Value Stream Mapping and Earned Value Management: Two Perspectives on Value in Product Development

by

Ryan Brent Whitaker

S.B., Aerospace Engineering (2003)
Massachusetts Institute of Technology

S.B., Physics (2003)
Massachusetts Institute of Technology

Submitted to the Engineering Systems Division
in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Engineering Systems

at the

Massachusetts Institute of Technology

September 2005

Signature of Autho ....
Engineering Systems Division
August 19, 2005

Certified by ....
Dr. Warren Seering
Professor of Mechanical Engineering and Engineering Systems
Thesis Supervisor

Accepted by ....
Dr. Richard de Neufville
Professor of Civil and Environmental Engineering and Engineering Systems
Chairman, ESD Education Committee

(This page intentionally left blank)

Value Stream Mapping and Earned Value Management: Two Perspectives on Value in Product Development

by

Ryan Brent Whitaker
ryanw@alum.mit.edu

Submitted to the Engineering Systems Division
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Engineering Systems
at the Massachusetts Institute of Technology

ABSTRACT

The concepts of value and value stream are crucial to the philosophy of Lean, and a better understanding of how these concepts relate to product development (PD) is essential for the creation of a Lean PD strategy. This thesis focuses on value by looking at PD processes through two different value perspectives: Product Development Value Stream Mapping and Earned Value Management. Product Development Value Stream Maps (PDVSMs) were created for two different PD projects, and the tasks from the maps were analyzed for how they each create value. The official value measurement for the two projects, Earned Value Management System data, was analyzed and compared to the PDVSMs. This comparison of the two value perspectives proved valuable, as it showed that despite some misalignments, they are congruent. The comparison also highlighted several flaws in EVMS. Finally, a combined EVMS/PDVSM hybrid management tool is proposed and discussed.

Thesis Supervisor: Dr. Warren Seering

Title: Professor of Mechanical Engineering and Engineering Systems

Acknowledgements

First and foremost, this thesis would not have been completed without the guidance of Prof. Warren Seering. Dr. Eric Rebentisch and Prof. Deborah Nightingale have also helped to lead me to graduation. I thank the LAI support staff for their big help with all my little problems.

Without the cooperation of Company X, this research could never have happened. I was given incredible access to information and freedom to explore. Special thanks go out to Wayne, Dan, and Doug. You were crucial to the execution of this work.

In addition to my advisors, the other students in the LAI PD group (Christoph Bauch, Tomer Posner, Martin Graebsch, Jin Kato, and Josef Oehmen) really helped me formulate my ideas over the course of many discussions.

My fellow LAI grad students have been so helpful. Jess, Chris, and Vikram shared my thesis pain this summer, and I hope they all succeed in the future. Erisa, Justin, and Jason, you also kept things lively. Ok, fine! I hope you all do well too.

I thank the National Science Foundation for supporting me through grad school, and the coordinating official, Brima Wurie, for answering all my strange questions.

I thank Eda Daniel in ESD for helping me with the degree switch. Also, Barbara Lechner in Aero/Astro is an angel. I think she hides her halo in her hair.

I want to thank MIT in general. The past 6 years have been challenging, frustrating, cold, painful, memorable, and at the same time, wonderful. I have really changed while here, and I cannot think of any way that the change has not been for the better. It's really going to be tough to leave here for somewhere less academic. I will miss overhearing dorky conversations and knowing at least a little about what is being said. I will miss feeling like an idiot 6 days a week. I say something smart every once in a while. I swear.

I'd like to thank the people with whom I've played softball and football. You don't know how much those sports offered me release. Special thanks to the team I was on this summer. That was the first time I have ever been an active part of a team that won a championship of any sort. Also, thanks to Dr. Geiger for surgically repairing my knee so that I can play these games that I love so much.

I also want to thank the city of Boston. What a great place. If I could just relocate you to somewhere without snowstorms and insane humidity, I would be a Bostonian for many years. I especially would like to thank the 2004 World Champion Boston Red Sox. They took me on a rollercoaster ride every year, but last year especially was up and down. Many games were heart-stopping. I can't believe that comeback. You know, the greatest comeback in history. Papi, Manny, Varitek, Damon, Millar, Bellhorn, Mueller, Trot, Cabrera, Schilling, Foulke, Timlin, Pedro, Lowe, Youk, and who can forget the man

who stole the ALCS, Dave Roberts! After 6 months of gut-wrenching drama and nearly giving up hope, they gave me three weeks where I could not go anywhere without hearing about baseball. What bliss! No one got anything done, especially the days after those west coast games. From the soul-searing destitution resulting from Game 3 (CJ and I just turned the game off in the third), to the flicker of hope that built day-by-day and culminated with Johnny's slam in Game 7, it was the greatest. No one else has been able to change my emotions so incredibly. And then the sweep completed the joking prophecy I made after Game 3: "all we need is an 8-game win streak...against the hottest team in baseball, and then against the next hottest." And you gave me a great reason to conduct a great social experiment: out of 127 people I reached up my hand to on the night we won it all, 93 of them gave me a high-five. It's a great sign of how happy everyone was. Just in case you were curious.

I thank my mother for her love, support, and understanding. Thanks to you, Steve, for keeping the most important woman in the world happy.

I have to thank my awesome friends who put up with all my complaining about this thesis. They helped keep me going. Geeta Gupta, you know you are as close to a sister as I could have. I hope we will still be friends decades from now. Tim, 6 years, and I still haven't had a good reason to kick your ass. Impressive--that must mean you are a good man. Keep up the good work, and I want to see a video of you benching 500 some day. Deez and Barber, you are my best friends back home, and are just about the best people I know. Julia, thanks for the detailed proofreading and for making this summer wonderful in spite of the thesis stress. To the rest of my friends, who will NEVER read this, I won't bother to say anything that special about you, even though *most* of you deserve it. Sean, Clay, Aman, Martinez, CJ, Nelson, Darien, Damian, Barnthouse, Jones, Weathers, Phaedra, Flams, Nat, JimmyK, and Geoff—you all made my time here great.

Finally, I have to thank myself. Yes, this is weird, but dammit, I deserve thanks. I have somehow motivated myself to write this big thesis. Sometimes I think we all forget to appreciate ourselves. Thanks Ryan for getting me a master's degree!

Table of Contents

List of Figures

List of Tables

List of Abbreviations

PD	Product Development
PDVSM	Product development Value Stream Mapping
EV	Earned Value
EVMS	Earned Value Management System
MIT	Massachusetts Institute of Technology
LAI	Lean Aerospace Initiative
BCWS	Budgeted Cost of Work Scheduled
BCWP	Budgeted Cost of Work Performed
ACWP	Actual Cost of Work Performed
A-Spec	A-Level Specifications; System Specifications
B-Spec	B-Level Specifications; Development Specifications
ORD	Operational Requirements Document
TRD	Technical Requirements Document
SRS	Software Requirements Specifications
SDD	Software Design Document
TBD	To Be Decided
TBR	To Be Resolved
SDP	System D Prototype
SW	Software; Controller Software
HW	Hardware
HW1	Controller Hardware Item #1
HW2	Controller Hardware Item #2
MEE	Controller Mechanical Engineering Effort
IPT	Integrated Product Team
WBS	Work Breakdown Structure
FAIT	Fabrication, Assembly, Integration, and Testing

1 Introduction

1.1 Motivation

As the concepts of Lean originally developed by Toyota after World War II have proven to be powerful in improving efficiency in manufacturing systems, many have attempted to apply them in other areas of business. From health care to office paperwork, the overarching philosophy of Lean—eliminating waste and creating value—has proven useful for allowing organizations to do more with less. In this age of reduced defense budgets, the Lean Aerospace Initiative (LAI) at MIT has set out to help the US aerospace industry transform to a more Lean state. This is a slow process, and many of the specifics of how to make this transition are still to be uncovered.

In order for industrial firms to truly be Lean enterprises, the lessons from Lean manufacturing must be transferred both upstream and downstream to make the value flow continuously from raw materials and ideas into products customers want. In the upstream processes, market identification and product development (PD), more impact can be made on the final outcome of the value flow than can be in manufacturing or testing. In spite of the huge potential of PD, industry research indicates that for it is rarely executed optimally. A *typical* aerospace work package in the product development process is either idle or having non-value adding work performed on it 77% of the time (McManus 2004). Thus, PD is ripe for the implementation of Lean, but with its long batch times, uncertain processes, iterations, intangible products, and non-linear flow, PD *seems* to be too far removed from manufacturing for Lean to apply. In order to understand how Lean can help improve PD, many areas must be explored.

1.2 Problem Statement

With this thesis, I aim to help make Lean more practical within complex product development. I am attempting to discover if creating a Product Development Value Stream Map (PDVSM) is effective for learning about a PD process. I seek to learn whether a PDVSM helps to identify value and its flow and to better understand how value is created in the product development process. I also want to know how well a current system used for measuring value, the Earned Value Management System (EVMS), compares to a PDVSM and associated value analysis.

1.3 Thesis Overview

This thesis describes two case studies carried out at a LAI member company, referred to as “Company X”. One project involved the development of requirements for a software subsystem, and the other dealt with a hardware prototype within a spiral development process. For each project, its context, process, and issues are explained, and one or more PDVSM is presented. The process steps of these maps were analyzed for the kinds of value they created within the project. Data from the project’s official measures of value, EVMS, are analyzed and compared to the PDVSM and value analysis. Interesting points and disconnects between these two perspectives on PD value were identified and examined.

In the following chapter, the major concepts of Lean are presented with a discussion of the product development process, some problems endemic to PD, and research into the implementing Lean in PD. The third chapter gives a little background about LAI and the general research strategy of the LAI Product Development group, along with the specific strategy for this thesis research. After that, a chapter for each of the projects studied form the heart of this thesis. The sixth chapter collects the research findings, lessons learned about performing this type of research, and direction for future research.

2 Background

The following chapter offers a brief introduction to the concepts of Lean and product development, as well as prior attempts to join the two. It concludes with a short explanation of the Earned Value Management System, which played a large part of this thesis.

2.1 Lean

In short, Lean is an overarching philosophy of creating value in an efficient manner and eliminating waste. It can be applied to various parts of business to achieve unimaginable gains in productivity that can help businesses thrive. Any attempt to fully explain both the simplicity and intricacies that create this powerful system of thought in a few pages will be woefully inadequate. A great deal of literature exists that does a much better job of explaining Lean, and a taste of this follows.

The core concepts of Lean were first brought to attention in the 1990 book *The Machine That Changed the World* (Womack, Jones, and Roos, 1990), based upon research done by the MIT International Motor Vehicle Program. In their book, Womack, Jones, and Roos chronicle the rise of Lean production at the Toyota Motor Corporation in the decades after World War II and describe the unprecedented efficiency it enables. Of course, Toyota did not refer to these concepts as "Lean" but merely as the "Toyota Production System" (TPS), which had evolved due to lack of resources and labor in the post-war manufacturing environment. The standard practices of mass production could not be used, and Toyota had to be creative in finding a new way to do more with less. While the Toyota Production System was bolstered by many cultural influences and by ideas from hundreds of employees, Shigeo Shingo and Taiichi Ohno had the greatest influence on it. They are considered the fathers of TPS.

In their successful 1996 follow-up *Lean Thinking*, Womack and Jones sought a way to help the success of Lean within a production environment to be extended to an enterprise level. They defined five principles that were key to Lean production's success: value, the value stream, flow, pull, and perfection. These principles are thought to be applicable outside of production, and this has been the goal of much research (Womack and Jones, 1996).

They define the "value" of a product or service as "a capability provided to a customer at the right time at an appropriate price, as defined in each case by the customer" (Womack and Jones, 1996, p. 353). In other words, the value of a product is how much it is worth to the customer to have the product where and when they want it and at what level of quality. By focusing on customer value (and thus what the customer is willing to pay for), an enterprise can eliminate all activities that do not contribute to its creation (those the customer *isn't* willing to pay for), and thereby save time and money. All of Lean is based upon delivering customer value.

The "value stream" is "all the actions, both value added and non-value added, currently required to bring a product from raw material to the arms of the customer or through the design flow from concept to launch." (Morgan, 2002) The value stream often encompass many business enterprises (such as designers, suppliers, and final assemblers), and by looking at the entire value stream, one can identify how to improve the process of turning an idea into a product. According to Womack and Jones, in analyzing value streams, all activities will fall into one of three types:

Value adding: Contributes to the value of the product
Necessary, but non value adding: Cannot be avoided
Non value adding: Should be eliminated immediately; waste.

To understand these activities, let's think about painting a new car. Paint protects the frame and body of a car from the elements, and provides aesthetic appeal to most car owners. While some customers may insist that they do not care about the paint job on their car, most people find the color to be an important consideration when purchasing an automobile. Whether for the aesthetic or the protective purpose, each customer gets value from having paint on their car. Thus, the act of applying paint to the car inherently adds value to the product, and thus is a value-adding activity. After a while, all paint-spraying equipment tends to get clogged if not routinely cleaned and maintained. Thus, cleaning the sprayer is an activity that does not contribute value directly to a product, but is necessary for the painting system that in turn adds value to many vehicles; it is necessary non value added. Many small tasks may be associated with the painting job which are not really needed. Perhaps the paint arrives from the paint manufacturer in large drums that must then be poured into smaller containers that can be inserted into the painting system. If the automobile manufacturer were able to coordinate with its paint supplier to deliver the paint in smaller containers that fit easily into the painting equipment, the non value added task of pouring the paint into a new container could be avoided.

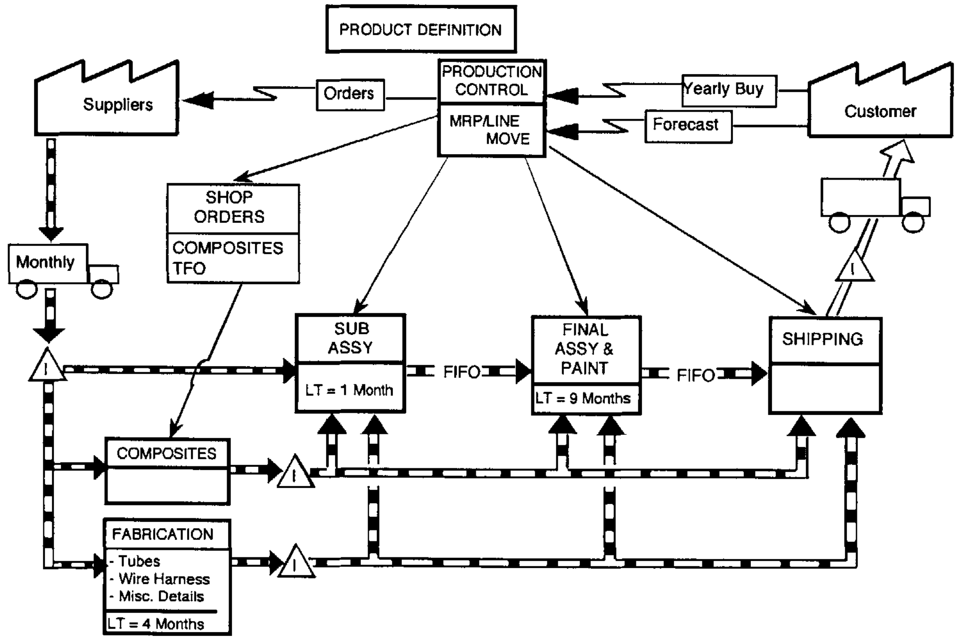


Figure 2.1. Sample *Learning To See* Value Stream Map (Millard, 2001)

One of the best ways to visualize and analyze the value stream of a process is to create a value stream map (VSM). Value stream mapping is a technique for drawing the value stream that allows one to see the flow of information and materials between activities. The fundamental strategy of using a value stream map involves first creating a map that reflects the current state of the process being mapped. This map is then analyzed for waste and value creation, and a future-state map is created, which represents how the process *could and should* operate. An improvement plan can then be created to enable the transformation from the current state to the future state. As the process is improved over time, the current state can be mapped again, and compared with the future state map again to redefine the improvement strategy. Value stream mapping has proven to be incredibly valuable in the world of manufacturing. The mapping format and techniques developed by Rother and Shook (1999) in their *Learning to See* have proven to be useful for identifying waste and enabling process improvement. A sample of a *Learning to See* map is shown above in Figure 2.4

Once the value of a product has been assessed, and its value stream analyzed, focus should be applied to making the steps in the product creation process *flow* smoothly. That is to say that the output from one process task should move smoothly into the next task. When flow is not achieved, parts and information to be processed pile up as inventory in between tasks. In a manufacturing environment, these parts take up space and must be kept in order for when they will be processed. This piling up of inventory is but one of many kinds of impediment to flow known as *waste*. Ohno identified seven kinds of production waste, and Womack and Jones added one more (Womack and Jones, 1996). These will be further explored later.

Lean is not the first system to recognize the value of flow, as Henry Ford employed the continuous flow principle in his moving assembly line in the early twentieth century. Ford was not being Lean but just very efficient with his mass production system, as the methods he used "only worked when production volumes were high enough to justify high-speed assembly lines, when every product used exactly the same parts, and when the same model was produced for many years" (Womack and Jones, 1996, p. 23). The concept of flow has influenced Lean thought and practices over the years, and helped encourage the use of small batch sizes and a system of cards and bins to manage internal inventory (known as "kanbans"), as well as the just-in-time inventory system (which has incorrectly become thought of as synonymous with Lean).

Once a value stream has been aligned such that the value can flow from one step to the next, the concept of "pull" can be applied to it. This concept states that an upstream task should not be performed until a downstream task asks for its output. In other words, any given task should behave as if the next task in sequence is its customer, and should only provide an output when that task asks for an input. Pull is the opposite of "push"-based systems of production. In a push system, an upstream task is performed and its output sits around until it can be used by the downstream task. This encourages inventories building up, is often the result of manufacturing being performed in large batches (which mass-production systems view as the most efficient way to make an item). When in a manufacturing environment, pushing a large batch of items onto a downstream task not only creates inventories, but can lead to a great deal of waste if a problem is found with the quality of the output items. In the ultimate flowing pull-based system, an order would be placed by the customer and placed at the end of the manufacturing line. The operator of the last task would ask the operators of upstream tasks for the exact items needed for his task. This pattern would continue, and each task would output exactly the number of items needed by its downstream task, in what has been dubbed "single-piece flow".

The final key concept in Lean is the pursuit of perfection. All wastes cannot be eliminated overnight and production systems cannot be made to perfectly flow the first time. Lean requires an enterprise to never be satisfied, to always seek ways to perform better. As long as progress is being made, with perfection of the value-delivery system the goal, the journey to a Lean state is ongoing. Toyota, by far the recognized exemplar of a Lean enterprise, continually strives to improve itself. A company just starting the Lean journey should not think that it can "become" Lean instantly. The implementation of Lean is a considerable undertaking. Oftentimes, in order to properly eliminate wastes and properly align the value stream such that the value can flow at the pull of downstream activities, a redefining of the work to be done is required. Departmental structures that may have existed for years (which made sense historically but now cause waste) may need to be broken down, with a new organization of functions and product teams formed. A fundamental cultural shift must be made as all employees must embrace the desire to make not only their job better, but to make the entire value stream flow better.

In their *Decoding the DNA of the Toyota Production System*, Spear and Bowen suggest that all the employees of the TPS form a "community of scientists", with each using the scientific method every day in order to solve problems in the pursuit of perfection (Spear and Bowen, 1999). They claim that in addition to the Lean principles, four fundamental rules are implicitly employed at Toyota. These rules explain the paradox of how Toyota uses rigid specification to create operational flexibility and adaptability. The first of these rules states that all work must be specified as regards content, sequence, timing, and outcome. By doing this, workers know exactly what is expected and how they are expected to do it. Also, with detailed specification of these facts, they are able to experiment to perfect their tasks. The second rule states that all customer-supplier relationships must be standardized and direct, allowing for more accountability between business partners and reducing variability within the value creation process. The third rule states that the physical arrangement of the production line must allow every product and service to be able to flow in a simple, direct, and specified path through the plant. The fourth rule states that any improvement to the specifications dictated by the three above rules must be performed using the scientific method, under the guidance of a teacher, at the lowest level possible. Observations must lead to hypothesis, which are systematically tested until they match observation. Any time a problem cannot be solved at the lowest level, it must go up to someone on a higher level within the organization who may have more insight into solving the problem. One systematic way that problems are solved is the use of the "5 'Why's" approach. Whenever a problem is observed, the question of "why?" is presented and answered. The question is then repeated with respect to the previous answer until there is no answer. At this point, the employees have more insight into where to begin their experimentation. Usually "why?" need not be asked more than five times to uncover any problem. Thus, the TPS can be constantly improved by practicing the scientific method, which is enabled by rigid specification. This fact embodies the Lean principle of pursuing perfection. (Spear and Bowen, 1999)

On the whole, Lean has made great strides since the 1990 MIT book. The concepts that make it work have been identified, and many companies have started to see the positive synergistic effects of employing them all as appropriate. As Lean has been transferred from the relatively high-production automotive industry to other industries, organized research has been needed to determine how to properly transfer these ideas to best fit each industry. Lean principles have spread and many related techniques are being applied in such diverse areas as supply chain management, accounting, office paperwork, and even health care. However, the application of Lean in these areas and others is in its infancy. Product development is an area particularly ripe for the application of Lean principles and techniques. Researchers are currently struggling to find the best ways to apply Lean to PD with early indications of success.

2.2 Product Development

Product development is a wide area, with a vast amount of literature related to it. This section does not attempt to comprehensively represent this literature, but provides the reader with an understanding of what the context and basic steps of the process are, along with explain how the process is different when dealing with large, complex systems such as those that are often found in the aerospace/defense industry.

At some point, some people sat down and really thought hard about how to make each and every one of the products that one encounters—a telephone, a computer, a car, a blender, a cordless drill, or an ink pen. The result of the product development process is evident all around—from the ergonomic shape of a stapler to the sound output from a pair of speakers— in the form and function of the products.

In the simplest view possible, an enterprise is a group of people that take ideas and turn them into money by creating and delivering products, processes, services, and/or systems. While it would be foolish to imply that an enterprise is such a simple entity, there really are two major components of the above statement: *creating* the product and *delivering* the product. Product development is a general term that describes the creation process: taking an idea for a product and turning it into a set of instructions for its manufacture and assembly. Manufacturing, sales, and distribution are responsible for the delivery of the product.

