24(8): 687-698.

Baccarini, D. 1996. The concept of project complexity--a review. International Journal of Project Management, 14(4): 201-204.

Baccarini, D., G. Salm, and P.E.D. Love. 2004. Management of risks in information technology projects. Industrial Management & Data Systems, 104(3): 9.

Bajaj, D., J. Oluwoye, and D. Lenard. 1997. An analysis of contractors' approaches to risk identification in New South Wales, Australia. Construction Management & Economics, 15(4): 7.

Benoit, A.A., M. Patry, and S. Rivard. (2001) Managing IT Outsourcing Risk : Lessons Learned. 19.

Berger, C. 1993. Kano's methods for understanding customer-defined quality. Center for Quality Management Journal, (Fall): 3-35.

Bhaskaran, S.R. and K. Ramachandran. 2011. Managing Technology Selection and Development Risk in Competitive Environments. Production & Operations Management, 20(4): 541-555.

Boehm, B.W. 1991. Software risk management: principles and practices. Software, IEEE, 8(1): 32-41.

Boehm, B.W. 1988. A spiral model of software development and enhancement. Computer, 21(5): 61-72.

Browning, T.R., L.M.A. Co, and F. Worth. 2001. Applying the design structure matrix to system decomposition and integration problems: a review and new directions. IEEE Transactions on Engineering management, 48(3): 292-306.

Browning, T.R., et al. 2002. Adding value in product development by creating information and reducing risk. IEEE Transactions on Engineering Management, 49(4): 443-58.

Browning, T.R. and S.D. Eppinger. 2002. Modeling impacts of process architecture on cost and schedule risk in product development. Engineering Management, IEEE Transactions on, 49(4): 428-442.

Browning, T.R. and R.V. Ramasesh. 2007. A survey of activity network-based process models for managing product development projects. Production and Operations Management, 16(2): 217-240.

Budget, U.O.o.M.a. 2009. Budget of the US Government, Fiscal Year 2010. P. 6.

Campagnolo, D. and A. Camuffo. 2010. The Concept of Modularity in Management Studies: A Literature Review. International Journal of Management Reviews, 12(3): 259-283.

Canada, T.B.o. 2010. A Guide to Project Gating for IT-Enabled Projects, T.B.o.C. Secretariat, Editor. 31.

Carr, M.J., Konda, Suresh L., Monarch, Ira F., Ulrich, Carol. 1993. Taxonomy-Based Risk Identification: Pittsburgh, PA. 90.

Carrillo, J.E. and R.M. Franza. 2006. Investing in product development and production capabilities: The crucial linkage between time-to-market and ramp-up time. European Journal of Operational Research, 171(2): 536-556.

Catrinu, M.D. and D.E. Nordgård. 2011. Integrating risk analysis and multi-criteria decision support under uncertainty in electricity distribution system asset management. Reliability Engineering & System Safety, 96(6): 663-670.

Chan, F.T.S. and N. Kumar. 2007. Global supplier development considering risk factors using fuzzy extended AHP-based approach. Omega, 35(4): 417-431.

Chapman, C. 1997. Project risk analysis and management—PRAM the generic process. International Journal of Project Management, 15(5): 273-281.

Chapman, C. and S. Ward. 2003. Project Risk Management Processes, Techniques, and Insights. Second ed, ed. J.W. Sons. Chichester, England: John Wiley & Sons Ltd. 360.

Chapman, R.J. 2001. The Controlling Influences of effective Risk Identification and Assessment for Construction Design Management. International Journal of Project Management: 13.

Chapman, R.J. 1998. The effectiveness of working group risk identification and assessment techniques. International Journal of Project Management, 16(6): 333-343.

Chen, C.-C. and M.-C. Chuang. 2008. Integrating the Kano model into a robust design approach to enhance customer satisfaction with product design. International Journal of Production Economics, 114(2): 667-681.

Chen, S.-J. and L. Lin. 2003. Decomposition of interdependent task group for concurrent engineering[small star, filled]. Computers & Industrial Engineering, 44(3): 435-459.

Chittister, C. and Y.Y. Haimes. 1993. Risk associated with software development: a holistic framework for assessment and management. IEEE Transactions on Systems, Man and Cybernetics, 23(3): 710-723.