Ulrich and Eppinger (1999) explain the several major steps of general product development. First, the needs of the customer must be identified and then used to establish target specifications. These needs and specifications can be related to the product appearance, physical characteristics, functional characteristics, or overall performance, and are used to drive the development of the product. Once the product's goals are established, system-level design is performed. In this stage, the system architecture is selected and decisions are made about how to decompose the product into subsystems and components that can work together to deliver the desired functionality. Next comes the detail design, in which the product is fully defined. Drawings are made and software is laid out. This stage really is the heart of product development, and is where most engineering effort is expended. While several important high-level decisions were made during the system-level design, it is during the detail design that literally hundreds of finer decisions must be made to fully define the product. Next comes testing and refinement, in which prototypes are made and the product's design is finalized. Finally, during production ramp-up the manufacturing system is set up and prepared to make the products and the distribution channels are established. (Ulrich and Eppinger, 1999)

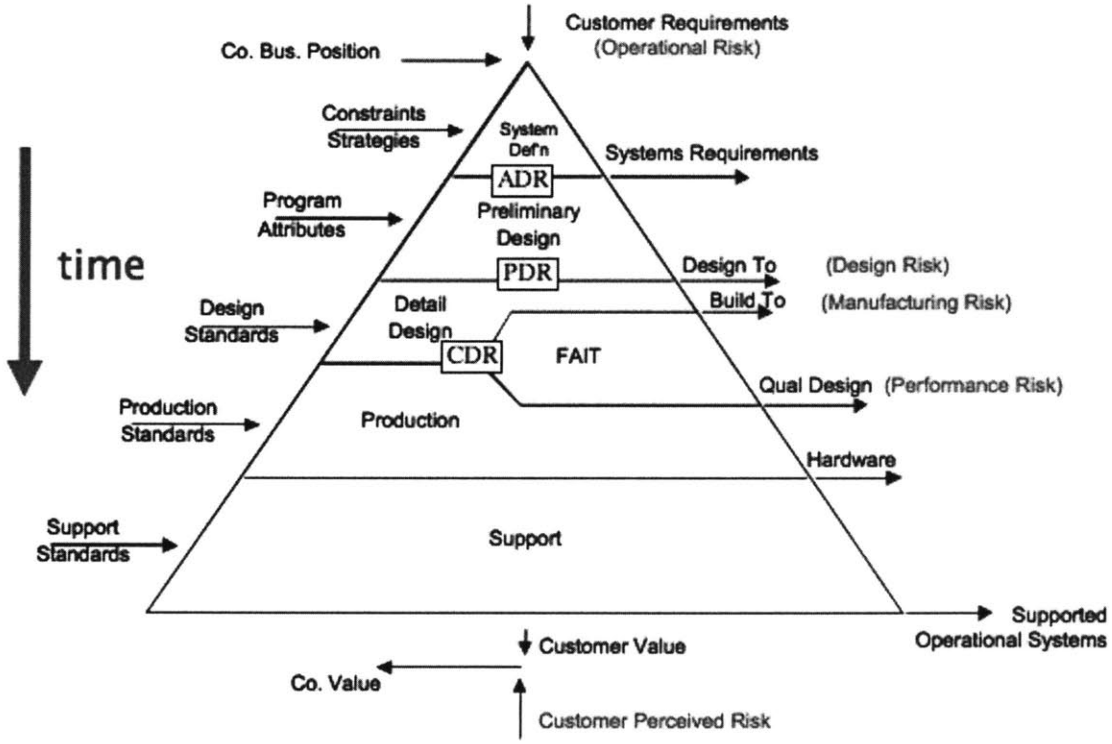
Product development within the aerospace/defense industry differs slightly from the above general process. First, the products in this industry are generally large and complex. Many systems literally push against the laws of physics, and many need state-of-the-art technology to do so. The complexity of most of these systems has many ramifications. In order to be manageable, the development must be broken down into understandable sections. Managing the large development efforts is usually done with many layers of management and leadership. As the complexity of a system development increases, so does the number of people required for its execution. The more people that are involved in a project, the greater the need is for increased communication and management effort. Also, the complexity of these systems makes them very expensive, and in order to make the development economical, large orders are often placed. With large contracts for expensive products come very interested customers. Since most customers are large enterprises in and of themselves, such as national governments or airlines, that are committing large sums of money to the development and delivery of products, they want to be assured the desired functionality will be delivered. Information must flow in two directions for the development to be successful. First, the requirements for the product must be created and unambiguously passed on down to the designers. Developing and communicating the requirements is a difficult process, and is often done in a series of requirements documents, with each being more specific about what the product system must do. At certain points in the course of the design, design reviews are held for the product development leadership team (and often, also the customer) to review the status of the design and the design process.

Meeting customer needs is a crucial component for success in the world of product development. Consumer products are designed and then manufactured and sold, in hopes that they meet the needs and wants of the customer. Due to the complexity of aerospace systems, fully defining exactly the needs of the customer (and other system development stakeholders) is a difficult process. These needs are reflected by statements of the desired system functionality called “requirements”. These requirements serve to communicate the functions desired by the product’s stakeholders to the engineers that must implement them. The requirements derivation process is often very difficult, and is often performed by systems engineers. This process begins with the definition of overall system operational characteristics by the customer. These operational requirements are then turned into a set of technical requirements. As the overall system design is derived from these technical requirements, high-level requirements are made for each of the subsystems. These high-level requirements are used to begin exploring the subsystem design and are thereby expanded into detailed specifications that fully define the requirements of each subsystem. Only after the system’s requirements have been fully defined should the detailed system design begin in earnest, lest work be done on some area that contradicts a later-developed requirement. Developing and managing these requirements throughout the entire development process is often so complicated and so essential for the successful design of a complicated system that it often requires the work of systems engineers, software and hardware engineers, and senior program leadership.

As the requirements reach various levels of maturity, they must be communicated back to the customer to make sure they reflect what is really desired by the stakeholders. In order to do this, and to communicate the status of the product's design, design reviews are held throughout the process. These reviews are often large meetings in which the status of the design and the requirements are presented, along with budgetary and schedule information regarding the direction of the next steps of the development. Usually, two such reviews are held: a Preliminary Design Review (PDR) and a Critical Design Review (CDR). These reviews divide the "detail design" described in the Ulrich and Eppinger process into two phases: preliminary and detail design. The PDR is held at the end of the preliminary design phase, and CDR is held at the end of the detailed design. These design reviews allow the stakeholders to comment on the design, to point out issues, and to request changes. Sometimes, a another design review will be held. Known as the Advanced Design Review (ADR), this review usually comes before the others, at the end of the system design, and reviews the requirements and high-level system design so that the preliminary design can begin.

The figure below shows the lifecycle of a typical aerospace product. This graphic is a little tricky at first, as it displays a lot of information in an atypical fashion. The flow of time is downward, and the triangular shape indicates the increasing value delivered by the development. This triangle is divided into many layers that indicate the stages of the process, each with inputs on the left and outputs on the right. First the company developing the system will work with the customer to develop the requirements and create the overall system design. After this stage is where the ADR might be held. Next, the preliminary design begins, further defining the requirements and establishing the major parts of each subsystem design. This preliminary design is presented at the PDR, and offers a solid basis for the detail design effort. The detail design results in a fully-defined design for the system, its subsystems, and all components. After the design is reviewed and approved, fabrication, assembly, integration, and testing (FAIT) is performed. This step is akin to the "testing and refinement" expressed by Ulrich and Eppinger's general process (Ulrich and Eppinger, 1999) , and is basically a prototyping phase to ensure that the system can be produced and will function as required before full-scale production begins. Within the FAIT process, components are produced and individually tested, as are assemblies made of them. Also, the software and hardware are integrated and tested, and the overall system functionality is ensured. After FAIT is complete, production of the system begins, and the system is delivered to the customer. If more than one unit of the system is contracted, they are usually delivered in waves. The development enterprise will continue to offer support to the customer to help fine tune and maintain the system. As this process proceeds, more value is added to the system and the amount of risk of not receiving a functional system perceived by the customer decreases. The value of the process lies not only in the receipt of the functional system but also in the progressive reduction in risk. Within this product lifecycle the PD process can be considered to begin at the top and end before production. This point is where the creation of the system ends and the delivery of functionality begins.

Figure 2.2 Product Development Process in the Aerospace and Defense Industry (adapted from Millard, 2001)



2.3 Problems in PD

The established process for product development in aerospace and defense has been developed over the past decades, but having been around a long time does not mean it is perfect. Oftentimes, system developments within this industry run over budget, behind schedule, and/or under-perform the requirements. When viewed from a Lean perspective, it is easy to understand why this is the case.

Various research into the aerospace industry has found some facts that astound the outsider but are no surprise to anyone within the industry. Figures 2.2 and 2.3 below show the results of two separate studies into how engineers spend their time. The first of these graphs shows the results of a survey. Engineers were asked to estimate how much of their work effort (in time-card hours) was spent adding value, how much was spent in support tasks, and how much was wasted, and they reported 40% was wasted and only about 30% was value-added. (McManus, 2000) Another, more formal study found that 30-40% of engineering effort is wasted (Joglekar and Whitney, 2000). Figure 2.3 shows this analysis of time spent in a different light. Instead of looking at it from the point of view of the individual engineer, this study followed the path of an individual work package as it passed from one engineer to another in the course of its development. It was found that 60% of the time, a job is sitting idle, waiting to be addressed by the next engineer (Young, 2000).

Figure 2.3 Value Assessment of Aerospace Engineering Activity (as % of hours worked) (McManus, 2004)

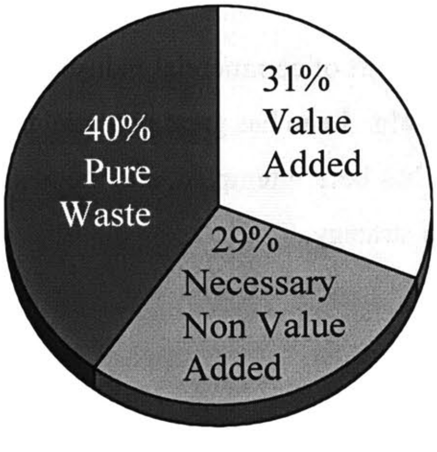
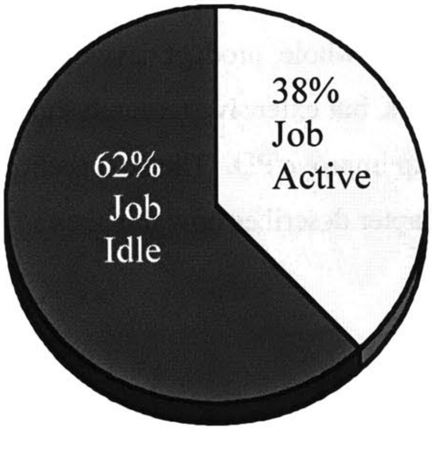


Figure 2.4 Assessment of Activity Performed on Aerospace Work Packages (as % of actual hours) (McManus, 2004)



Assuming that these study results are *typical* of various product development tasks, they have been combined into a notional result shown in Figure 2.4. The data combines to indicate that value is being added to a work package only about 12% of the time! Obviously if these numbers are representative, then PD needs a great deal of help to become more efficient! These numbers have been supported anecdotally by many Lean events held at LAI member companies in which 75-90% idle times were identified in bottleneck processes (McManus, 2004).

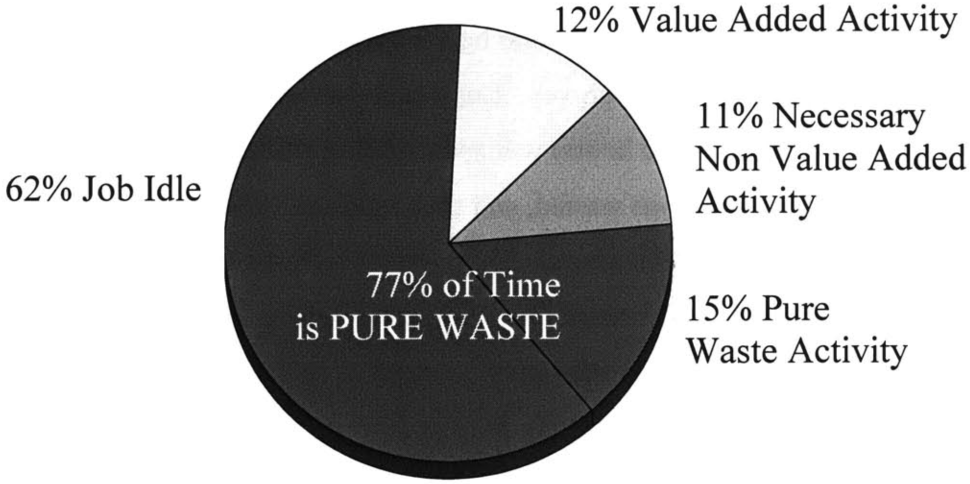


Figure 2.5 ***Notional*** **Value Assessment of a** ***Typical*** **Aerospace Engineering Work Package (McManus, 2004)**

On the whole, product development has been an essential part of commercial industry for years, but extensive research shows that it is in need of help. Lean has great potential to help improve PD. The next section chronicles how this has been attempted, and the next chapter describes how this thesis fits within the research strategy.

2.4 Lean and PD

Lean enterprises must design and manufacture the right products in an efficient manner. To deliver the right products, enterprises must create products that meet the needs of all stakeholders. Each enterprise must be well-integrated such that the PD process adds value throughout the product lifecycle and for all stakeholders. To be efficient in delivering the systems, most enterprises have begun improving their manufacturing capabilities using Lean concepts. The final piece of the puzzle is efficient development of systems. By applying Lean to PD practices, it is hoped that PD process efficiency and consistency can increase. Specifically, the goal is to make PD processes have shorter cycle times and be cheaper. (McManus, 2004)

Finding ways to make PD more efficient and effective is the goal of this research. Several people have made great strides in pursuing how to make PD more Lean, and have consistently encountered several differences between engineering and manufacturing that make application of Lean techniques to product development difficult. Morgan (2002) notes five major differences between PD and manufacturing that make it difficult to apply Lean to PD:

1) PD deals with the flow of information, not physical entities
2) PD time measures are much longer and often ill-defined
3) Much PD work is intangible, and it is more diverse and less predictable.
4) PD information flows are often non-linear and multi-directional, with iterations, reciprocal flows, and therefore more communication needed
5) PD needs more people, with a more diverse set of knowledge and skills

Slack(1998) was the first to describe how the Womack and Jones's Lean principles could all be applied to the PD process. He found that the value, value stream, and flow principles needed to be altered to make sense with respect to PD. He questioned the application of the pull principle to PD, but supported the generality of the perfection principle. He also presented a discussion of the analogs to manufacturing waste within product development and how manufacturing flow techniques can be applied to a product development context.

2.4.1 Value

In his analysis of all the Lean principles, Slack focused upon the value principle. He suggested that the value of product development can be decomposed in various ways. First, it can be broken down into customer value, employee value, and company shareholder value, and he showed a notional relationship between these three. He further decomposed the customer value into three major components quality, cost of ownership, and time. This decomposition is shown below in Figure 2.6. Additionally, he defined a simple equation for estimating the value of product development. (Slack, 1999)

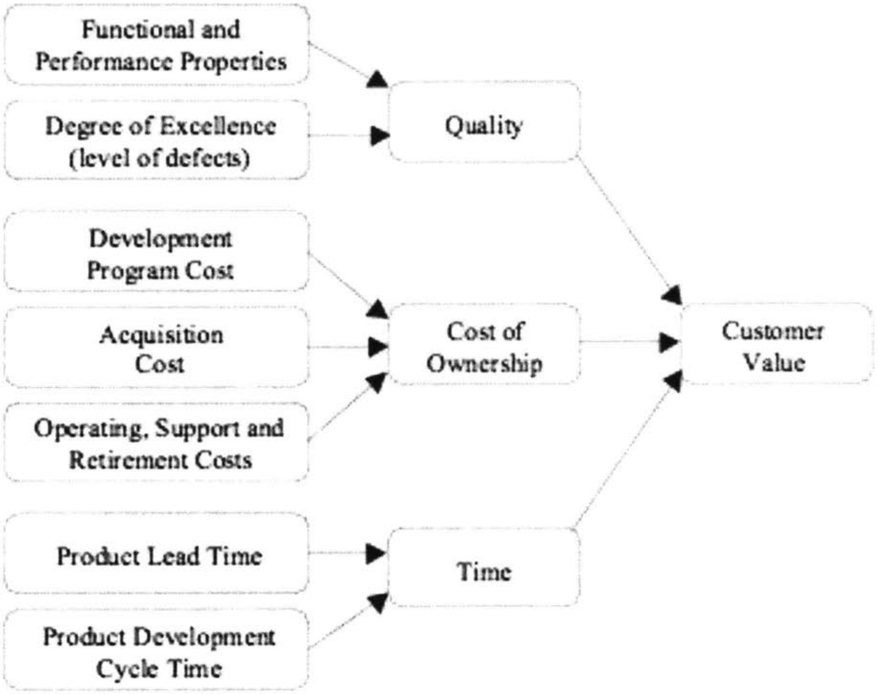


Figure 2.6 Slack's Decomposition of Customer Value (Slack, 1999)

Browning (1998 & 2003) has suggested that the status of a product development process can be measured by tracking the uncertainty of the process output. As risk goes down, information about the product, and therefore value, increases. He also claims risk exists in the forms of technical, cost, schedule, technology, market, and business risks and that reduction in risk can be used as a measure of value. He created highly detailed equations for analyzing the uncertainty associated with a certain task, as well as the overall risk this caused. Finally, he also argues that product value is a function of performance, affordability, and timeliness.

Chase (2001) is highly suggested reading for learning about value in product development. He created a framework for value creation within the PD process based upon his analysis of many previous definition of value and several case studies. He agreed with Slack's first value decomposition, suggesting that the value of product development includes delivering a successful product to the customer, profits for the shareholder, and lifetime satisfaction for the employee. The elements of his framework were tasks, resources, environment, and management, and each of these was divided into various levels of value attributes—how a task contributes to value, shown below in Table 2.1. By combining these value attributes, one can construct the value of a PD process. Additionally, he found that the value of product development is more easily understood when broken down into *product* value and *process* value. His research also showed that development efforts that used earned value management had fewer tasks behind schedule than those that did not. Chase's conceptual framework was based on the idea that as product development tasks accumulate information, risk is decreased and thus, product value increases. Once the product is complete and all value accumulated, it can then be exchanged for money with the customer. He also surmised that product value is the nexus of a product's cost, schedule, and performance (similar to Slack's Time, Quality, and Cost of Ownership decomposition).

Oehmen (2005) collected the ideas from Chase, Slack, and Browning (among others) to create a framework of value within PD. Similar to Chase, Oehmen separates value into *product value* and *PD process value*. For each of these, *Time*, *Cost*, and *Quality* are the generic goals. Table 2.1 shows these six major categories of value. These categories are used for the value analysis of the value stream maps later in this thesis.

Table 2.1 Oehmen PD Value Framework (Oehmen, 2005)

	Time	**Cost**	**Quality**
Product	(lead time)	(lifecycle cost)	(performance)
Process	(schedule adherence)	(budget adherence)	(conformity to standards)

2.4.2 Value Stream

The use of values stream maps (VSM) within PD has gained popularity in the past few years. Millard (2001) analyzed various techniques that claim to show PD process information. He found that Gantt and Ward charts were better at representating for the timing of tasks, but that Process Flow Charts, *Learning to See* diagrams, and Design Structure Matrices were the best for analyzing the process.

By far, Morgan (2002) has made the most complete attempt to understand how to employ Lean in PD. He explored the application of Lean in depth and proposed a systems approach to creating a "High Performance Product Development Process". To make such a system, an enterprise must have an integrated, coherent system consisting of goal-aligned, mutually-supportive subsystems: human, process, and technology. For the human subsystem to work, it must effectively integrate many groups of diverse technical specialists. He suggested that improving the process subsystem was the first step to achieving this PD system, and his research focused upon this. His analysis was based upon an analysis of the Toyota Production System and the product development processes Toyota uses. He argued for the usefulness of value stream mapping and developed his own map style.

Morgan (2002) elucidated many of the issues surrounding the use of VSMs in PD. In addition to identifying many major problems with PD, he proclaimed that value stream mapping is very powerful. He says that VSM not only should be used as the first step for optimizing the PD process, but also that it enhances the use of other optimization tools, strengthens the grasp of the overall process (discouraging local optimization & encouraging finding leverage points), is useful in synchronizing concurrent and cross-functional tasks, makes cause-and-effect relationships more evident, and serves as a common language between functional organizations for discussing the PD process. He claims that "VSM is perhaps the most effective tool for improving PD concurrency because of its ability to display multiple functional activities simultaneously at different levels of detail". For each of the five major differences between PD and manufacturing he identified, he developed countermeasures for use in making VSMs for PD. He suggested separating the value stream map into horizontal layers, each representing a different functional group, all aligned with a single time scale. He also proposed that this time scale could change as one looked at the map at different levels of detail. Thus, he plans to have multiple levels of detail to make a map that can be "zoomed". Furthermore, he created his own set of symbols for representing activities, information flows (differentiating between push and pull, and various information types), events, delays, and activities. He proposed that a data box be attached to each task that displays a great deal of information about it. Finally, he gave steps for creating and analyzing the current state map, as well as suggestions for creating a future-state map.

McManus's (2004) *Product Development Value Stream Mapping (PDVSM) Manual* offers a guide for practitioners to make a PDVSM. It takes many of the lessons learned from Millard's and Morgan's work, along with those from numerous industry case studies to create a near-cookbook for detailing the creation of a current state map, analyzing it for waste and value, and for creating a future-state map. An example of one these PDVSMs is shown on the following page as Figure 2.7. While this Manual offers a great methodology and set of symbols, it is limited in scope and scale to the mapping and improvement of a single, definable component-level hardware design PD process. Thus, this cookbook approach had to be modified for use on the projects in Chapters 4 and 5.

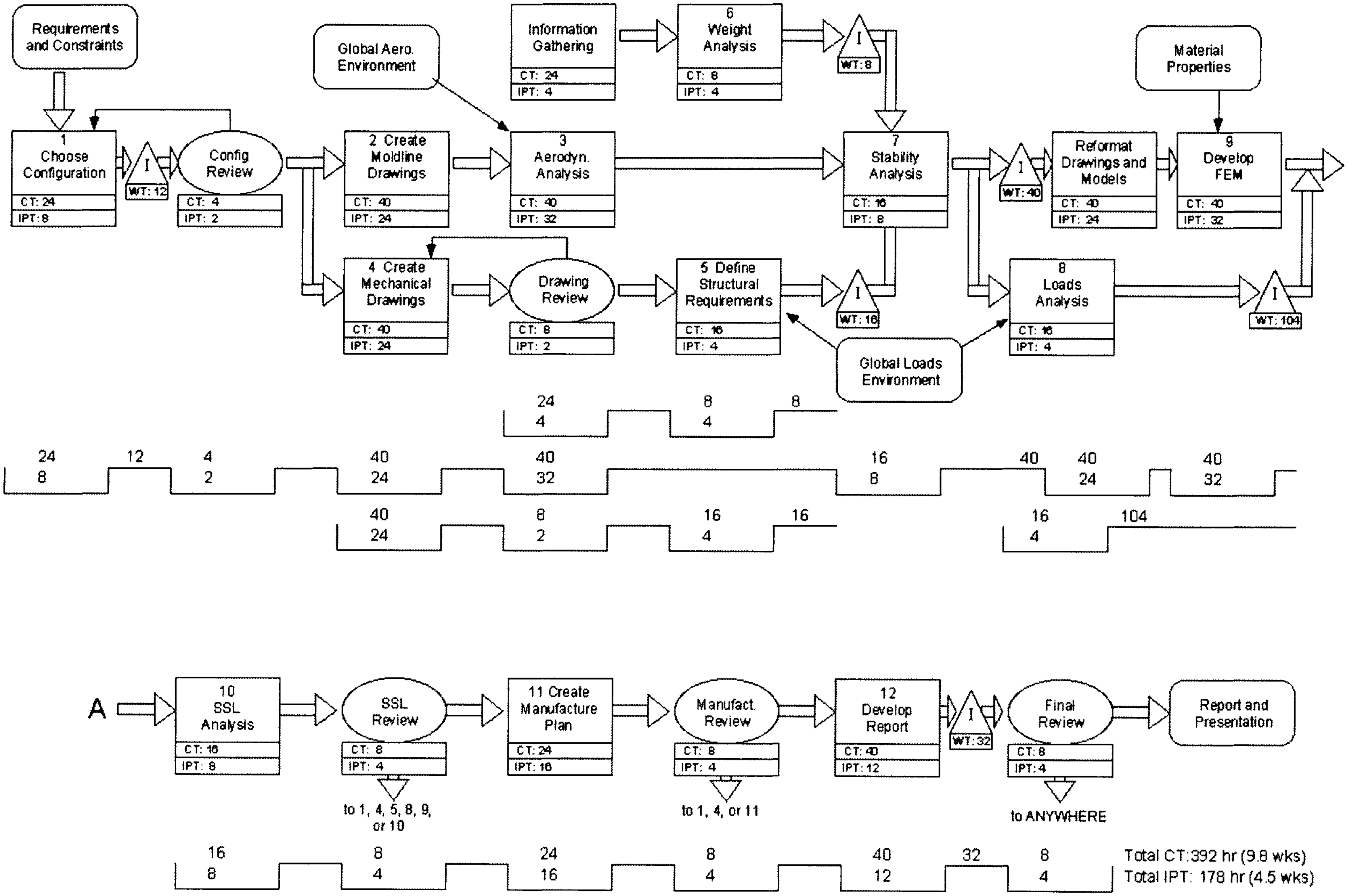


Figure 2.7 Example VSM from PDVSM Manual (McManus, 2004)

2.4.3 Flow

The flow of value within PD is more difficult to analyze than it is within manufacturing. Value flow within PD has been with the flow of information. According to Sawhney and Parikh (2001), "In an information economy, improving the utility of the information is synonymous with creating economic value. Where intelligence resides, so does value." Thus, most focus on flow within PD has been upon how information flows between project tasks and the impediments to the flow, or waste. In addition to his work on value stream mapping, Millard (2001) described the quality of information flow in terms of form, fit, function, and timeliness. Graebsch (2005) studied the flows of information within PD processes, defining, categorizing, and analyzing the quality and content of information and its transfer between people. He found that more information flow is not necessarily better, but that high-quality information transfers, especially those planned in advance, are more likely to require less future communication and do a better job of transferring information value. He also systematically developed the requirements for a Lean Product Development Display. Also, Millard (2001) translated the seven classic manufacturing wastes into what they meant relative to PD. Morgan (2002) found eleven classes of information waste. Several others did similarly, and Bauch (2004) collected and organized the various types of information wastes in PD. He created a framework that consisted of what can be wasted, main waste drivers, and sub-categories of these waste drivers. He then correlated these waste drivers to analyze how one type of waste breeds others, and used this analysis to create a strategy for sequential elimination of waste types.