Choi, D.W., J.S. Kim, and H.G. Choi. 2009. Determination of Integrated Risk Degrees in Product Development Project. in World Congress on Engineering and Computer Science 2009, 20-22 Oct. 2009. 2009. Hong Kong, China: International Association of Engineers.

Choi, H.-G. and J. Ahn. 2010. Risk analysis models and risk degree determination in new product development: A case study. Journal of Engineering and Technology Management - JET-M, 27(1-2): 110-124.

Christensen, C.M. 1992. EXPLORING THE LIMITS OF THE TECHNOLOGY S-CURVE. PART I: COMPONENT TECHNOLOGIES. Production and Operations Management, 1(4): 334-357.

Clift, T.B. and M.B. Vandenbosch. 1999. Project Complexity and Efforts to Reduce Product Development Cycle Time. Journal of Business Research, 45(2): 187-198.

Closs, D.J., et al. 2008. Toward a theory of competencies for the management of product complexity: Six case studies. Journal of Operations Management, 26(5): 590-610.

Conrow, E.H. and P.S. Shishido. 1997. Implementing risk management on software intensive projects. Software, IEEE, 14(3): 83-89.

Cooper, L.G. 2000. Strategic Marketing Planning for Radically New Products. The Journal of Marketing, 64(1): 1-16.

Cooper, L.P. 2003. A research agenda to reduce risk in new product development through knowledge management: a practitioner perspective. Journal of Engineering and Technology Management, 20(1–2): 117-140.

Cooper, R., S. Edgett, and E. Kleinschmidt. 2001. Portfolio management for new product development: results of an industry practices study. R&D Management, 31(4): 20.

Cooper, R.G., S.J. Edgett, and E.J. Kleinschmidt. 1999. New Product Portfolio Management: Practices and Performance. Journal of Product Innovation Management, 16(4): 333-351.

Covello, V.T. and J. Mumpower. 1985. Risk Analysis and Risk Management: An Historical Perspective. Risk Analysis, 5(2): 103-120.

Curtis, B., H. Krasner, and N. Iscoe. 1988. A field study of the software design process for large systems. Commun. ACM, 31(11): 1268-1287.

Danilovic, M. and T.R. Browning. 2007. Managing complex product development projects with design structure matrices and domain mapping matrices. International Journal of Project Management, 25(3): 300-314.

de Klerk, A.M. 1999. Risk management heuristics in new product development. in Proceedings of Portland International Conference on Management of Engineering and Technology, 25-29 July 1999. 1999. Portland, OR, USA: Portland Int. Conf. Manage. Eng. & Technol. PICMET.

de Visser, M., et al. 2009. Structural ambidexterity in NPD processes: A firm-level assessment of the impact of differentiated structures on innovation performance. Technovation, 30(5-6): 291-299.

Defense, U.D.o. 2010. Spending Reforms Top Defense Priorities. [cited 2010 October 20, 2010]; Available from: http://www.defencetalk.com/spending-reforms-top-defense-priorities-29551/.

Dickson, G.C.A. and W.J. Haystings. 1989. Corporate Risk Management. London: Witherby and Company.

Doreian, P. and F.N. Stokman. 1997. Evolution of Social Networks. Vol. 1. Amsterdam: Gordon and Breach.

Emden, Z., R.J. Calantone, and C. Droge. 2006. Collaborating for New Product Development: Selecting the Partner with Maximum Potential to Create Value. Journal of Product Innovation Management, 23(4): 330-341.

Engineers, S.o.M. (2010) Need to Stay Up with Manufacturing Change in the Aerospace and Defense Industry? Want a Career in Aerospace Manufacturing? SME is the Industry's Best Resource

Eppinger, S., et al. 1994. A model-based method for organizing tasks in product development. Research in Engineering Design, 6(1): 1-13.

Eppinger, S.D. 2001. Innovation at the Speed of Information. Harvard Business Review, 79(1): 149-158.

Firesmith, D. 2004. Prioritizing Requirements. Journal of Object Technology, 3(8): 12.

Fisher, M.L. 1997. What is the right supply chain for your product? Harvard business review, 75: 105.

Fixson, S.K. 2007. What exactly is Product Modularity? The answer depends on who you ask, MIT Sloan School of Management: Cambridge, MA 02142.

Garcia, R. and R. Calantone. 2002. A critical look at technological innovation typology and innovativeness terminology: a literature review. Journal of Product Innovation Management, 19(2): 110-132.

Garrick, B.J. 1988. The approach to risk analysis in three industries: nuclear power, space systems, and chemical process. Reliability Engineering & System Safety, 23(3): 195-205.

Gidado, K.I. 1996. Project complexity: The focal point of construction production planning. Construction Management & Economics, 14(3): 213-225.

Gilbert, N., A. Pyka, and P. Ahrweiler. 2001. Innovation Networks - A Simulation Approach. Journal of Artificial Societies and Social Simulation Vol 4(No 3).

Gokpinar, B., W.J. Hopp, and S.M.R. Iravani. 2010. The Impact of Misalignment of Organizational Structure and Product Architecture on Quality in Complex Product Development. Management Science, 56(3): 468-484.

Golfarelli, M., S. Rizzi, and L. Cella. 2004. Beyond Data Warehousing: What's Next in Business Intelligence? in Proceedings of the 7th ACM international workshop on Data warehousing and OLAP 2004: Association for Computing Machinery.

Gomes, P.J. and N.R. Joglekar. 2008. Linking modularity with problem solving and coordination efforts. Managerial and Decision Economics, 29(5): 443-457.

Griffin, A. 1997. The Effect of Project and Process Characteristics on Product Development Cycle Time. Journal of Marketing Research, 34(1): 24-35.

Griffin, A. 1993. Metrics for measuring product development cycle time. Journal of Product Innovation Management, 10(2): 112-125.

Griffin, A. 1997. Modeling and measuring product development cycle time across industries. Journal of Engineering and Technology Management, 14(Copyright 1998, IEE): 1-24.

Group, I.S.H.W. 2010. Systems Engineering Handbook. January 2010 ed. Vol. v. 3.2. San Diego: International Counsel on Systems Engineering. 374.

Gupta, A.K., S.P. Raj, and D. Wilemon. 1986. A Model for Studying R&D. Marketing Interface in the Product Innovation Process. The Journal of Marketing, 50(2): 7-17.

Harland, C., R. Brenchley, and H. Walker. 2003. Risk in supply networks. Journal of Purchasing and Supply Management, 9(2): 51-62.

Harned, D.S., Lundquist, Jerrold T. . 2003. What Transformation Means to the Defense Industry. The McKinsey Quarterly, (Number 3).

Harrison, E.F. 1975. The Managerial Decision Making Process. Boston: Houghton Mifflin Company.

Hart, S., et al. 2003. Industrial Companies' Evaluation Criteria in New Product Development Gates. Journal of Product Innovation Management, 20(1): 22-36.

Hasson, F., S. Keeney, and H. McKenna. 2000. Research guidelines for the Delphi survey technique. Journal of Advanced Nursing, 32(4): 1008-1015.

Hedgebeth, D. 2007. Data-driven decision making for the enterprise: an overview of business intelligence applications. VINE, 37(4): 6.

Henderson, R.M. and K.B. Clark. 1990. Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms. Administrative Science Quarterly, 35(1): 9-30.

Hillson, D. 2006. Integrated Risk Management as a Framework for Organizational Success, in PMI Global Congress: Seattle, WA.

Hillson, D., Grimaldo, S., Rafele, C. 2006. Managing Project Risks Using Cross Risk Breakdown Matrix. Risk Management 8(1): 15.

Hillson, D. 2002. Project Risk Management: Future Development. 11.

Hobday, M. 1998. Product complexity, innovation and industrial organisation. Research Policy, 26(6): 689-710.

Hobday, M. 2000. The project-based organisation: an ideal form for managing complex products and systems? Research Policy, 29(7–8): 871-893.

Hofmann, H.F. and F. Lehner. 2001. Requirements engineering as a success factor in software projects. Software, IEEE, 18(4): 58-66.

Holt, K. 1996. Brainstorming--from classics to electronics. . Journal of Engineering Design, 7(1): 7.

Hölttä, K.M.M. and K.N. Otto. 2005. Incorporating design effort complexity measures in product architectural design and assessment. Design Studies, 26(5): 463-485.

Holzmann, V. and I. Spiegler. 2011. Developing risk breakdown structure for information technology organizations. International Journal of Project Management, 29(5): 537-546.

Howell, D., C. Windahl, and R. Seidel. 2010. A project contingency framework based on uncertainty and its consequences. International Journal of Project Management, 28(3): 256-264.