2.4.4 Pull

The principle of pull is much harder to apply in PD than in a manufacturing environment, especially due to the fact that PD does not have physical items to be passed from one step to the next. Also, due to the lengthy and uncertain PD process, one cannot be assured that customers will want the output before the process is complete. Finally, the iterative nature of PD tasks makes implementing a pull system difficult (Slack, 1999). Despite these difficulties, Oppenheim (2004) has tried to establish a Lean Product Development

Flow System. This proposed system uses a value stream map to plan the work for an extended project, and then breaks down the tasks involved into subtasks that can be completed during short, constant-duration "takt" periods. An integrative event is held between each of these periods to collect the information created by individual engineers during the takt period and to adjust planning.

2.4.5 Perfection

Finally, the Lean principle of "pursue perfection" has not been explicitly dealt with in previous research. Slack (1999) appropriately pointed out that this principle can be generalized to any area. Continuous improvement and creating a learning environment have been cited consistently as ways to encourage the constant improvement required by this principle (Spear and Bowen 1999, Morgan 2002). Pursuit of perfection really represents a hunger that must exist in a Lean enterprise—the desire to do better by solving the problems that prevent improvement.

2.4.6 Lean PD Summary

Lean is an improvement philosophy that holds great potential to help the problems seen within the type of complex product development in the aerospace/defense industry. While several other PD process improvements such as set-based design and axiomatic design seek to turn PD on its ear, Lean PD seeks to make the established process more efficient by identifying areas needing improvement and streamlining with a focus on creating value. Research has shown that Lean can be applied to PD, but there is much more that needs to be done in order to enable the creation of truly Lean PD systems.

2.5 Earned Value Management System Data Primer

While Lean has proven to be very useful by focusing on value, the concept of value is nothing new. Leaders have long sought a way to make their products more valuable, and to make sure that their development practices are efficient and effective. Several frameworks exist for tracking a project's progress and for managing project. As is the case with most systems, the observation involved with the progress tracking affects the outcome of the process. Project management structures impact the overall value of a project and can drive behavior within a PD process. By studying how these systems are used, and how they purportedly measure value, one can learn more about what businesses officially consider valuable.

One such project management framework that is widely used in the aerospace/defense industry is the Earned Value Management System (EVMS). Government regulations stipulate that for defense contracts of a certain size EVMS must be used for planning and management. Essentially, EVMS is a system for measuring the value of process tasks and estimating and updating the budget and schedule for a project. However, the "value" it measures is based upon the budget assigned to a given task, not to how it affects the outcome of the process.

EVMS starts with a Work Breakdown Structure (WBS). It breaks a project into small tasks to be accomplished, and assigns a budget and schedule for their completion by certain groups of people. The progress on each task is estimated as a percentage each month when EVMS reports are filed. By multiplying the percentage of task progress by the budgeted "value" of the task (equal to the task's budget), the amount of the "earned value" for that task can be calculated. When a task is totally complete, all of its "value" should be reported as having been "earned" (and should be equal to the budget for the task). This "earned value" is accumulated over the course of the project and should add up to the total budget of the project if everything is completed.

EVMS reports are filed every month, and the completion percentage for each task is reported in these files. The task's completion ratio is multiplied by the budgeted cost of the task to arrive at the task's earned value. If Task X is worth $10, and it is estimated as being 80% complete at the end of the month, then $8 of value has been earned.

Simple, right?! Well, not so fast.

This completion percentage is often difficult for managers to estimate. Once a task is complete, it is obviously 100% complete; before completion, however, it is often difficult to estimate just *how* complete the task is. What is the difference between a task being 70% and 90% complete?

Once the earned value for each task has been determined for a given month, they are combined to arrive at the total project monthly EV. This number is referred to as the "Budgeted Cost of Work Performed" (BCWP). The BCWP is really the "earned value" for the project. By comparing the BCWP with two special numbers, the status of the project can be ascertained.

First, the amount of "earned value" for a given month is compared with the amount of work that was planned to be completed for that month, referred to as the "Budgeted Cost of Work Scheduled" (BCWS). The BCWS is determined by the WBS, and is the amount of task completion planned to be done in a given timeframe multiplied by the budget of each of the WBS tasks. Essentially, it is just how the budget is divided into the schedule. The sum of the BCWS for each month of a project equals the total budget of the project. If a project has every task completed on it, then at its completion, the BCWS and the BCWP should be equal, and should equal the baseline budget of the project.

The BCWP should, but does not always, agree with the "Actual Cost of Work Performed" (ACWP), which unsurprisingly represents the amount of money actually spent to do the work that was performed. ACWP can be tricky because oftentimes it includes work that is done for tasks that are not claimed for EV until the next month. BCWS, BCWP, and ACWP can include both labor and material costs, and are usually reported in dollars. For each month, they are usually reported for both the individual task and the project as a whole.

BCWP, BCWS, and ACWP can be compared in several meaningful ways. Differences and ratios of BCWP to both BCWS and ACWP are calculated. Schedule Variance and Cost Variance are very important to EVMS. Schedule Variance (SV) is the difference between BCWP and BCWS. If the BCWP is larger than the BCWS, then the SV is positive, and more work has been completed than was planned. BCWP < BCWS, SV is negative, and the project is behind schedule on that task. Cost Variance (CV) is the difference between BCWP and ACWP. If the ACWP <BCWP, then less money has been spent to achieve a certain amount of work and thus to claim a certain amount of earned value. If ACWP is greater than BCWP, then it has cost more to do the work than was planned, and thus the project is behind budget, as is usually the case in the aerospace/defense industry. Essentially, SV and CV are used to assess how far off budget and schedule a task is. While it makes sense that CV is reported it dollars, it should not be a big surprise that SV is too. The variance in the schedule reported by SV is not really a measure of how far behind completion any given task or set of tasks is, but it indicates how far behind schedule the project is with respect to earning value.

Similarly, the Schedule Performance Index and the Cost Performance Index are used to check the how well the project is performing to schedule and budget, respectively. Schedule Performance Index (SPI) is the ratio of BCWP to BCWS. SPI is favorable when its value is 1.0 or greater, which implies that BCWP>BCWS and thus that work is on or ahead of schedule. Similarly, CPI is the ratio of BCWP to ACWP, and CPI values of 1.0 or greater are preferred. A CPI less than one implies that BCWP<ACWP and that the project is over budget for the amount of earned value that has been completed (i.e it has cost more to complete the work than was planned).

Whereas SV and CV are absolute measures of how far the project is away from its planned schedule and budget, SPI and CPI are relative performance measures that serve project leaders and managers better during the course of the project. Generally, they try to keep SPI and CPI between 0.9 and 1.1. In the case of Company X, they try to keep it closer to 1, between 0.98 and 1.02.

In summary, EVMS is a system used to manage the progress of large projects. Its many acronyms can be very confusing, and Table 5.2 attempts to eliminate any remaining befuddlement by giving each of the EVMS acronyms, its full name, a description in layman's terms, and (if applicable) the equation that generates it. As currently practiced, EVMS is not fully consistent with Lean or its application to PD. My research seeks better approaches for adapting EVMS to enable Lean PD.

Table 2.2 EVMS Terms

Acronym	Full Name	My Layman's Terms	Formula
EVMS	Earned Value Management System	"a way to keep track of project progress"	N/A
WBS	Work Breakdown Structure	"a division of the project into smaller tasks that can be completed by the teams assigned to them"	N/A
BCWS	Budgeted Cost of Work Scheduled	"what should get done in the time scheduled"	N/A
BCWP	Budgeted Cost of Work Performed	"what actually got done"	N/A
ACWP	Actual Cost of Work Performed	"what it cost to actually perform the work"	N/A
SV	Schedule Variance	"how much was completed relative to what was expected"	SV = BCWP - BCWS
CV	Cost Variance	"how much it cost to do what was complete relative to what was expected"	CV = BCWP - ACWP
SPI	Schedule Performance Index	"normalized measure of how far away from the schedule the task is"	SPI = BCWP/BCWS
CPI	Cost Performance Index	"normalized measure of how far away from the budget the task is"	CPI = BCWP/ACWP

3 Research Strategy

The Lean Aerospace Initiative at the Massachusetts Institute of Technology was formed in 1993 to determine how to best apply the concepts of Lean to the aerospace industry. This consortium, which consists of MIT, the United States Air Force, and several aerospace companies, seeks to help aerospace enterprises create value to achieve lasting success in their environment of constant change. The first years were spent determining that the concepts of Lean really were applicable to this industry. LAI now focuses on the creation of *Lean enterprises.* In their award-winning book *Lean Enterprise Value* (Murman et al., 2002), LAI researchers explain what this means:

> *A lean enterprise is an integrated entity that efficiently creates value for its multiple stakeholders by employing lean principles and practices.*
> (Murman et al., 2002)

In order to learn more about creating such Lean enterprises, LAI is divided into three major research focus areas: Enterprise Architecture, Enterprise Change, and Product Lifecycle. A group of researchers within the Product Lifecycle group have focused upon product development. Robert Slack, Richard Millard, James Chase, Dr. Hugh McManus, and Dr. Eric Rebentisch preceded my involvement in the PD group. Their work has shown that lean applies to PD, that value definition is difficult, that value stream mapping is powerful, and have provided a guide for creating PDVSMs and for the transitioning of product development systems to a more Lean state. In addition to McManus's and Rebentisch's continued involvement, newer members of the group include Prof. Warren Seering, Christoph Bauch, Martin Graebsch, Jin Kato, and Josef Oehmen. Recent research has attacked the application of Lean to PD from several angles. Bauch (2004) focused upon understanding PD waste, while Graebsch (2004) looked at information flow in the PD process and the creation of a Lean PD display. Oehmen (2005) studied the integration of risk management into Lean PD, with an additional focus on high-performance PD teams.

In different ways, both Kato (2005) and I have focused upon the usefulness of PDVSMs in identifying value creation. Kato focused primarily upon making *very finely detailed*

PDVSMs that looked at how individual engineers spent their time on relatively short-term software projects in Japan. He was able to use the wastes from Bauch's waste driver framework to identify and measure the waste seen in these processes. He found that information inventory was prevalent in these processes, and that over-processing, rework, and defective information were the three most significant waste drivers. My work also used PDVSM, but I studied much longer development processes. While my maps are very detailed, they are not at the same level of resolution as Kato's.

My research strategy was to find several projects within an LAI member company, create value stream maps of their processes, and compare the maps to official value measurements. Company X offered several candidate projects within a single facility, and was able to help me greatly. Funding was provided for the time that engineers and managers spent talking to me, and I was granted access to "unclassified" information and areas of the facility. The projects that I chose ultimately revolved around the types of complex and technically demanding products that Company X has made for years, and were thus representative of the typical work at this facility. While both projects were in multifaceted contexts, one project focused upon software development and the other dealt with hardware. The projects were in different stages of development: one was in the requirements phase of a standard PD process, and one was in a technology development phase for a much larger spiral development process. The diverse nature of these projects helps make their results more easily representative of complex PD in general, but the facts that there were only two case studies and that they were both performed at the same company facility detract from my ability to generalize these results.

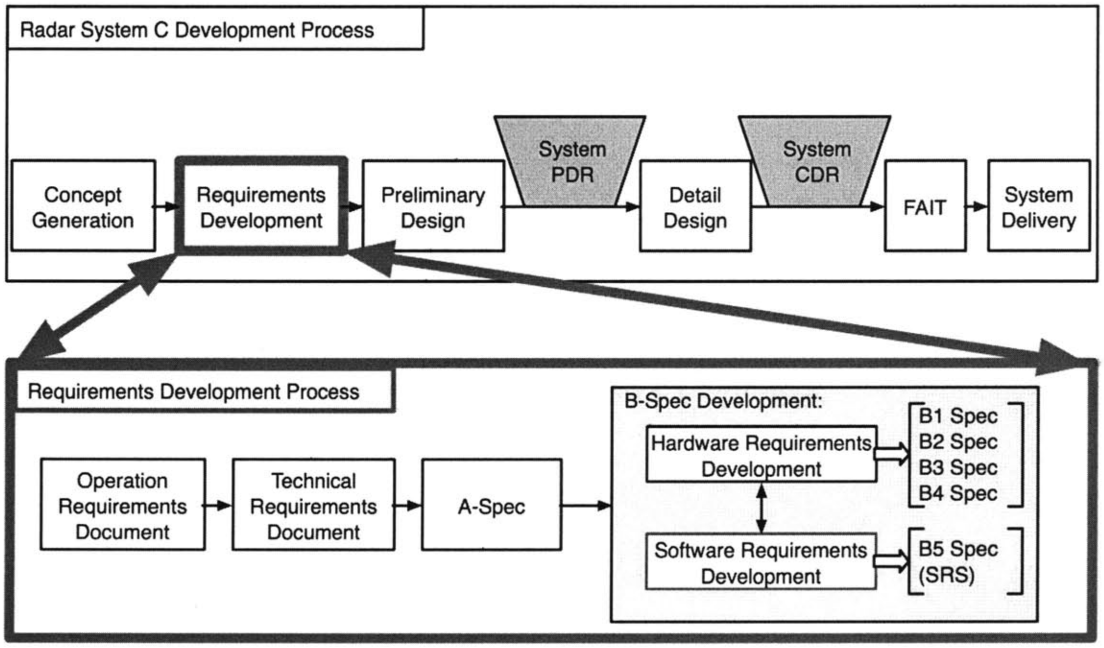
It is hoped that the combination of my work and Kato's (along with that of future students) will help clarify the proper level of detail and resolution needed in a map for identifying various type of value. Also, by studying the similarities and differences of our maps, one may be able to gather a better understanding of how to create the zoom-able VSMs proposed by Morgan (2002) and determine their usefulness. Ultimately, making such maps would be ideal for the Lean PD Display that Graebsch (2004) proposed.

In the chapters that follow, I explain my two case studies. I discuss the context of the development efforts, show the PDVSMs created, analyze the value of the tasks in the PDVSMs, and show the data for the official Company X measurement of value, the Earned Value Management System. I then compare my PDVSMs and value analyses to the EVMS data and point out discrepancies and alignments. In the final chapter, these results are summarized, and a hybrid PDVSM/EVMS system is proposed.

4 Project S – Software Requirements Development

In this project, engineers worked to create the requirements for a software subsystem to be used in the operation of a large radar system. Controlling the radar system and interpreting the information it gathers requires a great deal of information technology before, during, and after the system is actively in use. This specific software subsystem will be used before *and* after radar missions. The project's ultimate output was a large document (~200 pages) detailing the requirements needed to design the subsystem.

Figure 4.1 System C Development Process



4.1 Project S Context

The project involved a portion of the radar system's requirements development phase, as shown above in Figure 4.1. The System C requirements have recently been completed, and system preliminary design has begun. Next will come detail design, and then the system's physical components and subsystems will then be fabricated, assembled, integrated, and tested (FAIT). In parallel, each software subsystem will be coded and unit tested (tested in small sections), and then combined into a full software system and tested. The full software system will then be integrated with the hardware for overall system testing.

The requirements development process consists of four main tasks. First, an operational requirements document (ORD) is created. The System C government customer writes the ORD (~10 pages), which defines the main goals and minimum acceptable requirements for successful operation of the proposed system concept. Next, a Technical Requirements Document (~20 pages) was created from the ORD, and it defines performance requirements for the system and key sub-systems. Next, the "System Specifications" (Martin, 1997), also known as the "A Level Specification" or more commonly "A-spec" (~100 pages), is made by elaborating upon the TRD to create many high-level requirements for the various subsystems.

Portions of the A-spec requirements are then fleshed out to create the "B Level Specifications" for the various subsystems. Each of these B-specs serve as "Development Specifications" (Martin, 1997) for a specific subsystem. As is indicated in Figure 4.1, there are several kinds of B-specs. These cover system components (a B1-spec), critical items (B2), noncomplex items (B3), facility modifications (B4), and software (B5). Together, these B-specs are thousands of pages long and fully define the requirements for the system. Each of these B-specs is composed of hundreds of paragraphs detailing requirements in the form of "shalls".

A "shall" is a statement or paragraph that states a functionality that must be exhibited by the subsystem or part. In other words, a "shall" is a requirement of what the subsystem must do to be considered functionally operational. For example, were I to write requirements for a computer keyboard, the "shalls" might include: "The keyboard system *shall* allow entry of data into a computer system.", "The keyboard system *shall* be comprised of many keys.", and "Each key on the keyboard system *shall* enable entry of a different data symbol." (among many others). Since these "shalls" are such detailed and specific description of functionality, it should be no surprise that they are referred to collectively as "specifications". The system involved in Project S is much more complicated than a simple keyboard, and thus there are very many "shalls" needed to specify its functionality.

These software B-specs (B5) are known as "Software Requirements Specifications" (SRS). According to Martin (1997) the "SRS describes in detail the functional, interface, quality factors, special and qualification requirements necessary to design, develop, test, evaluate, and deliver" a software subsystem. The software for System C is comprised of eight subsystems, and each of these subsystems will have a corresponding SRS (B5 spec). Of these eight subsystems, the engineers studied worked on creating the SRS for Subsystem P. This SRS development effort is referred to as Project S.

The total budget for the requirements development and design of Subsystem P was 6% of the budget for the total software subsystem. The SRS development was budgeted to cost just 5% of the entire development effort for Subsystem P. Thus, this SRS effort was a small part of a single software subsystem development effort within a much larger system development effort. That is not to say that this was a miniscule project: its final budget was several hundred thousand dollars.

Each of these SRS documents is a few hundred pages long, and they are often referred to as "books" because of their size. As was the case for the other subsystems, the development of the Subsystem P SRS was managed by a "book boss". He was in charge of delivering the document.

4.2 Project Specifics

Subsystem P must allow a user to tell the radar system where and when to illuminate the sky, and it must be able to decode, interpret, and store the massive amounts of electronic data that are created by other software subsystems during a single radar usage. This specific software subsystem must also help with re-calibrating the overall radar system (to ensure system accuracy), as well as with limiting where the radars can shine (to minimize the likelihood of causing collateral damage).

4.2.1 Project S Marching Orders

The SRS "book boss" was given a set of four directions from the project leadership team on how to conduct Project S. First and foremost, the SRS must meet the requirements set out in the A-spec. Nothing should appear in the SRS that cannot be traced back to a requirement in the A-spec, and each relevant A-spec requirement must be addressed. Secondly, the group was directed to reuse as many requirements from another radar system as was possible. Third, they were instructed to simplify the document as much as possible. Finally, they were told that once all of the above conditions were met, they could then consider writing requirements for new customer demands or requests.

4.2.2 Requirements Reuse in Project S

Company X has a great deal of experience with creating radar systems. In order to leverage their existing knowledge, they chose to reuse requirements from previous radar systems in the creation of the System C requirements. Requirements reuse involves analyzing the requirements from one system and manipulating them so that they can be properly used in a new system. Sometimes, requirements are strictly copied verbatim, but other times, changes (small or large) may be required to make the requirements better suited for the new system.

The group was initially directed to reuse all of the requirements from a previous project, System H. However, since System C was thought to be more complex than System H, it was assumed that these reused requirements would comprise only about 80% of the new system's requirements. The other 20% of the System C requirements would be totally new requirements. This reuse plan did not last long.

A change occurred early in the project, as the government customer dictated that Project S should use a different system as a basis for requirements reuse. Systems C's design was to be based upon that of System T instead of System H. System T was a "Classified" project owned by a different company, and this led to a difficult process for the team to obtain the SRS for the subsystem of System T corresponding to Subsystem P of System C. Alphabet soup anyone?

As the SRS was developed, it was realized that reuse was not going to be as easy as had been thought, and the expectation was lowered to having about 60% of the SRS consist of reused requirements. This plan did not last either, as when it was released, only about 20-30% of the "shalls" in the Subsytem P SRS document were reused requirements. As will be shown with the Project S Value Stream Map later in Figure 4.3, the requirements reuse process was a highly iterative process. It started with the engineers learning about the specific sections of Subsystem P to which they were assigned and how they related to the other sections. The engineers then had to review System T's corresponding SRS in light of what they understood about their section. This allowed them to analyze which parts of the baseline document could be reused. The final step in this back-and-forth process was the actual crafting of the requirements. Sometimes this was as simple as copying a requirement from the System T SRS. However, most requirements needed changes to fit the new system. Small changes were usually handled by individual engineers, while larger changes generally required questions to be asked of other sections within Project S. Sometimes, particular issues arose which required communication between engineers on this project and those working on different subsystem SRSs.

4.2.3 Independence of Sections Within Project S

The software in Subsystem P consisted of several sections. Since there was so much reuse involved in this project, most of the team members were able to work on various sections of the SRS individually and independently. Sections were assigned so that a single team member would be responsible for one or more related sections. This nearly-independent work assignment allowed team members to work without needing frequent communication.

4.2.4 Project Re-Baselining

Another interesting aspect of this project is that its schedule and budget were "re-baselined" during the project. Re-baselining is a process in which the original budget and schedule, known as the "baseline", are altered considerably to establish a new standard against which all progress and status should be measured. In this case, the re-baselining was performed as the original contract budget was reallocated among the various System C subsystems, and it had a significant impact on the direction and detail of Project S. The Project S budget dropped 26%from Month 6 to Month 7 as a result of the re-baselining. While this in itself is not that strange, what is strange is the fact that this new budget was exceeded before the end of Month 7. Thus, the rest of the project was running over budget. This fact makes the re-baselining appear illogical

4.2.5 SRS Team

The team of engineers that worked on Project S was different than one might expect for such a project. Even though this was a requirements development project, there was only one systems engineer in the group. Normally, one might expect a group developing requirements to be mostly, if not completely, comprised of systems engineers. This lone systems engineer was the team leader and served as the "book boss" for the SRS, and was responsible for delivering the document. The team varied in size over the duration of the project, from only two, up to eight, and back down to one or two near the end. Besides the one systems engineer, all the other team members were software engineers.