Hsiao, A. and M. Ploughman. 2009. Strategic risk management in new product development. in 4th European Conference on Entrepreneurship and Innovation, 10-11 Sept. 2009. 2009. Reading, UK: Academic Publishing Limited.

Hulett, D. 2001. Key Characteristics of a Mature Risk Management Process, in Fourth European Project Management Conference: London, UK. 7.

INST, C.-M.U.P.P.S.E. 2002. Capability Maturity Model (Trademark) Integration (CMMI (Service Mark)), Version 1.1. CMMI (Service Mark) for Software Engineering (CMMI-SW, V1.1), Carnegie Mellon: Pittsburgh, PA 15213-3890. 645.

Institute, P.M. 2008. A Guide to the Project Management Book of Knowledge (PMBOK) Fourth Edition, Project Management Institute: Pennsylvania. 506.

Jacobs, M.A. 2007. Product Complexity: A Definition

and Impacts on Operations. Decision Line, October 2007: 7.

Jayaram, J. and R. Narasimhan. 2007. The influence of new product development competitive capabilities on project performance. IEEE Transactions on Engineering Management, 54(2): 241-256.

Johnson, D. 1997. The triangular distribution as a proxy for the beta distribution in risk analysis. Journal of the Royal Statistical Society: Series D (The Statistician), 46(3): 387-398.

Johnson, M.D. 2009. A Framework for Incorporating Time, Cost, and Fidelity Trade-offs among Design Assessment Methods in Product Development. in IEEE International Conference on Industrial Engineering and Engineering Management. 2009.

Jolly, D. 2003. The issue of weightings in technology portfolio management. Technovation, 23(5): 383-391.

Jones, M.W. 2010. Design for Affordability, Recapitalize to Lower Ownership Costs Available from: http://www.defensenews.com/story.php?i=4628379.

Kahn, K. ed. 2005. PDMA Handbook of New Product Development. Second ed., John Wiley & Sons: Hoboken, NJ. 625.

Karlsson, J. 1996. Software requirements prioritizing. in Requirements Engineering, 1996., Proceedings of the Second International Conference on. 1996.

Kayis, B., et al. 2006. Risk Quantification for New Product Design and Development in a Concurrent Engineering Environment. CIRP Annals - Manufacturing Technology, 55(1): 147-150.

Keil, M., et al. 1998. A framework for identifying software project risks. COMMUNICATIONS OF THE ACM, 41(11): 7.

Keizer, J.A., J.P. Vos, and J.I.M. Halman. 2005. Risks in new product development: devising a reference tool. R & D Management, 35(3): 297-309.

Kim, J. and D. Wilemon. 2003. Sources and assessment of complexity in NPD projects. R&D Management, 33(1): 15-30.

Kotz, S. and J. René van Dorp. 2004. The Triangular Distribution, in Beyond Beta: Other Continuous Families of Distributions with Bounded Support and Applications. 2004, World Scientific Publishing Co. 308.

Krishnan, V. and K.T. Ulrich. 2001. Product Development Decisions: A Review of the Literature. Management Science, 47(1): 1-21.

Kutsch, E. and M. Hall. 2010. Deliberate Ignorance in Project Risk Management. International Journal of Project Management, (28): 10.

Langlois, R.N. 2002. Modularity in technology and organization. Journal of Economic Behavior & Organization, 49(1): 19-37.

Larson, E.W. and D.H. Gobeli. 1989. Significance of project management structure on development success. Engineering Management, IEEE Transactions on, 36(2): 119-125.

Lee, H.L. 2002. Aligning supply chain strategies with product uncertainties. California management review, 44(3): 105.

Lee, S.M. and S. Hong. 2002. An enterprise-wide knowledge management system infrastructure. Industrial Management & Data Systems Engineering, 102(1): 8.

Liu, J., et al. 2010. Computing with Words in Risk Assessment. International Journal of Computational Intelligence Systems, 3(4): 396-419.

MacCormack, A., R. Verganti, and M. Iansiti. 2001. Developing Products on "Internet Time": The Anatomy of a Flexible Development Process. Management Science, 47(1): 133-150.

MacDonell, S.G. 2002. Comparative review of functional complexity assessment methods for effort estimation. Software & microsystems, 9(3): 107.

Mankins, J.C. 2002. Approaches to strategic research and technology (R&T) analysis and road mapping. Acta Astronautica, 51(1–9): 3-21.