One of these software engineers was special, in that he had formerly served as the "book boss" for the System T SRS that served as the basis for requirements reuse. In spite of the fact that Project S was "Classified" like the System T baseline SRS, the native version of the SRS could not be easily copied and brought to the facility where the team sat. While Company X had designed most of System T, the system was technically owned by Company Y. Due to government regulations regarding "Classified" documents, this senior engineer had to go to Company Y to view the SRS files. He then had to first convert the file format of the electronic version of the baseline document, so that there were no potential hidden characters in the file. Next, he read *each line* of the baseline SRS document *twice* to check that there was no "Secret" information before electronically bringing a portion back to the team. This process was intensely time consuming, and the engineer had many other commitments, so he had to complete this task over the course of several weeks, reading and transferring as much as he could about one day per week. Eventually, paper and electronic copies of the entire baseline SRS document were accessible to the team members.

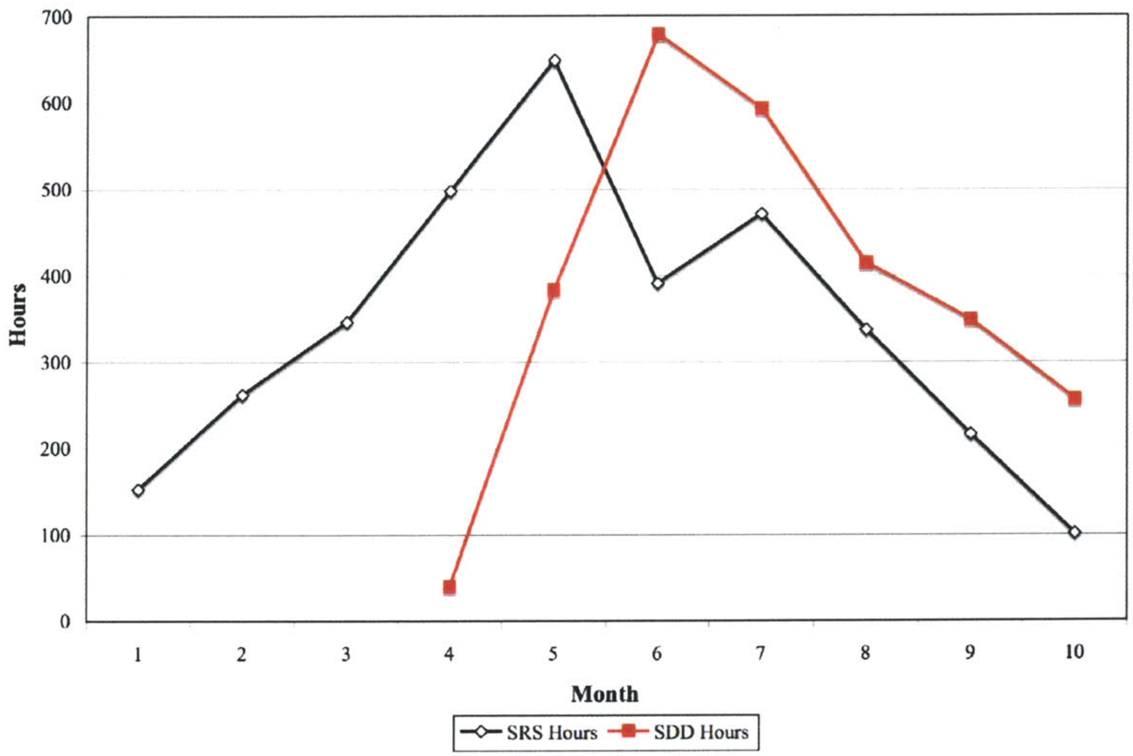
This senior engineer not only physically obtained this baseline document, but he also had keen insight into the process that went into creating those requirements. Because he had been the "book boss" for the baseline document, he was able to offer some degree of knowledge as to *why* some requirements had been written as they were. Additionally, he knew the people involved in that process, so if a particularly difficult question arose, he knew whom to contact to clarify any confusion.

While there was one highly experienced engineer on the team, there were also two engineers who had only recently graduated from college. One of these engineers worked on a section that was closely related to that of the senior engineer. These two had desks very close to one another, and the senior engineer offered much help to the junior engineer. The other junior engineer on the team was assigned a rather technically demanding section that required that he work mostly alone. However, he also had to work with the "book boss" (remember, the only Systems Engineer) on several issues relating his section to other sections within Subsystem P and to other SRSs. The rest of the team had various levels of experience.

While creating the software requirements, most of the team members were also actively working on the preliminary design for the Subsystem P Software Design Document (SDD). Some of the preliminary design was required to better understand the ramifications of the requirements as they were being developed. It also helped them realize when requirements were not sensible or feasible. They worked more fully on the preliminary design as the requirements development process was coming to a close.

Figure 4.2 below shows the number of hours the team worked each month on both the SRS and SDD. It shows that while the SDD effort began later, the peak in SDD effort came the month after that of the SRS effort. It is interesting to note that this month was the month the re-baselining occurred and also when the SRS effort dropped significantly.

Figure 4.2 Requirements Development (SRS) and Preliminary Design (SDD) Hours Worked



4.2.6 Team Mechanisms

The team used several mechanisms to facilitate the SRS development process. Weekly meetings were held in which progress and issues encountered were discussed. In each meeting, individuals shared his or her progress, and reported any issues that required assistance. Starting around Month 12, the weekly group meetings for Subsystem P were combined with those of another subsystem. These two groups were related, and it was determined that their individual design efforts would be helped through improved communication between the two. Additionally, the other subsystem's development team had been reduced to a single person, and project management wanted to ensure that this individual was continuing to make progress in the proper direction. Even in these combined meetings, the Subsystem P effort was discussed heavily.

Also helping the team's communication was the fact that most of the team members' desks were located near one another in a room especially designated for use by the System C development team (not just for the Subsystem P team). This room was open only to those who had the proper Security Clearance, and each desk had two computers: one on a special "Classified" network, and the other on a normal network connection. "Classified" information is not allowed to be stored on machines that were connected to the standard network. Having both machines available for use allowed the team members to easily share "Classified" information with one another while also being able to access outside "Unclassified" resources.

The team members used a special "Unclassified" software system, which enabled them to electronically post requirements for others to access, review, and create editing suggestions. Once their section's requirements were rather stable, each engineer entered them into a "Classified" software system called DOORS by Telelogic. This system let them check that all the requirements could be traced back to the A-spec, and kept the requirements in the form required by the Department of Defense.

The SRS team got a significant amount of feedback from the customer. Government representatives attended reviews and some weekly meetings. Out of the 40 weekly team meetings held between Month 2 and Month 14, government representatives were present at 5. Additionally, they attended the team's "internal" company review (Month 7), Preliminary Design Review (Month 10), and Interim Review (Month 11), providing feedback on several issues each time. In accordance with the government contract, the team was required to submit a draft version of their SRS to the government 45 days prior to the PDR. The government representatives reviewed the SRS as it was, and offered 73 official comments to the group. These comments were divided into three categories: A, B, and C. Category A comments were those that were critical and *had* to be addressed before the PDR. Categories B and C were comments that needed to be addressed but were not as important as those in Category A. Some of these comments requested additional capability or clarification of the requirements text, while others pointed out specific errors. The group resolved all Category A comments and several Category B and C comments before the PDR by either changing the SRS or denying the comments. The government representatives also offered 41 detailed comments to the team at the Interim Review, which were resolved by the SRS release in Month 14.

4.3 Value Stream Map

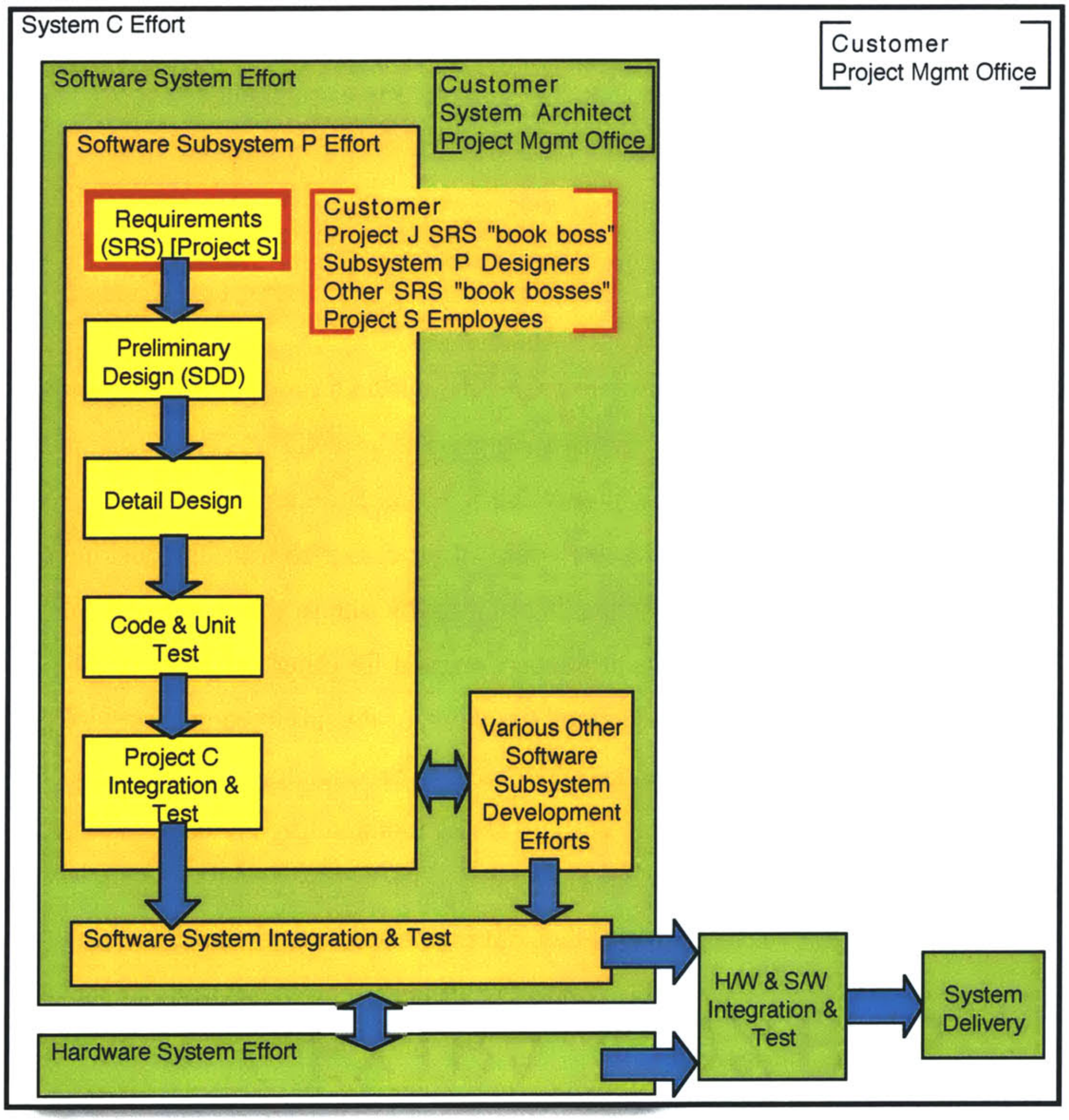
A Product Development Value Stream Map was created to examine the SRS creation process. The following discusses the Project S process and its overall value. Next, the PDVSM is presented and discussed, and each of the tasks in the PDVSM is analyzed for how it adds value to the process or SRS product. Creating this map enabled understanding of the relationship between the tasks the flow of information in this process.

4.3.1 Value Proposition

One should not attempt to create a map the flow of value within a process using a PDVSM without first understanding the value of the process as a whole. The value of Project S is not simply definable by one sentence; various stakeholders value it differently. The goal of the Project S SRS development process is to translate customer needs into specific, well-written requirements that will enable the further development of Subsystem P by using some requirements previously created for other programs and by creating other new requirements. To understand the value of this effort, one must look at how it adds value to each of the sets of stakeholders in this product development effort. One cannot decipher the stakeholder value without also understanding the context within which this effort is taking place. Figure 4.3 below is similar to Figure 4.1 in that it illustrates the context within which Project S lies, but it offers a better insight into how the various stakeholders perceive the value created by Project S.

The highest level of Project S is the System C Effort, which includes both hardware and software elements. The stakeholders most interested in the outcome of the System C Effort are the Customer and the Program Management Office. The Customer is most interested in getting a useful, functional, and reliable system at an acceptable price. The Program Management Office is most interested in making sure that this system is delivered to maximize the customer value. Delivering a successful System C will prove Company X's capabilities and could affect future contracts (which mean more money for the company), thereby offering some value to all Company X employees and stockholders.

Figure 4.3 Project S Context and Stakeholders



The Customer, System Architect, and Program Management Office are interested in the Software System Development, since it offers them value. The System Architect is responsible for ensuring that System C will operate well and thus provide value to the Program Management Office, and Customer. Since the software plays a huge part in the operation of System C, a successful Software System Development will offer value to each of these three.

When looking specifically at Project S, there are several sets of stakeholders. First, the various SRS "book bosses" are stakeholders in the outcome of the Project S SRS outcome. Within the Software System Effort, there are eight different software subsystems, and some interactions between these exist. Thus the "book bosses" are interested in how the requirements from each subsystem interact and affect their respective sections. Of course, the Project S SRS "book boss" is very interested in the outcome of this SRS effort, for he is ultimately responsible for it. Secondly, the Project S Employees extract value from the SRS development in the form of job satisfaction and lessons learned. They also get compensated for the execution of the work and are therefore very interested in the effort. Specifically, those that must perform the software design of Subsystem P are very interested in the outcome of the SRS effort, because the quality of the SRS will dictate how easy the design effort will be. Software Subsystem P is crucial for the operation of System C. The Customer holds a vested interest in the Project S SRS development because they want to make sure that all of their needs are well translated into actionable requirements so that this subsystem can be effectively designed.

To summarize, the value of Project S is relative to the level within which it is viewed. It is not as simple as saying "X is the value of Project S and that is it." While there is no single definition of the value of the process, most stakeholders were interested in the value from translating the customer needs into specific, well-written requirements that will enable the further development of the Subsystem P.

4.3.2 SRS Development Value Stream Map

The PDVSM created for the SRS development effort is shown on the following pages as Figure 4.4. The map consists of boxes, circles, and arrows between them. The boxes and circles represent tasks and product reviews. The arrows represent the flow of information between tasks. The SRS process flows from left to right, and the map is divided by month.

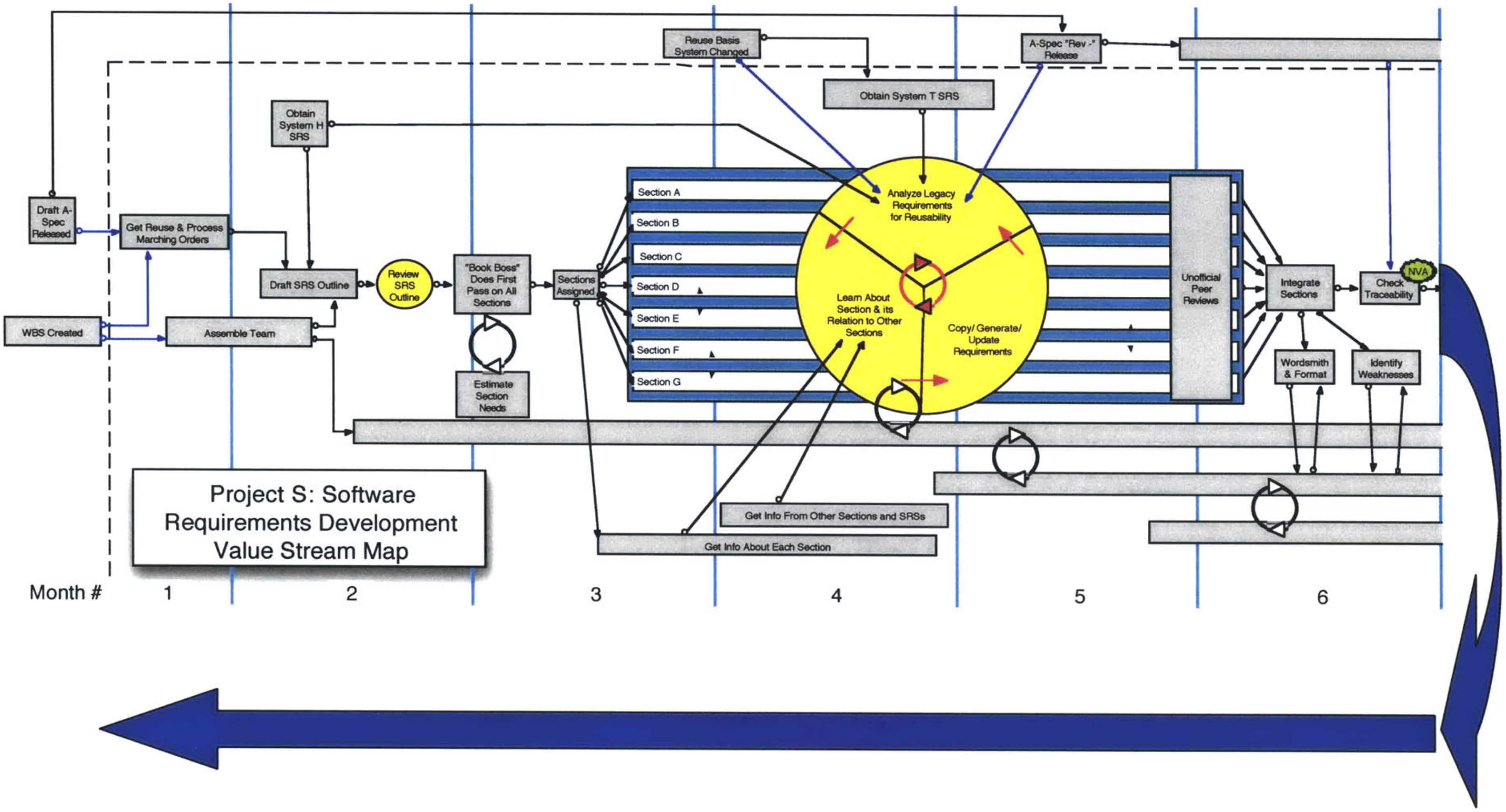
The process began with the creation of the Draft A-Spec, which led the team to creating an outline for the SRS. This outline was reviewed and then broken into sections and assigned to the team's engineers. The team then went through the main value-adding step of creating the first set of new and reused requirements (more on this later). Each section was then reviewed by another team member, and they were then all integrated. The requirements were entered into DOORS, and a supposedly "internal" peer review of the SRS (which was attended by the "book bosses" of the other SRSs and customer representatives) was held. After action items from this review were addressed, a draft of the Project S SRS was submitted to the customer for review approximately 45 days prior to the Preliminary Design Review. This submission enabled the customer to generate comments regarding the draft. These comments were addressed and presented at the PDR. More updates and two more reviews were held before SRS was first officially released in Month 14. The SRS was considered rather stable and was released at a level of maturity known as "Revision —" (or "Rev —") just before the completion of this thesis. Any subsequent changes to the document must pass through a review board. The next versions of the SRS will be known as "Revision draft", "Revision Final". Thus, "Rev —" is just the first fully complete draft of the SRS and is considered the first product output from Project S.

Other than using boxes and circles with arrows, the map does not follow the format given in the PDVSM Manual. Cycle Times and In-Process Times for each task were not taken. Also, the type of value added has not been indicated on the map: this is saved for the value analysis. Additionally, some frequently-occurring and ongoing tasks are represented by long bars on the map, which one would not find in the PDVSM format.

Moreover, circular arrows on this map indicate the reciprocal flow of information between tasks. In some cases, the circular arrows cannot be drawn, and a pair of straight arrows serves to indicate this type of flow. Also, starburst shapes indicate non-value added tasks. A dotted line runs along the left and top of the map to act as the Project S boundary. All tasks above and to the left of the line were performed by people outside of Project S. These tasks also impacted other subsystem developments, and information flows across this border are indicated by blue arrows.

Finally, the main value-adding part of this process is illustrated in a format that is not at all recommended by the PDVSM Manual. Shown spanning Months 3-6, this set of steps includes the analysis of the legacy requirements and creation of the SRS requirements. This sub-process is represented by a large circle divided into wedges according to the tasks: "Learn About Section & its Relation to Other Sections", "Analyze Legacy Requirements for Reusability", and "Copy/Generate/Update Requirements". Behind this circle run bars that represent the various independent sections of the SRS development, for each had to go through this iterative process. These tasks are illustrated within a circle because there was no distinct linear order in which these steps occurred. In the initial iterations of the map these steps were represented in a linear progression, but the "book boss" declared that to be non-representative of reality. In other words, the way that this information flowed among the tasks necessitated the shape shown in Figure 4.4. As they learned about their respective sections, the engineers were better able to understand what parts of the legacy requirements were suitable for reuse. As they figured out what was reusable, they either copied the legacy requirement, edited said requirement to fit the new system, or created new requirements that might serve a similar purpose. In order to create new requirements, the engineers had to learn more about their system. In copying and updating legacy requirements, the engineers gained a better understanding of the legacy document, and therefore understood its reusability more. Thus, information flowed in both directions among all three of these separate tasks. The result of these three tasks is the primary value of Project S: initial requirements reuse and requirement development.

Figure 4.4 Project S Value Stream Map



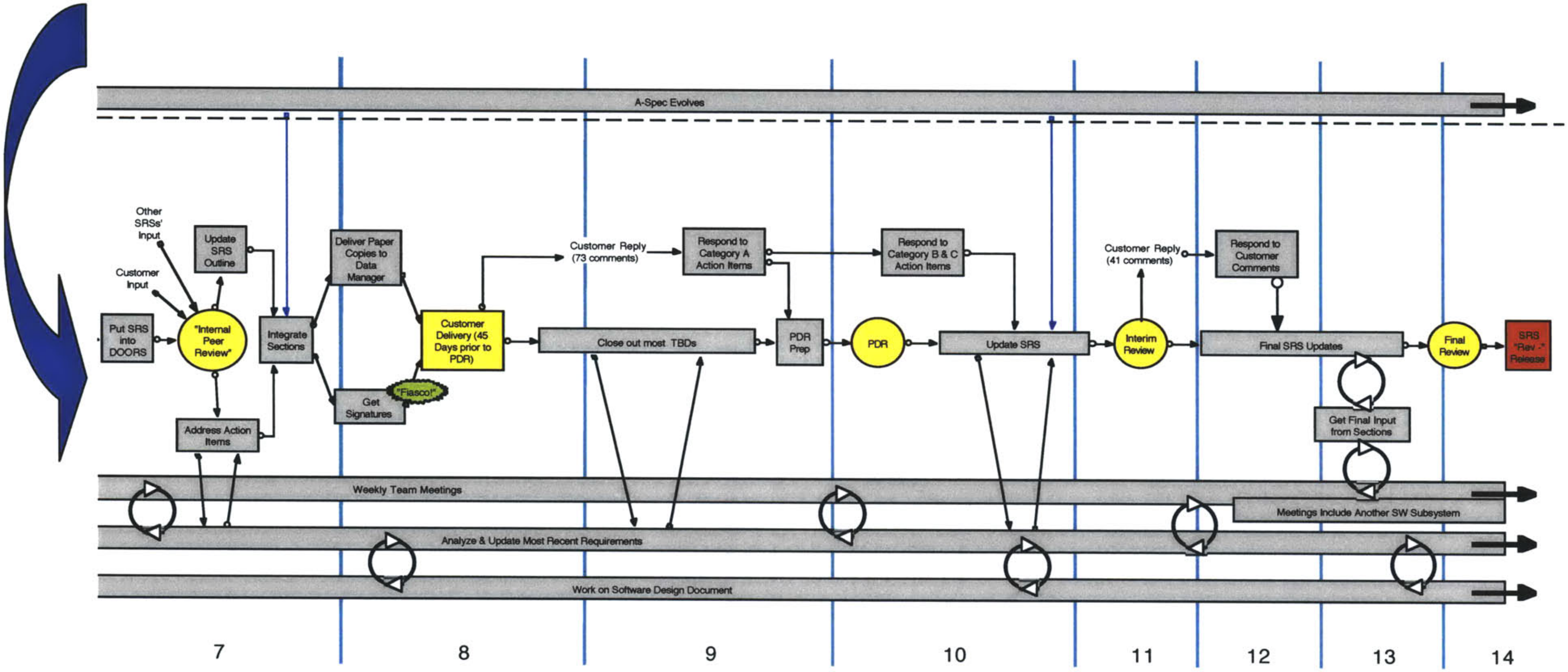


Figure 4.4 Project S Value Stream Map (continued)

4.3.3 Mapping Effort

The PDVSM effort was carried out in a series of steps. First, after several interviews with the "book boss", I learned about the project as a whole, the Project S context, who the members of the team were, what sections each of them worked with, and what types of resources the team had at its disposal. Unstructured interviews were then conducted with each member of the team to learn how they viewed their own section, how they viewed its relation to the other sections, and how they interacted with other members of the team. I also attended several of the team's weekly meetings to gain a better understanding of the issues they faced and how the team interacted.