Marhavilas, P.K., D. Koulouriotis, and V. Gemeni. 2011. Risk analysis and assessment methodologies in the work sites: On a review, classification and comparative study of the scientific literature of the period 2000-2009. Journal of Loss Prevention in the Process Industries, 24(5): 477-523.

Matzler, K. and H.H. Hinterhuber. 1998. How to make product development projects more successful by integrating Kano's model of customer satisfaction into quality function deployment. Technovation, 18(1): 25-38.

McDonough, E.F., K.B. Kahn, and A. Griffin. 1999. Managing Communication in Global Product Development teams. IEEE Transactions on Engineering Management, 4(46): 9.

McManus, H. and D. Hastings. 2006. A framework for understanding uncertainty and its mitigation and exploitation in complex systems. Engineering Management Review, IEEE, 34(3): 81-81.

Meyer, M.H. and J.M. Utterback. 1995. Product development cycle time and commercial success. Engineering Management, IEEE Transactions on, 42(4): 297-304.

Michelena, N.F. and P.Y. Papalambros. 1997. A hypergraph framework for optimal model-based decomposition of design problems. Computational Optimization and Applications, 8(2): 173-196.

Mihm, J., C. Loch, and A. Huchzermeier. 2003. Problem-Solving Oscillations in Complex Engineering Projects. Management Science, 49(6): 733-750.

Mikkola, J.H. 2006. Capturing the Degree of Modularity Embedded in Product Architectures. Journal of Product Innovation Management, 23(2): 128-146.

Mikkola, J.H. 2001. Portfolio management of R&D projects: implications for innovation management. Technovation, 21(7): 423-435.

Mojtahedi, S.M.H., S.M. Mousavi, and A. Makui. 2010. Project risk identification and assessment simultaneously using multi-attribute group decision making technique. Safety Science, 48(4): 499-507.

Morelli, M.D., S.D. Eppinger, and R.K. Gulati. 1995. Predicting Technical Communication in Product Development Organizations. IEEE Transactions on Engineering management, 42(3): 215-222.

Murmann, P.A. 1994. Expected development time reductions in the German mechanical engineering industry. Journal of Product Innovation Management, 11(3): 236-252.

Nakata, C. and K. Sivakumar. 1996. National Culture and New Product Development: An Integrative Review. The Journal of Marketing, 60(1): 61-72.

Newman, M.E.J. 2006. Modularity and community structure in networks. Proceedings of the National Academy of Sciences, 103(23): 8577-8582.

Nordgård, D.E. 2012. A framework for risk-informed decision support in electricity distribution companies utilizing input from quantitative risk assessment. International Journal of Electrical Power & Energy Systems, 43(1): 255-261.

Novak, S. and S.D. Eppinger. 2001. Sourcing by design: product complexity and the supply chain. Management Science, 47(Copyright 2001, IEE): 189-204.

Olson, B.A., et al. 2012. Problem management process, filling the gap in the systems engineering processes between the risk and opportunity processes. Systems Engineering, 15(3): 275-286.

Olson J., S.S., Diaz G., and Thompson G. 2009. Department of Homeland Security Science and Technology Readiness Level Calculator (VER 1.1), Homeland Security Studies and Analysis Institute: Arlington. 79.

Paté-Cornell, E. and R. Dillon. 2001. Probabilistic risk analysis for the NASA space shuttle: a brief history and current work. Reliability Engineering & System Safety, 74(3): 345-352.

Power, M. 2007. Organized uncertainty: designing a world of risk management: Oxford University Press. 233.

Raz, T. and D. Hillson. 2005. A Comparative Review of Risk Management Standards. Risk Management, 7(4): 53-66.

Ren é van Dorp, J. and S. Kotz. 2002. A novel extension of the triangular distribution and its parameter estimation. Journal of the Royal Statistical Society: Series D (The Statistician), 51(1): 63-79.

Ren, Y.T. and K.T. Yeo. 2004. Risk management capability maturity model for complex product systems (CoPS) projects. in Engineering Management Conference, 2004. Proceedings. 2004 IEEE International. 2004.

Rickards, T. 1974. Problem-solving through creative analysis. Epping: Gower Press

Rubenstein-Montano, B., et al. 2001. A systems thinking framework for knowledge management. Decision Support Systems, 31(1): 5-16.

Sanderson, J. 2012. Risk, uncertainty and governance in megaprojects: A critical discussion of alternative explanations. International Journal of Project Management, 30(4): 432-443.

Sarbacker, S.D. and K. Ishii. 1997. A Framework for Evaluating Risk in Innovative Product Development, in ASME Design Engineering Technical ConferenceASME DETC: Sacramento, CA. 10.

Sargis Roussel, C. and F. Deltour. 2012. Beyond cross-functional teams: knowledge integration during organizational projects and the role of social capital. Knowl Manage Res Prac, 10(2): 128-140.

Sauerwein, E., et al. 1996. The KANO Model: How to Delight Your Customers, in International Working Seminar on Production Economics: Innsbruck/Igls/Austria. 313 -327.

Sauser, B., et al. 2006. From TRL to SRL: The Concept of Systems Readiness Levels. in Conference on Systems Engineering Research. 2006. Los Angeles, CA: Stevens Institute of Technology.

Schmickl, C. and A. Kieser. 2008. How much do specialists have to learn from each other when they jointly develop radical product innovations? Research Policy, 37(3): 473-491.

Schwartz, M. 2010. Defense Acquisitions: How DOD Acquires Weapon Systems and Recent Efforts to Reform the Process, Congressional Research Service.

Scott, S., D. . 1983. Using the nominal group technique effectively. National Productivity Review, 2(2): 173-184.

Shane, S.A. and K.T. Ulrich. 2004. Technological innovation, product development, and entrepreneurship in Management Science. Management Science, 50(2): 133-144.

Sharman, D.M. and A.A. Yassine. 2004. Characterizing complex product architectures. Systems Engineering, 7(1): 35-60.

Sherman, J.D. 2004. Optimal modes and levels of integration, and the identification of cross-functional coordination deficiencies in concurrent engineering. Engineering Management, IEEE Transactions on, 51(3): 268-278.

Simpleman, L.M., Paul ; Bahnmaier, Bill ; Evans, Ken ; Lloyd, Jim. 2006. Risk Management Guide for DoD Acquisition. 148.

Slater, S.F. and J.C. Narver. 1995. Market Orientation and the Learning Organization. The Journal of Marketing, 59(3): 63-74.

Smartt, C. and S. Ferreira. 2012. Constructing a general framework for systems engineering strategy. Systems Engineering, 15(2): 140-152.

Smith, J.D., II. 2005. An Alternative to Technology Readiness Levels for Non-Developmental Item (NDI) Software. in System Sciences, 2005. HICSS '05. Proceedings of the 38th Annual Hawaii International Conference on. 2005.

Smith, P.G. 1999. Managing risk as product development schedules shrink. Research Technology Management, 42(5): 25-32.

Smith, P.G., Reinertsen, Donnald G. . 1992. Shortening the Product Development Lifecycle. Research-Technology Management, (May-June): 6.

Smith, R.D. 1998. Simulation Article. Available from: http://www.modelbenders.com/encyclopedia/encyclopedia.html.

Smith, R.P. and J.A. Morrow. 1999. Product development process modeling. Design Studies, 20(3): 237-261.

Sosa, M. 2008. A structured approach to predicting and managing technical interactions in software development. Research in Engineering Design, 19(1): 47-70.

Sosa, M.E., Eppinger, Steven D., Rowles, Craig M. . 2000. Designing Modular and Integrative Systems, in ASME 2000 International Design Engineering Technical Conferences and Computers and Information in Engineering ConferenceASME: Baltimore, Maryland.

Sosa, M.E., S.D. Eppinger, and C.M. Rowles. 2003. Identifying modular and integrative systems and their impact on design team interactions. Journal of Mechanical Design, 125(2): 240-252.

Sosa, M.E., S.D. Eppinger, and C.M. Rowles. 2004. The Misalignment of Product Architecture and Organizational Structure in Complex Product Development. Management Science, 50(12): 1674-1689.

Steinberg, R.M., et al. 2004. Enterprise Risk Management-Integrated Framework, in Internal Control - Integrated Framework Committee of Sponsoring Organizations of the Treadway Commission. 7.

Suh, N.P. 1999. A Theory of Complexity, Periodicity and the Design Axioms. Research in Engineering Design, 11(2): 116-132.

Sun, J., Y. Zhang, and A. Nee. 2001. A distributed multi-agent environment for product design and manufacturing planning. International Journal of Production Research, 39(4): 625-645.