Using the information gathered in the interviews and meetings, a rudimentary hand-drawn PDVSM was created. The author created six iterations of the map before presenting it to the "book boss" for critiquing. After a couple of back-and-forth meetings, it was realized that the representation of the reuse cycle (which at the time was illustrated as a strictly linear process) had to be different. A final hand-drawn map was completed, and then an electronic copy was made using OmniGraffle Professional software. A printout of the electronic PDVSM was reviewed by the "book boss", and approved. Slowly, small changes have been continually recommended, resulting in several versions of the electronic map being created.

4.3.4 Value Analysis

Using a value analysis framework developed by Josef Oehmen (2005), which is based on Robert Slack's decomposition of customer value, I analyzed the value of each of the tasks in the process. Oehmen's framework breaks down how a task adds value: by contributing either to the time, cost, or quality of the product and/or the development process. Table 4.1 shows this value analysis for Project S.

Table 4.1 Project S PDVSM Value Analysis

Task	Process/ Product	Time	Cost	Quality
Get Reuse & Process Marching Orders	Process	Provides schedule for project	Reuse dictates # hours to be worked on shalls	
Assemble Team	Process		Determines # & pay grade of people => cost of SRS	
Obtain System H SRS	Process	Enables reuse to begin		
Draft SRS Outline	Product & Process	Breakdown enables one to see when effort needed		Gives a structure for building the SRS
Review SRS Outline	Product	Ensures SRS is doable in time allowed	Ensures SRS is doable in budget allowed	Ensures quality of Outline
"Book Boss" Does First Pass on All Sections	Process	Gives better understanding of the work to be done; helps divide work to get job done on time		Discovers potential stumbling blocks early
Estimate Section Needs	Process	Divides work to ensure on-time delivery		
Sections Assigned	Process	Divides work to ensure on-time delivery		
Reuse Basis Changed	Process		[Had major negative impact—non-value added]	Requires usage of reuse basis more closely aligned with System C
Get Info About Each Section	Process			Adds to engineer's knowledge for reuse assessment
Get Info from Other Sections and SRSs	Process			Adds to engineer's knowledge for reuse assessment
Obtain System T SRS	Process	Enables reuse to begin		
Learn About Section and Its Relation to Other Sections	Process			Engineer knows what section needs are, better able to write req'ts
Analyze Legacy Requirements for Reusability	Process	If reusable, saves time	If reusable, saves money	Determines what doesn't need to be written newly; offers a set of already complete req'ts
Copy/Generate/ Update Requirements	Product			Actually creates the product!!!!!

Task	Process/ Product	Time	Cost	Quality
Unofficial Peer Review	Product & Process	Determines what needs to be worked on in the future		Checks product quality
Weekly Team Meetings	Product	Fine schedule adjustments as necessary		Discuss progress; identify & resolve issues; generate cohesiveness among independently-working team
Analyze & Update Most Recent Requirements	Product			Continually improves the SRS product quality; allows identification of other improvements
Work on Software Design Document	Product & Process			Enables team to understand ramifications of requirements decisions; encourages & enables team to make better product
Integrate Sections	Product & Process			Collects everything together; makes product into single item for first time; allows all to be seen together
Wordsmith & Format	Product			Checks minor quality issues; begins putting product in final format
Identify Weaknesses	Product			Improves quality
Check Traceability	Process			NOTHING—Putting requirements into DOORS will do this!
Put SRS into DOORS	Product & Process			Put product into final format; important for execution of rest of Project; important to customer; ensures SRS will not deviate from A-Spec; ensures quality
"Internal Peer Review"	Product	Assesses schedule to complete Project. Can redirect if progress known	Assesses cost to finish project	Checkss quality; identifies action items to improve
Update SRS Outline	Product & Process			Redirects process. Improves SRS product.
Address Action Items	Product			Improve quality
Integrate Sections	Product & Process			Collects changes; creates updated unified document

Task	Process/ Product	Time	Cost	Quality
Deliver Paper Copies to Data Manager	Process	Contractual requirement to deliver to customer		
Get Signatures	Product			Checks quality (forced management review)
Customer Delivery (45 Days Prior to PDR)	Product			Required task; allows for feedback from customer
Close out most TBDs (issues To Be Decided)	Product			Improve quality
Customer Reply (73 comments)	Product			Checks quality, Figure out how to improve
Respond to Category A Comments	Product			Resolves major quality issues
PDR Prep	Process			Makes PDR smoother
Preliminary Design Review	Product & Process	Checks status; determines how much time needed to complete	Checks status; determines how much $$$ needed to complete	Checks quality; get more items for improvement
Respond to Category B & C Comments	Product			Resolves minor quality issues
Update SRS	Product			Improves quality
Interim Review	Product & Process	Checks status; determines how much time needed to complete	Checks status; determines how much $$$ needed to complete	Check quality; get more items for improvement
Customer Reply (41 comments)	Product			Check quality, Figure out how to improve
Respond to Customer Comments	Product			Improves quality
Get Final Input From Sections	Product			Improves quality
Final SRS Updates	Product			Improves quality
Final Review	Product	Checks schedule status, especially relative to prelim & detail design.	Checks budget status, especially relative to prelim & detail design.	Final quality check
"Rev —" Release	Product			Product Delivered; all subsequent changes must pass through configuration control: assures quality

4.4 Project S Earned Value Management System Data

The following section explains the Earned Value Management System (EVMS) data that Company X used to track the progress of Project S. It shows how they think value was earned during the project. The EVMS data is based upon the Work Breakdown Structure (WBS) created by the System C leadership team and passed down to the Project S "book boss". This WBS broke Project S down into nine tasks. Each month, the completion percentage for each task was estimated and reported as the earned value (EV; remember, out of 1.0) claimed. The EV was then accumulated across the tasks to create one set of EVMS data for each month. This data was passed on to the System C leadership team so they could ascertain the overall system development status. The Project S EVMS cost account manager was responsible for handling the EVMS data. She was very helpful in providing access to very many EVMS data files and explaining them. Digesting and analyzing the data was a *formidable* task, requiring an iterative trial-and-error approach. This data is presented below after a brief explanation of the task break down. Table 4.2 below provides a refresher of the terms used in EVMS.

Table 4.2 EVMS Refresher

Acronym	Full Name	My Layman's Terms	Formula
EVMS	Earned Value Management System	"a way to keep track of project progress"	N/A
WBS	Work Breakdown Structure	"a division of the project into smaller tasks that can be completed by the teams assigned to them"	N/A
BCWS	Budgeted Cost of Work Scheduled	"what should get done in the time scheduled"	N/A
BCWP	Budgeted Cost of Work Performed	"what actually got done"	N/A
ACWP	Actual Cost of Work Performed	"what it cost to actually perform the work"	N/A
SV	Schedule Variance	"how much was completed relative to what was expected"	SV = BCWP – BCWS
CV	Cost Variance	"how much it cost to do what was complete relative to what was expected"	CV = BCWP – ACWP
SPI	Schedule Performance Index	"normalized measure of how far away from the schedule the task is"	SPI = BCWP/BCWS
CPI	Cost Performance Index	"normalized measure of how far away from the budget the task is"	CPI = BCWP/ACWP

4.4.1 Project S Earned Value Management System Data

To understand the EVMS data, one must understand how the project's earned value was counted. The WBS for Project S divided the SRS development into nine distinct tasks:

1) Analysis of Legacy Requirements
2) Update document outline to include new/added functionality
3) Increment 1 Shall development
4) Increment 2 Shall development
5) Increment 3 Shall development
6) Document Review and Release
7) Increment 4 Shall development
8) Increment 5 Shall development
9) Document Review and Release

Task 1 describes the reviewing of System H's SRS document to create an outline for the Project S SRS, as well as the analyzing of the System H requirements to determine which apply to Project S and then adding them to the new SRS. Obviously, this had to be repeated for System T once the reuse basis was changed. Task 2 is the updating of the outline created in Task 1 to reflect those aspects that make System C differ from System H. Tasks 3,4,5,7, and 8 represented the creation of new "shalls" for the SRS book. Task 6 represented the review and release of data to the government customer 45 days prior to PDR. Task 9 represents the release of the SRS to "Rev —".

The way that "earned value" was assigned to these tasks was interesting. Obviously, for Tasks 6 and 9 to be considered complete, a review must be held and the corresponding document must be released. Similarly, Task 2 was considered complete when a review was held for the SRS Outline. It was assumed that at the end of Task 1, all the reused requirements would be added to the SRS. Tasks 3,4,5,7, and 8 were each to be considered complete when a certain number of new "shalls" had been written and added to the SRS book. The number of new "shalls" for each task is represented below in Figure 4.5 as a fraction of the total expected number of SRS "shalls".

Figure 4.5 Division of New "Shalls" Among Tasks

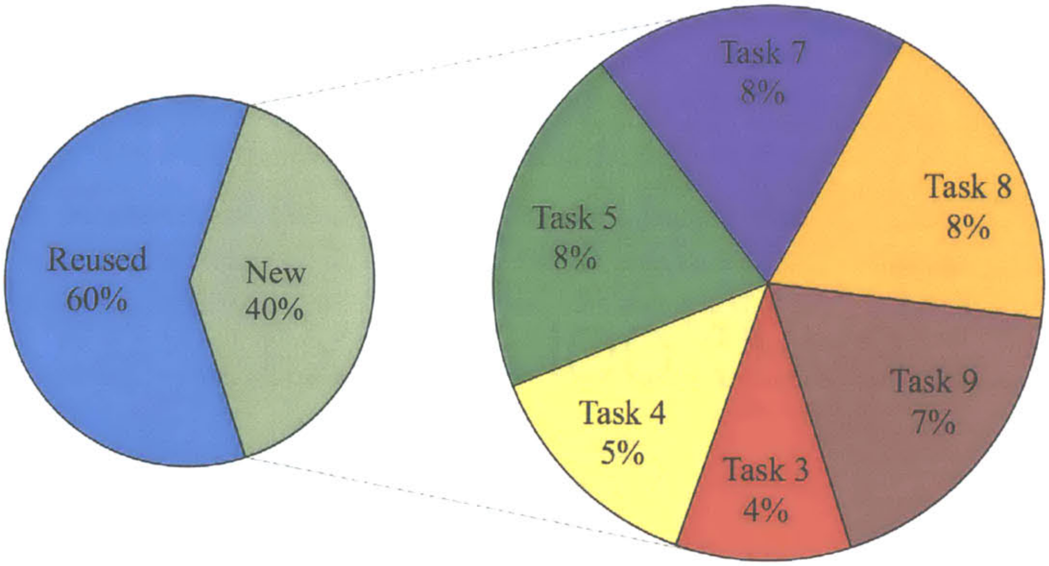


Figure 4.6 Project S Monthly Task Earned Value

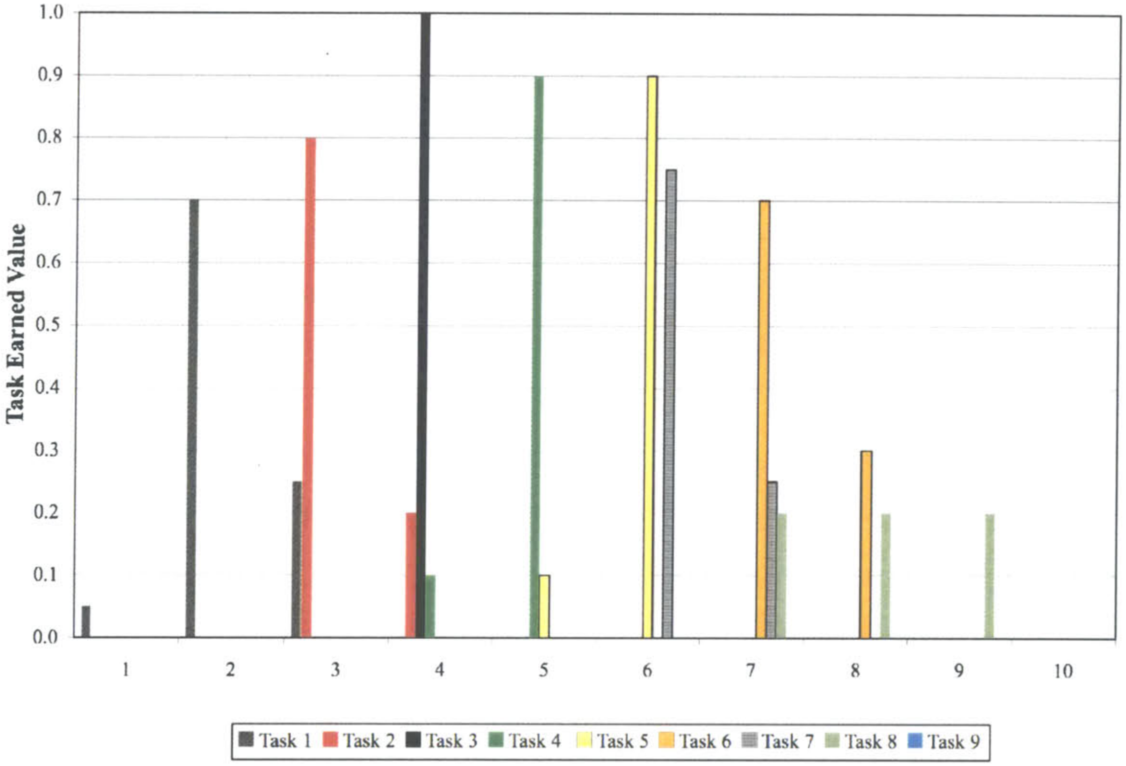


Figure 4.6 above shows the earned value claimed each month for each task. For instance, the first 5% of Task 1 was completed in Month 1, 70% of it was done in Month 2, and the remaining 25% was finished in Month 3. These tasks were mostly reported as complete in order, and each month (except Month 6) one task dominated the total monthly EV claimed. Figure 4.7 shows the same data, but with the EV accumulated such that one can more easily see how the total project EV amassed. It makes it perfectly clear that in Month 8 much more value had been earned than in Month 4 (as is expected).

Figure 4.7 Project S Cumulative Task Earned Value

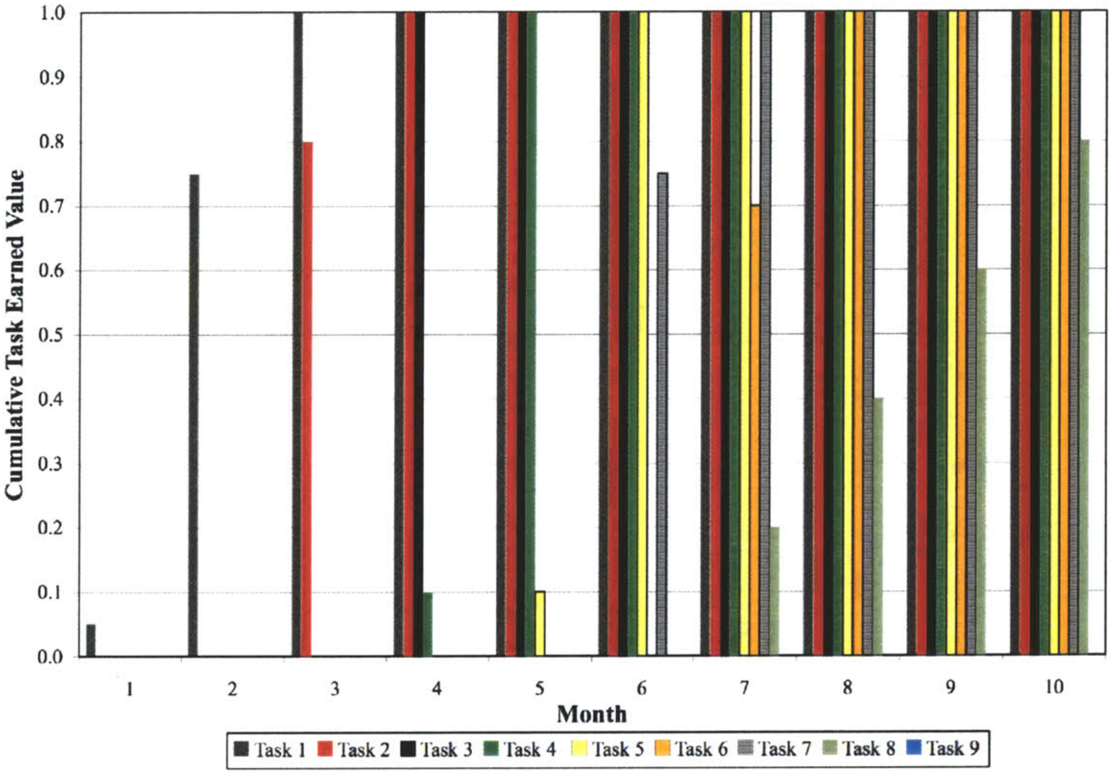


Figure 4.8 below shows the Project S BCWS, BCWP, and ACWP for each month. The numbers are graphed as a percentage of the original Project S budget in order to mask the real numbers. The schedule and cost information for all of the tasks was combined to get this EVMS data. Several facts can be taken from this graph that illustrates the overall health of the project. As of Month 10, the team had not achieved what they were scheduled to do. In fact, it was not until Month 7 that they even came within 10% of catching up to schedule. In Month 5, the team was 41% behind schedule. The Month 6

re-baselining is evident, as it caused the cumulative BCWS to actually drop as the monthly BCWS went from 22% to 3%. Normally this could not happen, as you would never expect less to be done after Month 6 than after Month 5. Regardless of who *your* coworkers are, it's hard for most engineering teams to do negative work. Also readily evident from this figure is the gross overrun of cumulative ACWP relative to BCWP. The team was over budget from the beginning of the project, and the cost performance only improved during the month of the re-baselining. This poor budget performance suggests either a large misjudgment in estimating the amount of time to perform the budgeted work, or that the team performed poorly. Another interesting point is that with the exception of Month 6, it cost the team more to do the work than was expected by a considerable margin. That means that every month, the team got more over budget. In Month 6 however, the team managed to do much more than planned (as they caught up to schedule) but spent less money in doing so than was expected.

Figure 4.8 Project S Cost and Schedule Data

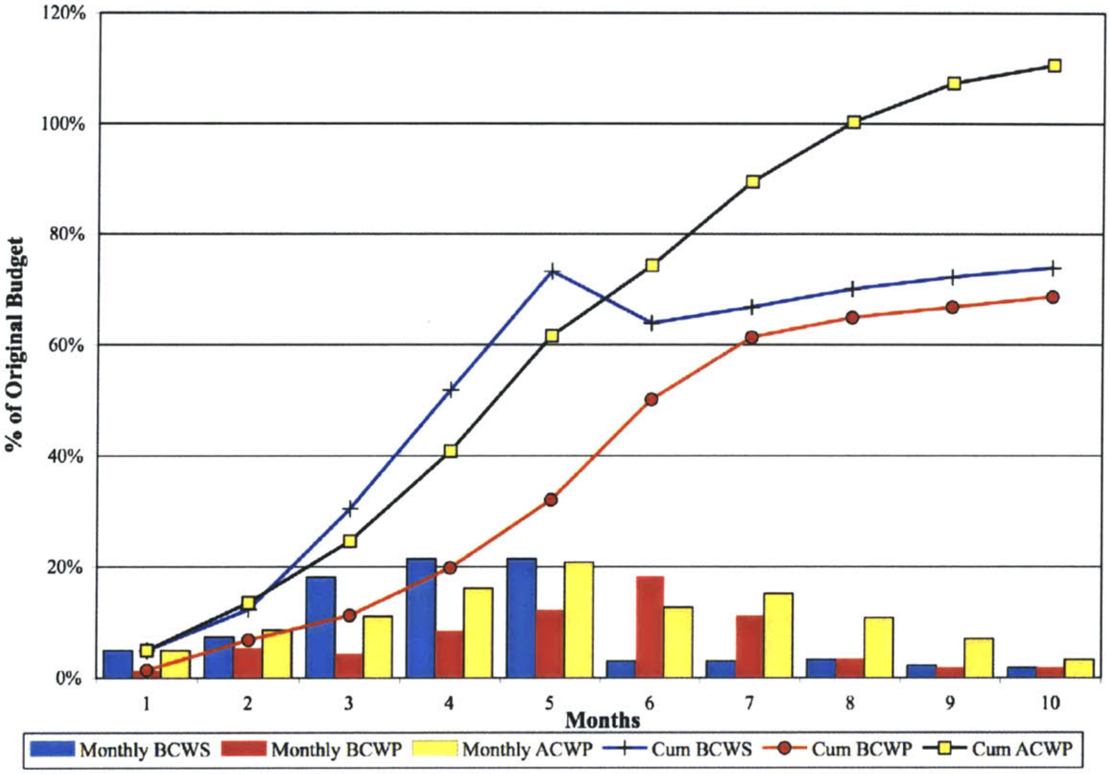


Figure 4.9 Project S Performance Indices

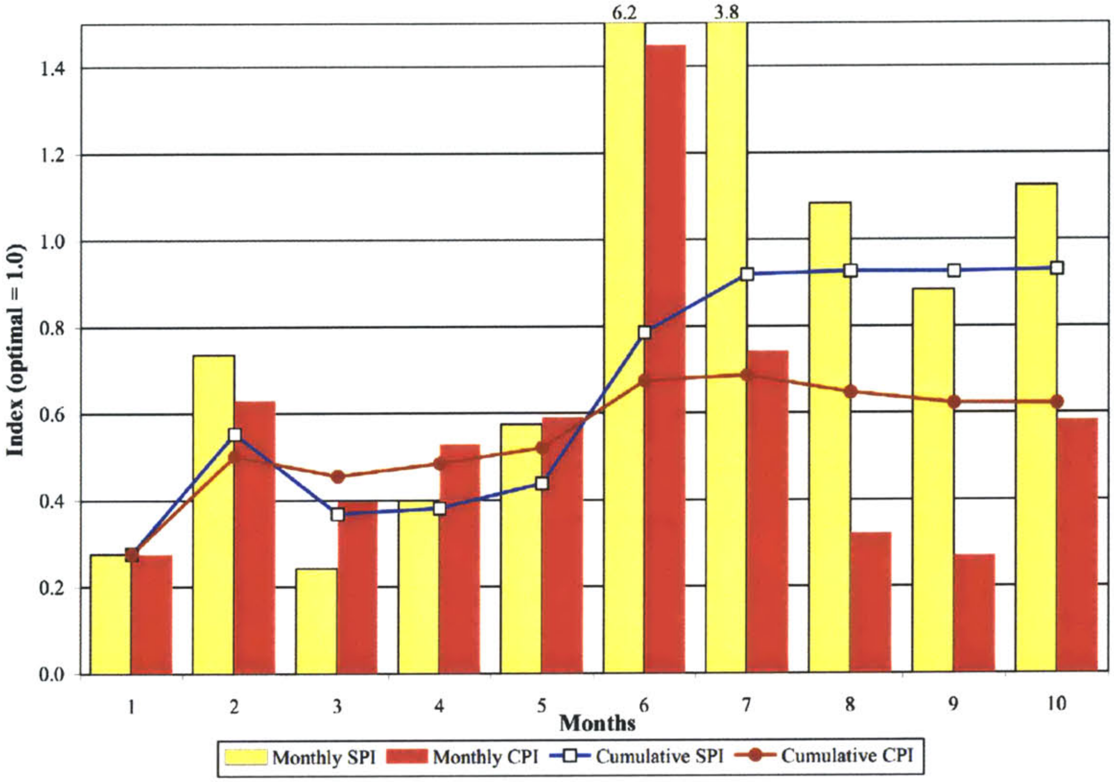


Figure 4.10 shows the project's cost and schedule performance indices (CPI and SPI), which represent the relation between BCWP & ACWP and BCWP & BCWS, respectively. Other than in Months 10 and 11, the project was behind schedule and over budget. In fact, only in one month did the CPI exceed 1.0, but at no point was it within 20% of the nominal value. Excluding Months 10 and 11, the project was always nearly 40% away from the optimal CPI value! While the CPI was nearly always far below optimal, after the re-baselining, the monthly SPI shot up and then returned to near the monthly optimal level. The incredibly high SPI in Months 6 and 7 shot the cumulative SPI from 0.44 to 0.92, and after this the project stayed within 10% of the optimal cumulative SPI. By Month 10, the team had nearly caught up to the prescribed cumulative and monthly schedules, but never got where it needed to be.

4.5 Comparison of EVMS and PDVSM

Two definitions of the value of Project S have been presented. The EVMS information shows how Company X's System C Leadership Team thinks Project S adds value, and the value stream mapping and analysis presents an outsider's view of the process and its value. It is interesting to compare the two to see how they align.

Shown below are two figures that show a lot of information. Figures 4.12 and 4.13 show monthly and cumulative EVMS data (respectively) laid over a small version of the value stream map. This EVMS data includes the EV data claimed for each of the tasks, as well as the project's overall BCWP and ACWP for each month. These graphs show how the actual process represented by the value stream map does not align with the EVMS data.

The EVMS data does not correspond perfectly with the VSM drafted. Too much earned value has been claimed early in the project, as draft versions of "shalls" had been added to the SRS but counted as if they were final versions. By Month 9, nearly all of the value of the project has been claimed, but there was still much work left to do. Because there was more work to do than there was value to officially earn, the team stopped using EVMS for the remainder of the project. As a result of their overzealous EV claims early in the project, the team needed to "unearn" value to properly continue using EVMS. This fact highlights a flaw in the EVMS system—one cannot claim negative value to correct previous errors.

Several other incongruities exist between the two value methods. First, while the EVMS data seems to indicate that Task 1, the "Analysis of Legacy Requirements" was complete by Month 3, the truth is that the team was still looking back at the baseline documents well into Months 9 and 10. Reviewing the legacy requirements and reusing "shalls" was a much more difficult task than initially thought, but the EV was claimed early. Also, while the PDVSM indicates that Task 2 should have been complete in Month 2 instead of Month 3, the truth is that there was some lingering aspects of the SRS outline creation that are not reflected on the map. Furthermore, Month 10 appears to be a very productive month, with 90% of Task 5 and 75 % of Task 7 earned value being claimed. However, less ACWP was charged that month than had been since Month 3 or would be until Month 8. This could be an artifact of the team claiming work in Month 6 that had been *nearly* finished in Month 5 but not complete or due to the re-baselining. Alternatively, the Section Integration activity in Month 10 collected many "shalls" that had been previously created. Once assembled, they were officially part of the SRS and more easily counted and aligned. Thus, it was easier to claim the requirements as being complete.

Some aspects of the EVMS data do make sense with respect to the PDVSM. Task 6, the review and release of the draft SRS was mostly claimed in Month 7 (when then review was held), with less being claimed in Month 8 (when the release was). Additionally, it makes sense that tasks 2-5 should be claimed mostly during the big requirements reuse cycle (even though these tasks were supposed to be analyzing the reused requirements and not creating new ones).

Overall, there is some overlap between how EVMS and PDVSM measure the value of Project S, but they are very difficult to compare outright. The way Project S has been broken down into tasks in this case has very little to do with how the project was executed. While the division of tasks created for estimating project progress seems to be rather straightforward, it does not adequately represent how each task was performed. It ultimately led to the EVMS data being less than perfectly useful for understanding the value of the project. According to the Project S "book boss", the Earned Value Management System was used rather loosely. Essentially at the end of each month, his boss asked him "How much progress have you made and how many "shalls" are in the SRS?". The real progress of the task was not represented by the EVMS data, and this is one of the main reasons for the lack on congruity between the two value perspectives.

Figure 4.12 Project S Monthly EV, BCWP, and ACWP Superimposed on PDVSM

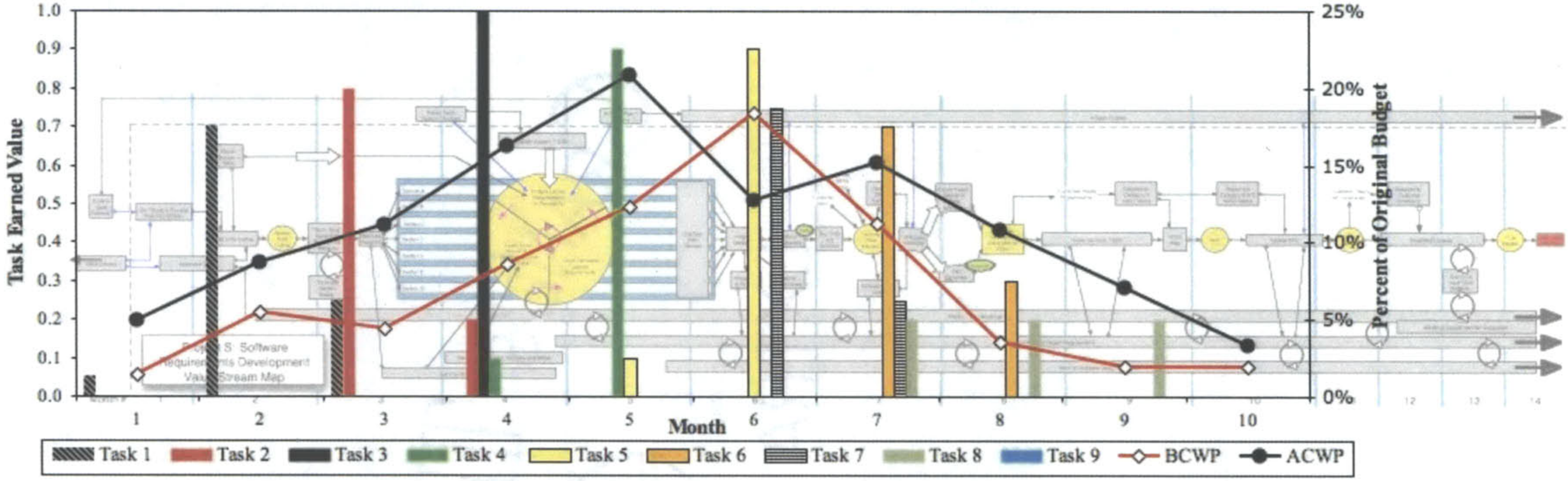
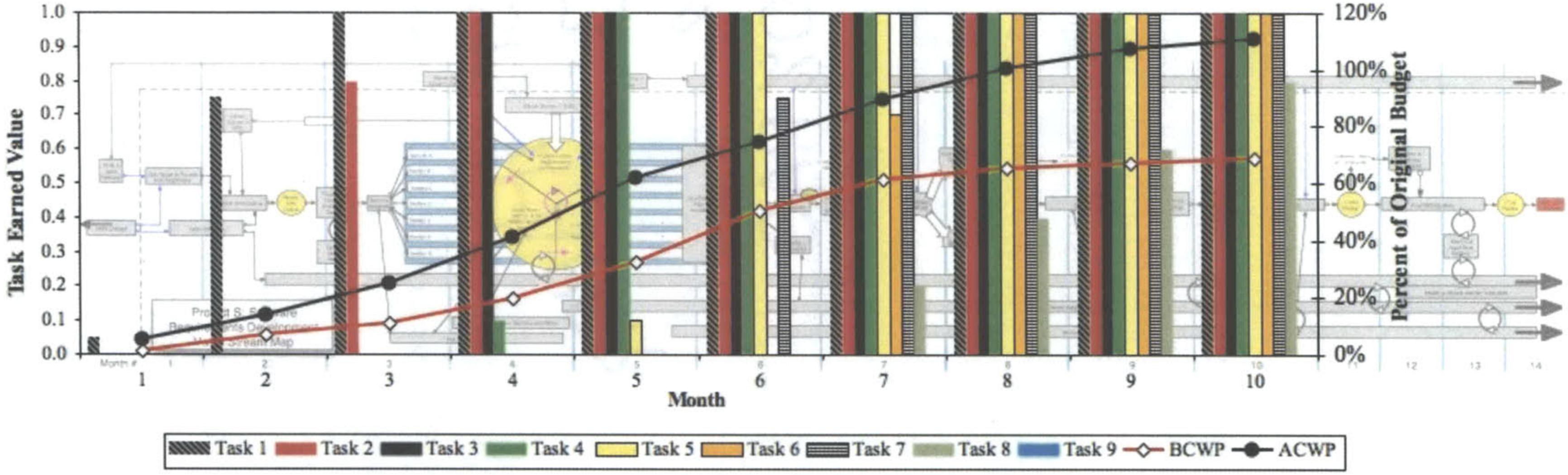


Figure 4.13 Project S Cumulative EV, BCWP, ACWP Superimposed on PDVSM



4.6 Conclusions

Project S has been successful from a quality standpoint. It was the first of the eight software subsystem SRSs submitted for customer delivery and it was one of the first ones to be released to "Rev —". All customer comments have been addressed and the customer is satisfied with Project S product. From a schedule and budget standpoint, however, the project has underperformed. As is evident from almost every EVMS graph, the number of hours the project took to complete was far larger than what was expected and planned for. This is most likely due to underestimating the work involved with the reuse effort, and to the fact that Project S was not as good of a candidate for reuse as was thought.

Also, the tasks from the WBS do not align well with the PDVSM. While the value stream map is certainly not perfect, it better reflects how value was added to the project than does the EVMS task list. The EVMS task breakdown is based upon number of "shalls" in the book. While this is a simple, understandable metric, it is perhaps not the best way to break down the process value.

A couple of problems arise with just using the number of SRS "shalls" to estimate process value. First, while a "shall" may be in the book, it may have many outstanding issues to be resolved. Thus, value may be claimed on a certain number of "shalls", but if not a single one of them has been resolved completely, then there is still plenty of value left to be earned. Additionally, in Month 6 there was what seemed to be an explosion of productivity. This seeming great work output is deceptive, because it was during this month that the "shalls" from the various engineers were assimilated into one document, and thereby a large number of "shalls" was reported. Most of these "shalls" probably existed in some form before this month, but were not claimed because they were not in the book. And again, not all of those "shalls" claimed as value in this time frame were complete. Many needed resolution, and the resolution of issues takes time and communication. Having so many issues to resolve was one major cause of the Project S SRS slipping its "Rev —" release date from Month 11 to Month 14.

Overall the project was both successful and unsuccessful in terms of customer value, depending upon how one views it. From a quality point of view, it met all expectations, but from schedule and cost perspectives, it underperformed. Additionally, the way the value of Project S was measured by the EVMS led to problems and overestimating actual progress. Hopefully, though, by taking their time with the SRS development, Company X will be able to produce a better quality Subsystem P and System C.

5 Project R: Hardware Prototype Development

5.1 Introduction

In this project, engineers worked to create a hardware prototype for a single antenna from a multi-antenna communication and radar system. This radar system will be incorporated into a much larger and more complex system (which itself is in the design stage). This project was much more complex and lasted longer than Project S. Because it is so large, only portions of its value stream map have been chosen for analysis.

5.2 Project R Context

Project R must be understood in relation to its context. This requires a careful examination of the program structure from multiple angles because of its complexity. Project R is the development of a single antenna for a communications system prototype, which is part of a much larger spiral development process. Spiral development involves iterations of design, production, and testing to make a final system that performs better than a single design process might. It implies a high degree of parallelism in the early stages, and accepts inherent inefficiencies as the overall system definition, partitioning, and concepts evolve together. As indicated below in Figure 5.1, a family of three different systems (D, E, & F) are to be designed, built, and fielded in a very long and expensive spiral development process. Several new technologies are needed to field all of these systems, and these technologies are being developed as prototypes within the development of System D. These prototypes, henceforth referred to as System D Prototypes, or SDPs, are multi-million dollar development efforts themselves. Information from testing the SDPs will be fed into the final design of System D. After System D has been built and fielded, lessons from it and from the SDPs will be used in the design of System E. Similarly, lessons from the SDPs and both Systems D and E will be fed into the development of System F. The development and manufacture of System D will take many years to complete (Systems E and F will not be complete for decades), and various parts of its development will inevitably serve as the basis of many theses. This chapter only presents a small sliver of the System D development effort.

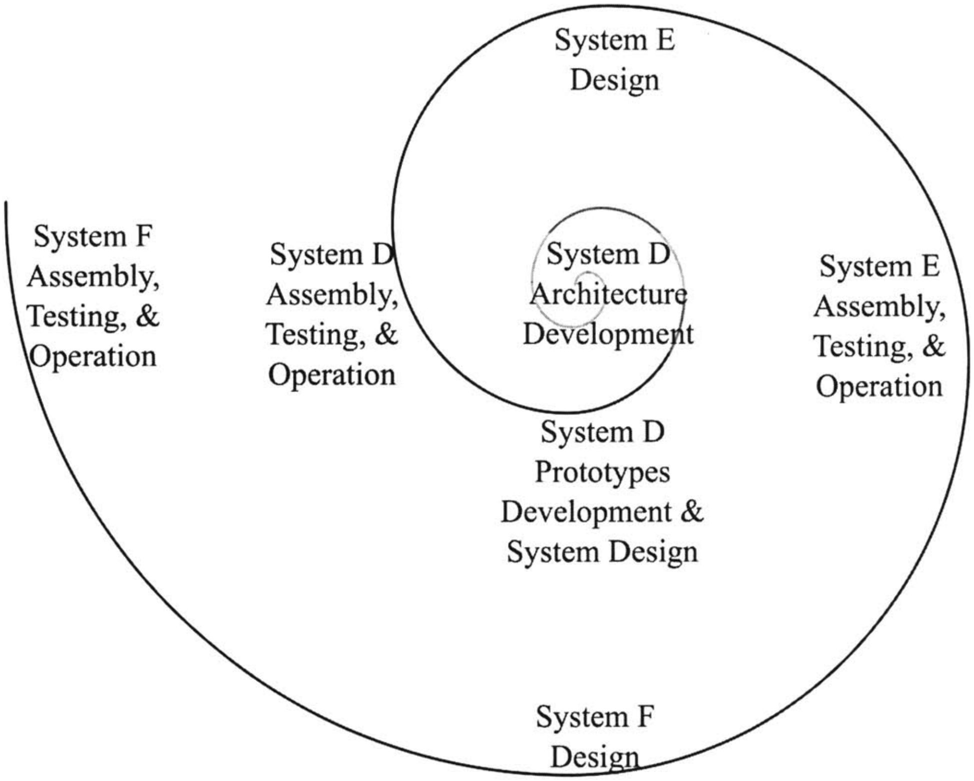


Figure 5.1 System D Spiral Development Context

Project R is a subset of one of the SDPs being created for System D. Figure 5.2 below indicates how Project R is a part of System D Prototype I. SDP I is a mock-up of Subsystem I, the housing that will incorporate many antennas. Some of these antennas only transmit signals, some only receive, and some do both. SDP I will be tested with to see to what extent the antennas interfere electromagnetically with one another. The goal of Project R is to design, build, and test one of the many SDP I antennas, Antenna N, prior to its integration with the others for the full SDP I testing.

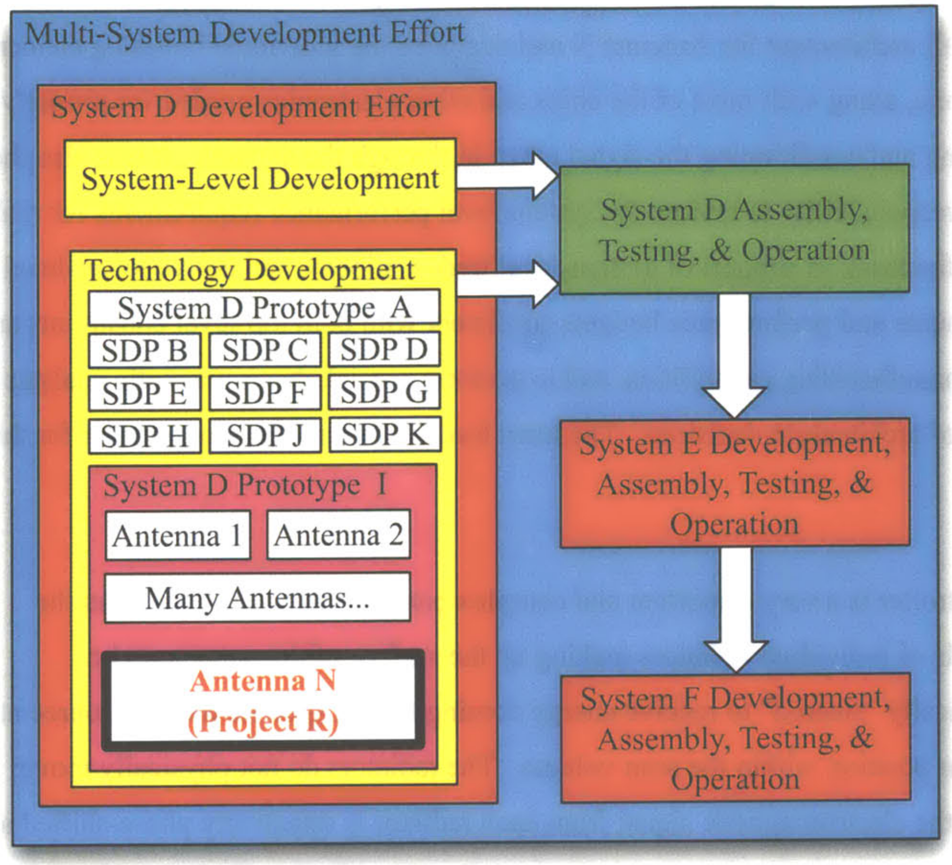


Figure 5.2 Project R Context

5.3 Project R Specifics

5.3.1 Team Structure

The Project R team consisted of many highly-specialized engineers working in six major groups. Each group consisted of one group leader and 5 to 15 engineers. The size of each group varied over time depending on workload demands. One overall Project R leader was responsible for the team, and he met regularly with the group leaders.

The six groups involved were the analog & radio frequency (A/RF) electronics group, the controller software group, the controller hardware group, the power group, the mechanical engineering group, and the procurement group.

The A/RF electronics group was essentially the lead group for the project. They created the overall architecture for Antenna N and designed the individual radiating elements of the antenna, along with most of the chips and other electronics needed for properly combining and conditioning the signal taken in through the antenna. It was also the A/RF group's responsibility to review the system level performance requirements for realism, provide feedback in support of system-level trades, allocate and derive lower-level requirements and performance budgets consistent with both top-level constraints and design/manufacturing capabilities, and to perform crucial electromagnetic analysis in support of architecture decisions. The team leader was also the group leader for this group.

The controller is a very important and complex part of Antenna N. It allows the thousands of individual radiators making up the surface of the antenna to be electronically "steered" to receive energy coming from a specific moving source at a particular position within the scan volume. The radiators do not physically move; instead, the electromagnetic signal from each radiator is selectively phase-shifted and combined to construct a beam in the desired direction. Both software and hardware portions of the controller are required to achieve this complex task. The controller development was therefore executed by two groups, one for the software and one for the hardware. There was a single controller behind all of the A/RF electronics that controlled the most of the antenna, but some portions of the controller hardware were distributed within the other Antenna N electronics. This approach provides maximum performance and minimum computing resources with only a slight increase in system complexity.

The power group was responsible for providing the proper amount of power to the antenna at the right voltage and current levels and with the proper conditioning. They had to make sure that the power was directed and distributed correctly within all the electronics.

The mechanical engineering group was responsible for determining the physical arrangement and interconnection of all the parts of the antenna, its electronics, its support structure, and the coolant system. They did analysis work on the various hardware subassemblies, and also produced all the drawings and diagrams for production.

The procurement group was responsible for obtaining materials and parts from vendors. They had to establish relationships with the vendors, obtain quotes for all materials and parts, integrate and manage the part delivery schedules, and ensure that the delivered parts functioned according to requirements. They also built some of the components for the SDP antenna that required rapid turn-around.

5.3.2 Communication

Program information and design data was communicated to individuals through the team hierarchy. The Project R leader met with all the group leaders at a weekly meeting, where major issues were often discussed. The team leader and the group leaders used a Microsoft Project schedule for managing the tasks involved in the project. When major Project R subsystems required communication between groups, individual engineers met with members of other groups as necessary, independent of the group leaders. Each group had its own internal weekly (at least) meeting. The members of the groups generally sat near one another. They mainly communicated face-to-face or over phone or email, and often exchanged large electronic data files. Shared network drives were used for storing the data shared within and among the individual groups.

5.3.3 Project Issues

In this complex project, many issues arose that impacted the project and ultimately led to its restructuring. From the date of the contract award, controversy surrounding the bid evaluation process delayed the program start by several months; however, the antenna delivery date was not allowed to change. This resulted in a severely compressed schedule before the work even began. Other major issues specific to Project R included the fluctuating antenna requirements, the "deliverable" status of the antenna, the interplay between the design of the SDP version of the antenna and the production version, the "common equipment", the repeated de-scoping of the SDP antenna size, the "Plan B" to mitigate risk, and budget problems.

5.3.3.1 Antenna Requirements

Much to the dismay of the engineers working on Project R, agreement on a set of top-level requirements for the antenna was not reached until late in the design cycle. The engineers were given suggested overall performance capabilities, along with some high-level requirements for the antenna, but gaps in the specification lingered throughout the design phase. By the antenna Preliminary Design Review (PDR), the team had received only what were called "good enough requirements", but these were incomplete at best. For example, a number of performance parameters were listed as To Be Determined (TBD). Determining these parameters often required decision to be made at higher levels of the System D design or by the customer technical experts, or had to be vetted by many different groups across the program and across the country. This process was slow, and was often trumped by short-term crises that had higher visibility. As a result, the various groups were updating the requirements for their parts of the antenna until the antenna Critical Design Review (CDR). The requirements for each of the groups should have been settled before the Advanced Design Review (ADR). Fortunately, by designing the SDP version of the antenna, the team had a good start creating a set of requirements for the production version.

Many major requirements changes came from higher levels within Company X or the System D leadership as the overall vision for System D changed. These changes included an added requirement that the antenna support communication with an additional class of sources, major geometrical and configuration changes to Subsystem I, and operating system changes for the controller software.

The fluctuation in the requirements led to many changes and rework over the course of the antenna development. These changes often occurred so rapidly that it was difficult if not impossible for the formal plan documentation to keep pace. Not surprisingly, the ultimate result of these changes was that the project was behind schedule and over budget as compared to the original baseline.