Swink, M., S. Talluri, and T. Pandejpong. 2006. Faster, better, cheaper: A study of NPD project efficiency and performance tradeoffs. Journal of Operations Management, 24(5): 542-562.

Tani, G. and B. Cimatti. 2008. Technological complexity: A support to management decisions for product engineering and manufacturing. in Industrial Engineering and Engineering Management, 2008. IEEM 2008. IEEE International Conference on. 2008.

Tatikonda, M.V. and S.R. Rosenthal. 2000. Technology novelty, project complexity, and product development project execution success: a deeper look at task uncertainty in product innovation. Engineering Management, IEEE Transactions on, 47(1): 74-87.

Tatikonda, M.V. and G.N. Stock. 2003. Product technology transfer in the upstream supply chain. Journal of Product Innovation Management, 20(Compendex): 444-467.

Tchankova, L. 2002. Risk identification - basic stage in risk management. Environmental Management and Health, 13(3): 7.

TD-1-12, I.T. 2008. Risk Identification: Integration & Ilities (RI3) Guidebook, D.o. Defense, Editor Department of Defense. 73.

Technology, N.I.o.S.a. 2002. Risk Management Guide for Information Technology Systems, National Institute of Standards Computer Security Division: Gaitherburg, MD. 54.

Terwiesch, C., C.H. Loch, and A. De Meyer. 2002. Exchanging preliminary information in concurrent engineering: Alternative coordination strategies. Organization Science, 13(4): 402-419.

Thomke, S.H. 2003. Experimentation Matters. Boston. MA: Harvard School Publishing. 307.

Thompson, P.A. and J.G. Perry. 1992. Engineering Construction Risks, a Guide to Project Risk Analysis and Risk Management. Dordrecht: Kluwer Academic Publishers.

Tornatzky, L.G. 1982. Innovation characteristics and innovation adoption-implementation: A meta-analysis of findings. IEEE Transactions on Engineering Management, 29(1): 28.

Troy, L.C., T. Hirunyawipada, and A.K. Paswan. 2008. Cross-Functional Integration and New Product Success: An Empirical Investigation of the Findings. Journal of Marketing, 72(6): 132-146.

Ullman, D.G. and B.P. Spiegel. 2006. Trade Studies with Uncertain Information, in Sixteenth Annual International Symposium of the International Council On Systems Engineering (INCOSE), INCOSE, Editor INCOSE.

Ulrich, K. 1995. The role of product architecture in the manufacturing firm. Research Policy, 24(3): 419-440.

Vandani, B., S.M. Mousavi, and R. Tavakkoli-Moghaddam. 2011. Group decision making based on novel fuzzy modified TOPSIS method. Applied Mathematical Modelling, 35(9): 4257-4269.

Ward, S. and C. Chapman. 2008. Stakeholders and uncertainty management in projects. Construction Management & Economics, 26(6): 563-577.

Williams, C.A., M.I. Smith, and P.C. Young. 1998. Risk Management and Insurance. Irwin: McGraw Hill.

Williams, T.M. 1999. The need for new paradigms for complex projects. International Journal of Project Management, 17(5): 269-273.

Wu, D.D., et al. 2010. A risk analysis model in concurrent engineering product development. Risk Analysis, 30(9): 1440-1453.

Wu, L.F., R. De Matta, and T.J. Lowe. 2009. Updating a Modular Product: How to Set Time to Market and Component Quality. IEEE Transactions on Engineering Management, 56(2): 298-311.

Xu, Q., et al. 2009. An analytical Kano model for customer need analysis. Design Studies, 30(1): 87-110.

Yang, B. and N.D. Burns. 2004. Management of Uncertainty Through Postponement. International Journal of Production Research, (42): 15.

Yassine, A., et al. 2003. Information hiding in product development: the design churn effect. Research in Engineering Design, 14(3): 145-161.

Yeo, K.T. and R. Yingtao. 2009. Risk management capability maturity model for complex product systems (CoPS) projects. Systems Engineering, 12(Copyright 2010, The Institution of Engineering and Technology): 275-94.

Yu, A.S.O., P.S. Figueiredo, and P.T. De Souza Nascimento. 2010. Development Resource Planning: Complexity of Product Development and the Capacity to Launch New Products. Journal of Product Innovation Management, 27(2): 253-266.

Zhang, H., W. Qiu, and H. Zhang. 2006. An Approach to Measuring Coupled Tasks Strength and Sequencing of Coupled Tasks in New Product Development. Concurrent Engineering, 14(4): 305-311.