5.3.3.2 "Deliverable" Status

Throughout Project R, there was a constant debate among the team lead, the System D leadership, and the Company X leadership about whether or not Antenna N was "deliverable". The term "deliverable" carries with it a very specific meaning with respect to government contracts, and such product require a great deal of care in their design, manufacture, and testing. Usually, a piece of hardware is considered "deliverable" if it is a production item which is considered to be in final, full-functional shape. Since Antenna N was going to be integrated with SDP I for testing at a customer facility, some people thought that Antenna N was "deliverable"; others, however, felt that since it was just a prototype, it was not to be considered "deliverable" but merely "shippable". Further, the original expectations of those planning the antenna design never included the added effort that accompanies a "deliverable" product. The debate raged on, but no strong decision was ever made. However, a de facto decision emerged, as the fight against the "deliverable" status was eventually given up. Essentially, although no formal decision had been made, the customer expectations gradually increased until the program office directed the team to proceed as if Antenna N were "deliverable".

Making Antenna N "deliverable" led to a great deal of work. Drawings had to be updated formally to reflect any deviations from the design, stenciling and paperwork had to be redone, and a more rigorous regimen of quality inspections and approvals were demanded for the antenna. The impacts of this decision were fel by all six groups.

Additionally, this "deliverable" status led to CDR being a much more difficult process than originally planned. In the initial project planning, the CDR was going to be a single-day internal review where System D technical experts and program leadership could review the Antenna N status and identify any loose ends to be tied up. However, the debate over the "deliverable" status of Antenna N led to the program office elevating the CDR to a customer-level review. The customer presence meant that Company X had to add internal meetings to review the design information prior to the customer seeing it. First, each group held their own peer review to collect their information. Next, there was a week-long Company X internal review. Due to customer schedule issues, a month passed before the customer CDR. Overall, the CDR ballooned from just a peer review and one-day internal CDR to a massive two weeks of review, one-month delay, and then a three day "dog and pony show" for the customer.

5.3.3.3 SDP Antenna vs. Production Antenna

The team of engineers that worked on creating the SDP version of Antenna N (Project R) also worked on creating the production version as well. As planned, the team was to work on a single design, and then break it into the SDP and production versions once an acceptable design freeze point was reached. These separate designs would be done in parallel, and after the SDP testing was complete, lessons would be fed back into the production design. Unfortunately, an adequate design freeze point was never reached, and thus the engineers worked primarily on the SDP version (Project R), and then on the production version after Project R ended. This change of plans is indicated below in Figure 5.3. In order to avoid any confusion, all further references to the antenna will be to the SDP version of the antenna. This prototype is the focus of Project R. Any reference to the production version of the antenna will be specified.

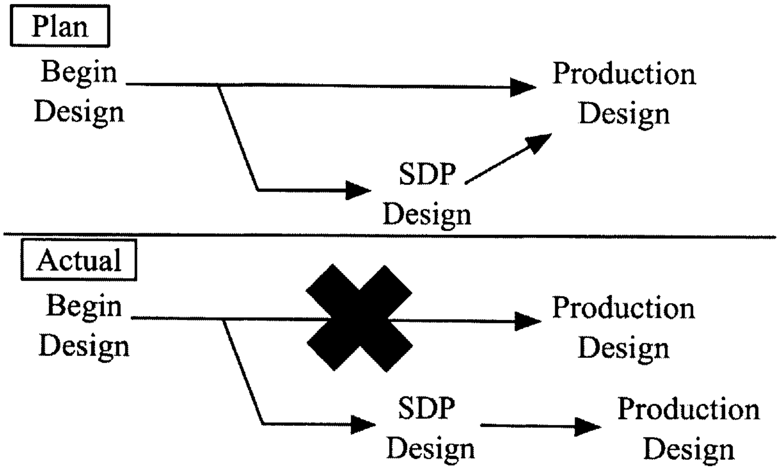


Figure 5.3 Relationship Between SDP and Production Versions of Antenna N

5.3.3.4 “Common Equipment”

As well as designing and building Antenna N for the SDP I testing, several of the engineers on the Project R team worked on supporting equipment for SDP I. This equipment that they were designing was to be common among all of the antennas in the SDP I testing and was developed concurrently. Thus, many of the engineers had to work on both Project R and the “common equipment” design at the same time (with the common equipment design requiring a great deal of interface with the companies developing the other antennas). The common equipment design effort has been excluded from this chapter, but it unquestionably added to the engineers’ workload.

5.3.3.5 De-Scoping of Antenna Size

The final *production* version of Antenna N for Subsystem I is planned to be very large and contain thousands of radiators. For the SDP, however, it was considered unnecessary to make Antenna N full-sized. Originally, it was planned to have about 25% of the radiators of the production antenna. In other words, Antenna N would have 3 times as much empty surface as it would have covered with active radiators. Prior to the ADR, the requirement was changed to having only 20% active elements. This was requested by the design team and was necessary because of the architecture partitioning. It had no impact on the system performance or design effort because it occurred early in the development. After the CDR, it was further reduced to 13% active antenna. This was a change directed to the design team by the System D’s system engineering group as a result of more detailed performance analysis and projected cost information. When Project R ended, Antenna N was set to have just 6% of the number of active radiators that the production version would have. This also resulted from updated analysis and further attempts to reduce overall non-recurring cost. These changes in the scope of the product forced the Antenna N team to adapt, especially the mechanical engineers working on the structure.

5.3.3.6 Plan B

The preliminary design of Antenna N called for the design of a new kind of a specific circuit and a novel system architecture. This new architecture and circuit promised improvements over previous designs, but there was some risk that the new circuit and architecture would not be feasible. To mitigate this risk, Project R began a more traditional approach as a “Plan B” after the PDR. In “Plan B”, the groups assessed what would have to change if the new architecture and circuit could not be used. For most groups, this meant that new parts had to be designed in parallel with the “Plan A” parts. Since this additional effort was not anticipated in the original plan, this obviously had an adverse effect on the team’s productivity and artificially skewed the earned value reports.

5.3.3.7 Problems with Vendors

Project R suffered significantly from vendors that were unable to deliver the quality and quantity of product as their quotes had claimed. In the case of one major part, the relationship with the vendor was ultimately severed, and a new vendor had to be found quickly. Also, the manufacturer of one of the major Antenna N subassemblies was forced to modify its normal production processes to meet customer-directed schedule cuts that occurred well after the detailed manufacturing plans were in place. The risks were identified and communicated up to the leadership hierarchy. In the end, the risks were realized and units were being produced at a rate less than half of that planned, and. This fact contributed to the aforementioned 13%--> 6% active antenna change as well.

5.3.3.8 Re-Baselining

Approximately halfway through the program, the customer requested significant additional scope changes and asked Company X to suggest tasks in the baseline plan that could be eliminated so that the project would remain cost-neutral. As a result, the entire Project R schedule and budget were re-baselined. Begun in month 10, but not implemented until month 15, the re-baselining effort left much of the project statusing efforts in limbo for a time. Budget was reallocated among the Project R tasks, and some tasks were cut to keep the budget the same. If the team continued following the original plan, they would have performed some tasks that had no budget or added no value to the project. During this time, the team was allowed to continue as was deemed prudent, and each work package’s budget was allowed to increase with the money actually spent each month.

5.3.3.9 Budget Issues and the Restructuring of Project R

At the highest levels, the evolution of the system concept, performance requirements, and architecture drove cost into the development effort and risk into the schedule that was never envisioned in the original plan. Because of the issues discussed in previous sections, Project R also fell behind schedule and over budget. This was true of many of the subsystem design activities that rolled up into the System D design effort, and it was realized that budget had to be cut somewhere. Measurements had recently shown that the antenna that was most likely to interfere with Antenna N actually had a very low probability of interference over all of the ranges critical for mission success. Also, a critical chain analysis had shown that it would be impossible to receive all the parts from the vendors in time to assemble, integrate, and test Antenna N before the SDP I testing. These three facts made it obvious that Project R could be stopped and funds conserved without increasing overall System D technical risk. This led to the end of Project R. However, this was not the end of the team, as those working on common equipment still had to deliver a subset of the common equipment for the SDP I testing. Also, due to the "deliverable" status, all drawings associated with Antenna had to be corrected and brought up to a very precise standard. The SDP Antenna N design was later used as a starting point for the production Antenna N design effort.

5.3.4 The Team and I

I had good interaction with members of the team. I met often with the team leader and on several occasions with the group leaders. Interviews were normally face-to-face, with phone calls and emails used to answer specific questions. The team was willing to provide me with a plethora of data relating to how the project proceeded and official Company X product development processes even if program constraints forced these processes to be modified. Project data included access to the Microsoft Project files used to manage the project and analyze the critical chain of development tasks, several miscellaneous project-specific documents, variance reports, manpower reports, and most importantly, project EVMS data. If anything, I was given access to *too much* data.

5.4 Project R Overall Value Stream Map

A PDVSM was created to examine Project R. In the following sections, the value of Project S is presented, along with a discussion of the project itself. The PDVSM is shown, and its tasks are analyzed for their respective value contributions. Through the process of creating this map, I learned a great deal about the process steps, how they are related, and how the team worked together.

5.4.1 Value Proposition

Again, prior to creating a PDVSM of a process, one must have a good understanding of the overall value of the process. Only then can one begin to digest how that value is created. Project R had several goals that can be used to analyze its value. The first goal was to create a prototype of Antenna N that could prove the functionality of the novel architecture and circuitry used to perform the specified communication function. Another goal of equal or more importance was that the antenna, when integrated with SDP I, could provide valuable information about how well the SDP as a whole performed against the test criteria. Finally, Antenna N was to provide information so that the designers of the production version could have a solid foundation to work from.

Figure 5.2 above is similar to Figure 4.3 in the Project S chapter in that it represents the context of the project. Instead of altering and repeating Figure 5.2 with just the stakeholders added to it, I will state the stakeholders for each level of detail.

For the most part the only interested stakeholder in the multi-system development effort is the customer, who wants to ensure that they get a family of new systems that meet performance requirements. Those interested in the System D development are the customer, the program management office, and the system architect. Each finds value in the development of a system design that is stable enough to allow for further development and inclusion of the new technologies from the SDP efforts. However, they also want the system architecture to be flexible enough to function even without some of the SDP technologies if they prove untenable. All of these same stakeholders are interested in the outcomes of each of the SDP technology development efforts.

For the SDP I effort, there are several interested parties. First, the customer, program management office, and system architect care about the outcome of this important technology development effort, and how this will affect the System D development. Second, all of the companies that are making the various antennas for the SDP see value not only in knowing how well their specific antennas perform, but also in how they interact with the other antennas and the common equipment. They want to be sure that their antennas will be placed on the production version of System D. Finally, Company X is one of these companies, and they also see value in understanding the performance of the common equipment when integrated with the other antennas.

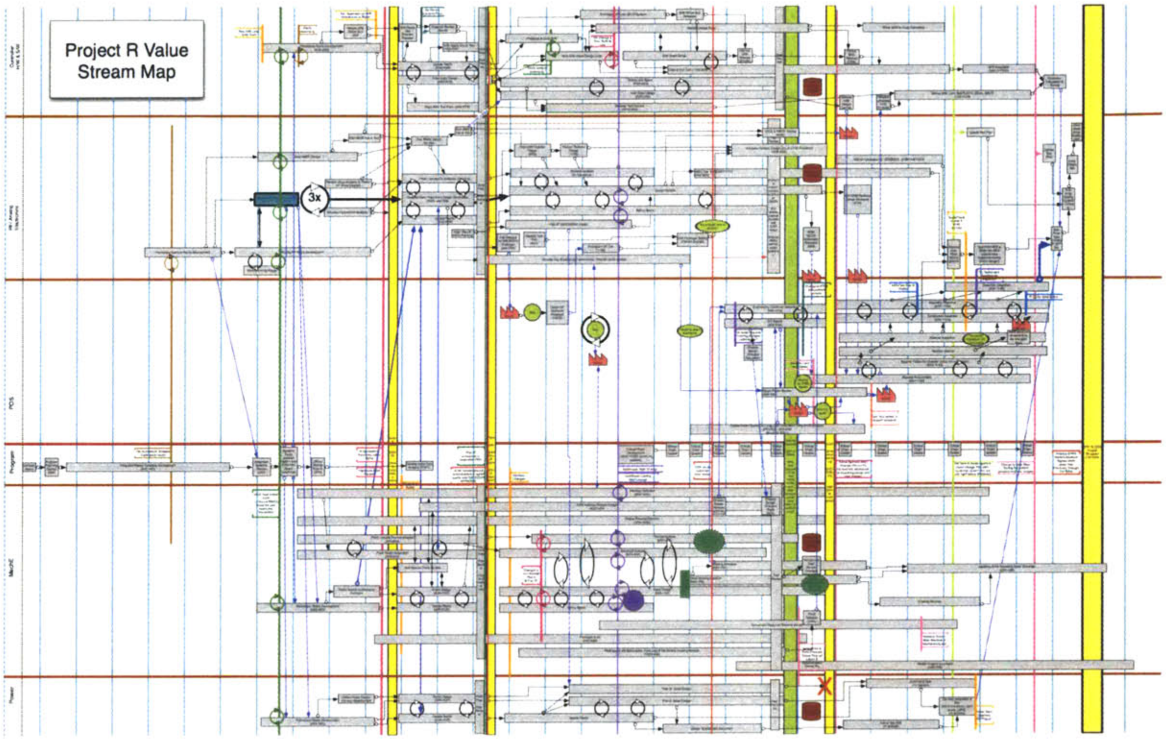
For Project R, many of the same stakeholders are interested. The customer is interested in the performance of this specific antenna, as it will provide crucial communications capability for System D. By creating a functional prototype, Project R provides value to the customer by giving them confidence that the production antenna will be a success. Again, Company X wants its antenna to perform well so that it will be easier for it to design and manufacture the production version to be placed on System D. They also want to use Project R to refine their ability to produce such antennas and new architectures. Every opportunity that they have to expand their realm of expertise is valuable to them. The Project R leader and team members all hold an interest in the antenna's outcome. For most of them, this project has been their primary work component, and they would like to see it succeed. It has provided them with many lessons and taught them a great deal as they have worked on the design and development of the antenna. Additionally, most of them will be working on the production version of the antenna, and these lessons learned, coupled with having Antenna N as a basis, will prove valuable as they begin the production antenna design.

Overall, the value of Project R and its context differs depending upon one's point of view. The higher-level view of the overall project has fewer major stakeholders, and they will not be able to extract their value for years to come. When looking at the project with sharper resolution, one sees that there are more interested stakeholders, who see value in the outcome of the prototype antennas and the learning required to create them and their successors. With the overall value proposition thusly stated, the value stream map can be better understood, and then the value of the individual tasks can be analyzed.

5.4.2 Project R PDVSM

A shrunken version of the full current-state value stream map for Project R is shown below in Figure 5.4. Regrettably, a full resolution image of the map is unavailable for several reasons. First, when printed at the smallest size such that it is legible, it is takes up 8 sheets of paper (4 wide by 2 tall), and this thesis is not conducive to such a large format map. The fact that this much space is required to be able to read the map is a testament to the complexity of Project R. Also, some of the processes in the map refer to specifics of the project and other Company X proprietary information, which cannot be published. Finally, the complexity of the project and map are not conducive to being easily understood. Showing the entire map would most likely be confusing for the reader. For these reasons, the full map will not be displayed or analyzed in detail in this thesis. This is a shame because of the great effort that went into making the map.

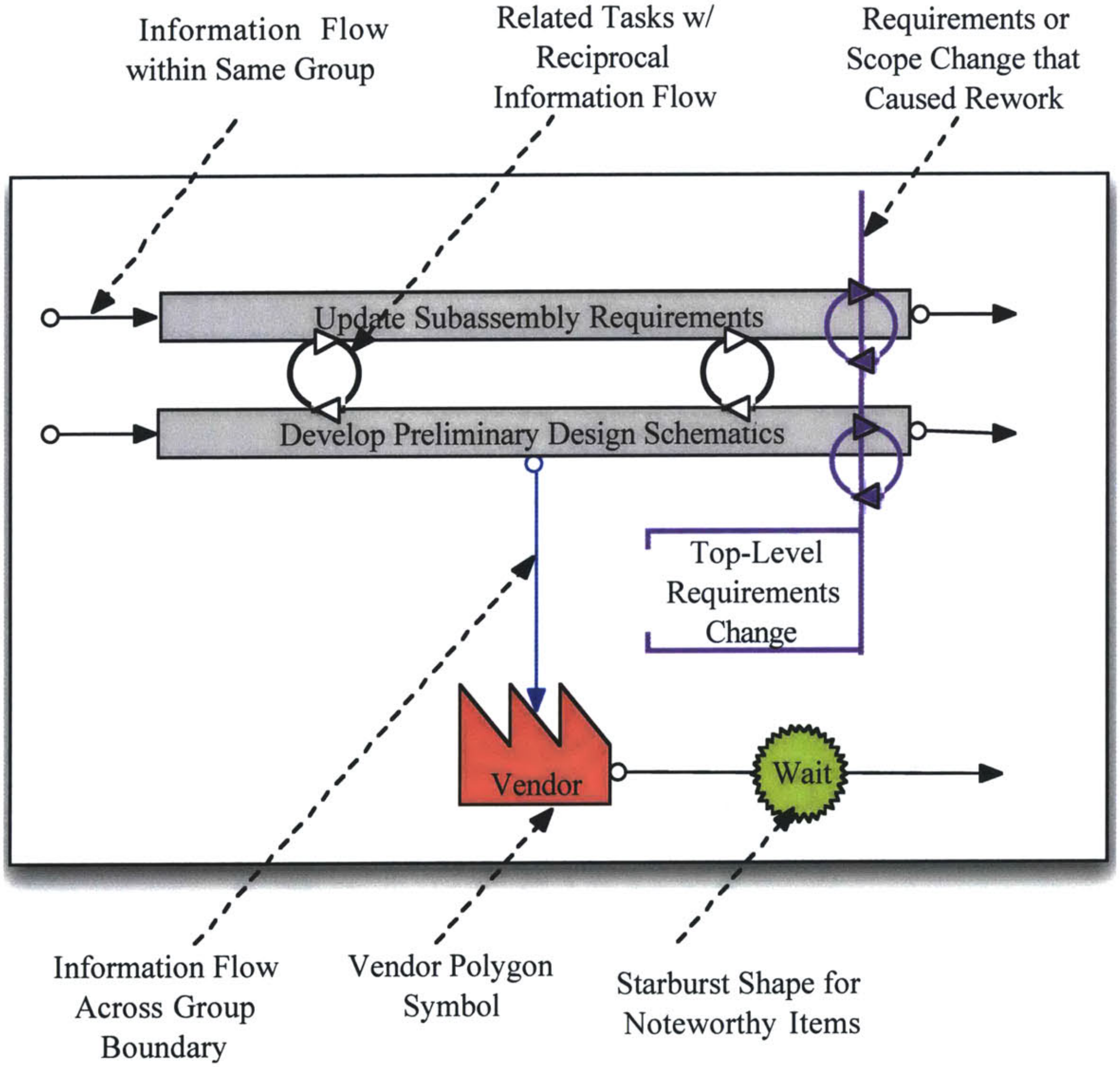
Figure 5.4 Shrunken Full Project R Value Stream Map



From looking at Figure 5.4, one can figure out the format of the full Project R Value Stream Map. Tasks are shown as long grey boxes, connected by arrows to indicate the flow of information (and therefore value, according to Graebsch (2005) and others) between them. As is usually the case, time flows from left to right, and is broken up into a timeline consisting of monthly intervals, as Morgan (2002) suggests. The map is also broken up into six horizontal bands (as Morgan espouses); one for each of the groups (controller hardware and software, A/RF electronics, procurement, mechanical, and power) and one for other project management and planning tasks. Tasks and events that involved more than one group are indicated by tall boxes that stretch across the bands. Readily apparent are the ADR, PDR, and CDR, which stretch the full height of the map. A similar box on the far right of the map indicates when the project was essentially shut down, and a box to the left of the CDR indicates the week-long internal review that happened a month before CDR.

As discussed, there were several changes to the system requirements and scope, among other outside influences, which affected the course of the project. Such changes are marked by a solid colored vertical line, which represents *when* the change happened or was put into place, and which stretch across the bands of the groups that were affected by each change. Text next to each of these lines indicates what the content of the change was. Many of these changes caused rework of some tasks, and the tasks that had to be reworked are indicated by colored circles with arrowheads. The lines and circular arrows are colored the same for each change.

Figure 5.5 Examples of VSM Formatting Items

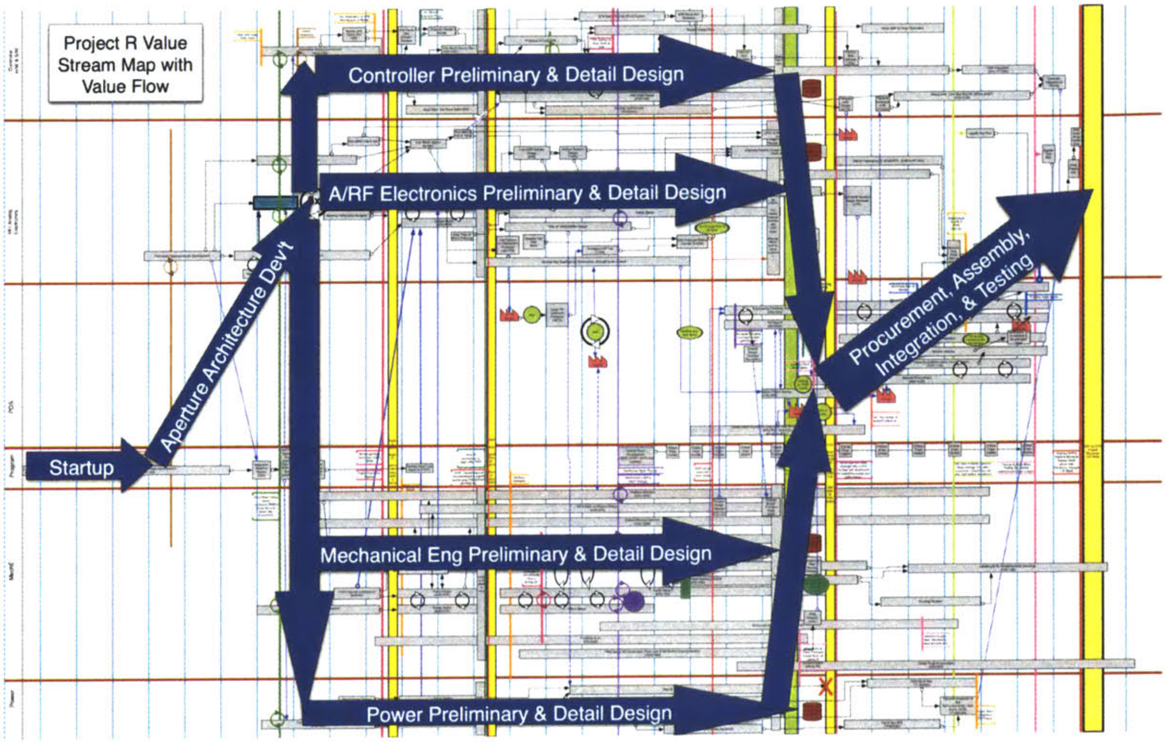


There are several other interesting representational aspects. Noteworthy facts or events are marked with starburst shapes that contain text to explain their content.

Information that flows across the bands is indicated with a blue arrow instead of the black reserved for intra-band information flow. Additionally, a special polygon is used to indicate when interaction with a vendor is required. This shape is similar to the shape used in *Learning to See* value stream maps (Rother and Shook, 1999). Furthermore, some sets of closely-related tasks were performed with a back-and-forth flow of information. Such sets of tasks are indicated with a circle having two white arrowheads connecting the boxes. Examples of these formatting items are shown above in Figure 5.5.

Because the map is complicated and the flow of value is not as readily apparent as it was for Project S, a second view of the map is shown below in Figure 5.6. It is identical to the above figure, but is laid over with arrows that indicate the major flow of information (and thus value).