ZHANG, Y.F., J. SUN, and A.Y.C. NEE. 2001. A distributed multi-agent environment for product design and manufacturing planning. International Journal of Production Research, VOL. 39(No 4): 625-645.

Zsidisin, G.A., et al. 2004. An analysis of supply risk assessment techniques. International Journal of Physical Distribution & Logistics Management. International Journal of Physical Distribution & Logistics Management, 34(5): 16.

ABSTRACT

FRAMEWORK FOR QUANTIFYING AND TAILORING COMPLEXITY AND RISK TO MANAGE UNCERTAINTY IN DEVELOPING COMPLEX PRODUCTS AND SYSTEMS

by

DARRELL WILLIAMS

August 2013

Advisor: Dr. Ratna Babu Chinnam

Major: Industrial Engineering

Degree: Doctor of Philosophy

In recent years there has been a renewed interest in product complexity due its negative impact on launch performance. Research indicates that underestimating complexity is one of the most common errors repeated by new product development (NPD) teams. It was concluded that the companies that successfully manage complexity can maintain a competitive advantage. This is particularly true of CoPS projects (Complex Products and Systems) which are defined as large-scale, high value, engineering intensive products and systems. Investment in CoPS projects continues to grow worldwide, with recent estimates placed at over $500B annually.

In this research we present methods to improve the planning and coordination of complexity and risk in CoPS projects to support launch success. The methods are designed to be consistent with systems engineering practices which are commonly used in their development. The research proposes novel methods for the assessment, quantification, and management of development complexity and risk. The models are initiated from preliminary customer requirements so they may be implemented at the earliest point in the development process and yield the most significant cost savings and impact.

The models presented are validated on a large-scale defense industry project and experimental case study example. The research demonstrates that development complexity and risk can be effectively quantified in the early development stages and used to align and tailor organizational resources to improve PD performance. The methods also provide the benefit of being implementable with little disruption to existing processes as they align closely with current industry practices.

AUTOBIOGRAPHICAL STATEMENT

I grew up fueling an interest in science and technology and decided in high school that I wanted to pursue a career in engineering. Shortly after graduation I began attending a local community college in hopes of one day finding the resources to transfer to a four year university. In 1990 I was offered that opportunity by enlisting in the US Air Force, where I served honorably for four years supporting space and missile operations off the coast of California. Over the next several years I was able to complete my goals of earning bachelor's degrees in both industrial engineering and business management, while gaining practical experience in electrical and communication systems technology.

Shortly after joining the military, I married my wonderful wife Catherine, which began a fantastic journey for us as we moved about the country living in such places as Colorado, California, and eventually back to Michigan where we both grew up. Over that time, we worked together, to accomplish our educational goals as we shared workspace on the dining table each night, followed by evening walks and bicycle rides to the beach. Thankfully, our journey continues today after twenty plus years together.

My engineering career began in 1994 in the automotive industry where I worked for several Tier 1 suppliers developing and launching electrical and mechanical products. At this time I traveled internationally to coordinate various manufacturing and design activities. Through the support of my wife I was able to continue my education into graduate school, eventually earning master's degrees in both industrial engineering (University of Michigan - Dearborn) and business administration (University of Colorado) by 1998.

In 1999 I continued my growth in product development as I transitioned from Tier 1 automotive to the OEM (Chrysler Corporation), working in such areas as: program management, product strategy, manufacturing, engineering, and international sales and marketing. During this time I continued traveling internationally to countries in Western Europe to coordinate product strategy events, negotiations, and various PD activities.

In 2008 I returned to college to pursue a PhD in engineering, to begin preparing for a more autonomous role in product development research, teaching, and consulting. During the same time period (which marked my fifteenth year in the automotive industry) I transitioned into the defense industry to broaden my experience in product development. Through the continued support of my wife and family I was able to successfully transition into the new industry while completing my doctoral work.

During my time in the defense industry I worked in various roles in PD including technical program management, systems engineering, process engineering, risk management, and general business development / proposals. It was my work in the defense industry that provided the foundation for my dissertation research into complexity and risk management.

Currently my research interests include topics related to integrated product development (IPD) and methods for improving the launch and sustainability of complex / innovative products. In the future my goal is to extend this research into new products and industries to provide useful, practical tools for launching complex products.