Figure 5.6 Project R VSM with Value Flow Indicated



Value starts flowing at the left, as project planning is executed and general antenna performance specs are loosely assigned. Then, the A/RF electronics group develops an initial set of antenna requirements and also creates and analyzes the antenna architecture. Once the architecture is established, each of the four major developmental groups (controller, A/RF electronics, mechanical engineering, and power) can proceed with their respective design tasks. The development effort divided along hardware subassembly lines managed by integrated product teams (IPTs) with representation from the major disciplines involved. Responsibility for the IPTs was assigned to the discipline leader who had the most design activity related to the subassembly. The IPTs met weekly to ensure that each designer was working to the latest information. While there is a considerable amount of cross-group communication (especially between the mechanical engineering group and the others) for the more complex hardware items, the predominant flow of information for the simpler assemblies during preliminary and detail design stays within each respective group. Around the time of CDR, each group sends out its design for review by technical subject matter experts and management and then material and component procurement begins. Once parts have been obtained, they are inspected and unit-tested. The antenna parts are then assembled, integrated, and tested (indicated by the last, upward-pointing arrow).

This map is rather large and complex. To make this value stream map easier to understand for the reader, and to enable a value analysis of its tasks, a two-pronged attack has been chosen. First, a higher-level abstraction of this map was made, which removes size and complexity from the map. This map essentially offers a lower-resolution view of the process. Contrarily, a set of higher resolution maps have been made. These maps illustrate the development tasks associated with the development of three major subassemblies of Antenna N: the controller, the radiator, and the structure. By focusing on such smaller development effort, more detail can be shown. For the high-level map and each of the subassembly maps, a value analysis of the tasks has been performed, and EVMS data is presented. Also, the EVMS data for each subassembly is compared to its respective VSM. The high-level map and analysis is in Section 5.5, and those for each of the subassemblies are in Section 5.7-5.9.

5.4.3 Mapping Effort

Similar to the Project S PDVSM, the mapping of this project took several iterations. First, I interviewed the team leader and the group leader to learn about the project and how team communicates. I acquired several process-related documents and began to try to understand the Project R process. I then met with the team leader and all the group leaders for a PDVSM activity, in which I asked them all to mark the tasks performed by their groups with Post-It Notes on a *very* large piece of paper (about 4'x20'). After transcribing this task list, I was able to start making connections between the tasks and get dates for when the tasks occurred. I made several maps, and met with the team leader and group leaders to get feedback on the most recent map. After six iterations on paper, I used the same software to create several electronic iterations.

Several issues arose during the process of refining the map. First, people forget, and data gets lost. This simple fact led to a lot of guesswork on my part to make the map accurate. A lot of Project R documents did not list the actual dates for the start and end of various tasks. A number of tasks were given similar dates which spanned far longer than any of the given tasks should have taken. During program execution, this provides some buffer against constant plan updates as the program scope fluctuates, but it sacrifices some of the resolution needed for value stream mapping. The main reason why people forgot things was that my collaboration with the Project R team did not begin until several months after CDR and well into the manufacturing phase for the SDP antenna. Thus, by making the map as the project progressed would have very much improved its accuracy. Also, when dealing with such a complex process, there are hundreds of process steps. Trying to map this many steps and their interactions to a two-dimensional surface is a difficult task. One must continually make decisions about where tasks should be placed relative to the others and how to connect each set of tasks to indicate how the information flowed in the project. Overall, this was a difficult, time-consuming task.

5.5 Project R High-Level Data

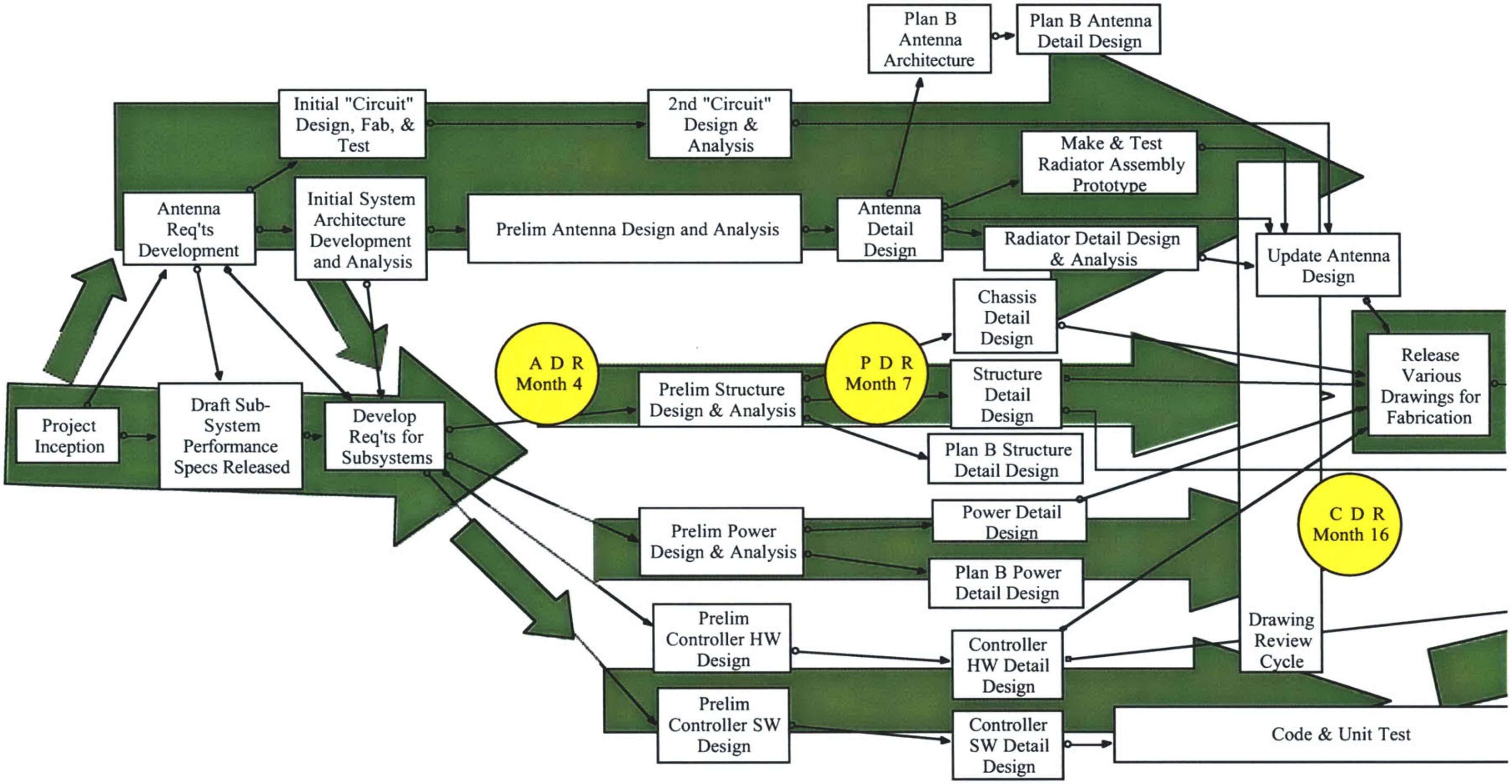
5.5.1 Project R High-Level Value Stream Map

Shown below in Figure 5.7 is the Project R High-Level Value Stream Map. It was created as an abstraction of the full Project R Value Stream Map, and each box on this map represents one or more tasks from the full map. Unlike the full detailed map, this one is not inherently time-scaled. The arrangement of the boxes indicates the approximate relative time in which each abstracted task was performed, but the boxes are not stretched out like they are in the full map (as they are in the detailed subsystem value stream maps). Arrows indicating the flow of value are shown behind the boxes. These arrows correspond to those in Figure 5.7 (although the controller arrow in a different place), and help one to see how the high-level VSM relates to the full detailed VSM. Design reviews are indicated by circles.

While each of the tasks on this map were required for project progress, some were more important than others. The project inception, antenna requirements development, system architecture development and subsystem requirements development set the stage for the preliminary design of the various antenna subsystems. The detail design fleshed out these designs to create a set of drawings that could be sent off to vendors for fabrication. Once materials were procured the various parts were assembled, integrated, and tested.

(This page intentionally left blank)

Figure 5.7 Project R High-Level Value Stream Map



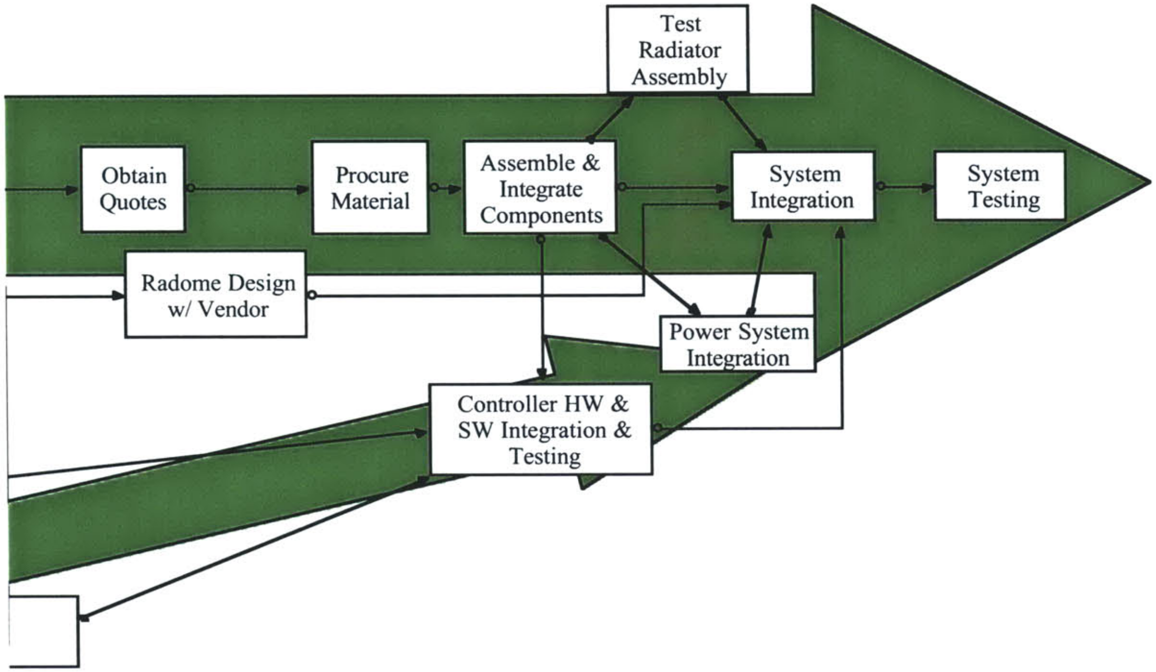


Figure 5.7(cont) Project R High-Level Value Stream Map

5.5.2 Project R High-Level Value Analysis

Table 5.1 Project R High-Level Value Analysis

Task	Product/ Process	Time	Cost	Quality
Project Inception	Process	Establishes schedule	Establishes budget	
Draft Subsystem Performance Specs Released	Process & Product	Sets goals; Determines the effort req'd => Amount of time and budget red's		Establishes quality demanded
Antenna Req'ts Development	Process & Product	Sets goals; Determines the effort req'd => Amount of time and budget red's		Determines what product will be like
Initial System Architecture Development & Analysis	Product	Determines the effort req'd => Amount of time and budget red's		Establishes product concept
Develop Req'ts for Subsystems	Product			Establishes what product will but
Initial "Circuit" Design, Fab, & Test	Product			Creates & tests quality of product component; reduces risk of failure
Prelim Antenna Design & Analysis	Product			Creates & tests overall product concept
Advanced Design Review	Process & Product	Check status		Checks quality
2nd "Circuit" Design & Analysis	Product			Improves design quality; Tests new design quality
Antenna Prelim Design	Product			Creates overall product plan
Prelim Structure Design & Analysis	Product			Creates & tests quality of product components
Prelim Power Design & Analysis	Product			Creates & tests quality of product components
Prelim Controller HW Design & Analysis	Product			Creates & tests quality of product components
Prelim Controller SW Design & Analysis	Product			Creates & tests quality of product components
Preliminary Design Review	Process & Product	Checks schedule performance	Checks cost performance	Checks quality
Antenna Detail Design	Product			Refines product
Plan B Antenna Architecture	Process & Product	Risk mitigation to reduce likelihood of costly & lengthy rework		Risk mitigation strategy increases chances of quality product
Plan B Antenna Detail Design	Process & Product	Risk mitigation to reduce likelihood of costly & lengthy rework		Risk mitigation strategy increases chances of quality product
Make & Test Radiator Assembly Prototype	Product			Creates & tests crucial product component concept; Test its quality

Task	Product/ Process	Time	Cost	Quality
Radiator Detail Design & Analysis	Product			Creates product component; Test its quality
Update Antenna Design	Product			Refines overall product plan
Chassis Detail Design	Product			Creates product component
Structure Detail Design	Product			Creates product component
Plan B Structure/Chassis Design	Process & Product	Risk mitigation to reduce likelihood of costly & lengthy rework		Risk mitigation strategy increases chances of quality product
Power Detail Design	Product			Creates product component
Plan B Power Detail Design	Process & Product	Risk mitigation to reduce likelihood of costly & lengthy rework		Risk mitigation strategy increases chances of quality product
Controller HW Detail Design	Product			Creates product component
Controller SW Detail Design	Product			Creates product component
Code & Unit Test	Product			Creates product component & tests its quality
Drawing Review Cycle	Product			Checks quality of product
Critical Design Review	Process	Checks status		Gets customer feedback; Checks quality, status, & direction
Release Various Drawings for Fabrication	Process	Enables fabrication & assembly		
Radome Design w/ Vendor	Process & Product	Decreases length of communication cycle with vendor		Ensures vendor design process & product of good quality
Obtain Quotes	Process	Determines time & budget for procurement		
Procure Materials & Components	Process	Enables fabrication & assembly		
Controller HW & SW Integration & Testing	Product			Creates product & tests its quality
Power System Integration & Testing	Product			Creates product & tests its quality
Assemble & Integrate Components	Product			Creates product & tests its quality
Test Radiator Assembly	Product			Creates product & tests its quality
System Integration	Product			Creates final product
System Testing	Product			Checks product quality

5.5.3 Project R Overall EVMS Data

In addition to creating value stream maps, I was able to obtain a considerable amount of EVMS data regarding this project. This information is the closest thing that Company X has to a measurement of the value of tasks. This value is measured against a given baseline scope and budget. These data were extracted from spreadsheets used by the System D program office to monitor the project's status. While there was an immense amount of data, the categories of focus are the monthly and cumulative BCWS, BCWP, ACWP, CPI, SPI, and the monthly earned value claimed. Earned value (EV) was claimed relative to the work breakdown structure, which divided the project into work packages. Each of these work packages consisted of several tasks, which were broken down into milestones and further into "inchstones". Each of the milestones and "inchstones" for each work package were weighted for their relative value. As each "inchstone" or milestone was reached, the weighted earned value for that task was added to the respective work package's earned value reported. This earned value, as well as the cost and schedule information, were reported each month. Overall numbers for Project R are presented here, and specific information will be presented with each of the subassemblies in Sections 5.7 -5.9. Due to difficulties in obtaining the proper data, no overall Project R earned value numbers are presented. Table 5.2 below reiterates the meanings of all the various EVMS acronyms.

Although the EVMS system has its advantages, the urge to use it to provide the complete development story for any project must be resisted. It is only one tool in the program management toolbox. Equally important are the critical chain schedule, manpower forecasting sheets, engineering design notebooks, and team meetings. This last is probably the best indicator of the health of the program, since it provides insight into existing issues and indicators of what lies ahead, things that EVMS cannot do. There is no substitute for direct communication with the design engineers themselves. Another limitation of EVMS is that it assumes that the plan is constructed with perfect accuracy or is at least agile enough to reflect program changes instantaneously. Variances are attributed solely to performance. However, when task scope changes, any measure against the original plan and conclusions drawn from that measure are of questionable value. In a development program, especially early on, the concepts/approaches/trades change almost continuously, and it is difficult to keep the plans updated. This tends to skew the performance indices. In this particular case, changes in requirements and system architecture were made at a higher level in the program. These were not anticipated either in the bid or in the subsequent planning. EVMS therefore loses some of its accuracy.

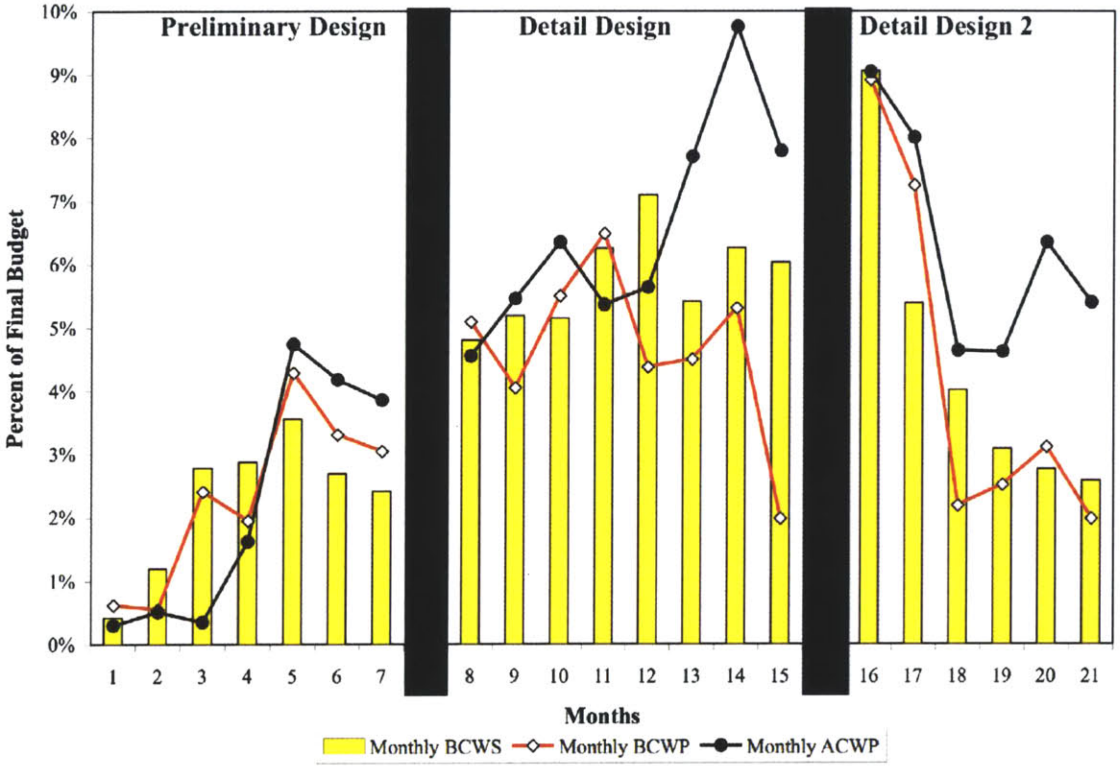
Table 5.2 EVMS Refresher

Acronym	Full Name	My Layman's Terms	Formula
EVMS	Earned Value Management System	"a way to keep track of project progress"	N/A
WBS	Work Breakdown Structure	"a division of the project into smaller tasks that can be completed by the teams assigned to them"	N/A
BCWS	Budgeted Cost of Work Scheduled	"what should get done in the time scheduled"	N/A
BCWP	Budgeted Cost of Work Performed	"what actually got done"	N/A
ACWP	Actual Cost of Work Performed	"what it cost to actually perform the work"	N/A
SV	Schedule Variance	"how much was completed relative to what was expected"	SV = BCWP - BCWS
CV	Cost Variance	"how much it cost to do what was complete relative to what was expected"	CV = BCWP - ACWP
SPI	Schedule Performance Index	"normalized measure of how far away from the schedule the task is"	SPI = BCWP/BCWS
CPI	Cost Performance Index	"normalized measure of how far away from the budget the task is"	CPI = BCWP/ACWP

Figure 5.8 shows the monthly cost and schedule information for Project R, and Figure 5.9 shows the cost and schedule performance indices (CPI and SPI). Together, these graphs suggest how the project progressed. Months 1-7 were the preliminary design phase, leading up to the PDR. In Months 1-4, the team was underproductive but was spending less money than expected, resulting in the low CPI and high SPI. In Months 5-7, the team caught up to schedule by spending a lot of money, which brought the cumulative CPI and SPI back near 1.0. Month 8 marked the beginning of the detail design phase, and the team performed to plan this month. In Month 9, the team was underproductive

and ran over budget, but was able to bring the cumulative indices back near 1 by increasing productivity in Months 10 and 11 and reducing costs during Month 11. After Month 11, the project performance indices fell as each month the team was underproductive and overspending.

Figure 5.8: Project R Total Monthly Cost & Schedule Data

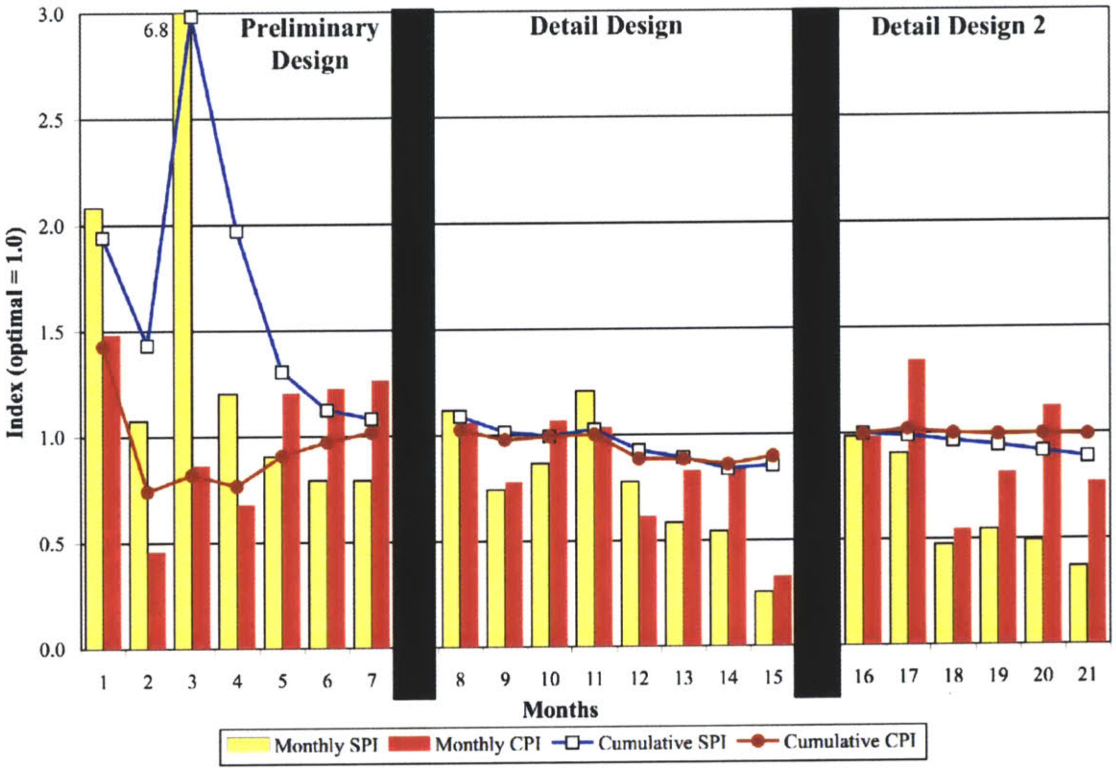


If one assumes a static, representative plan, then based on the cumulative performance indices, the team did not achieve all that it had planned during the detail design phase (and spent too much doing it!). However, the data above does not provide insight into the reasons why the variances occurred. Additionally, the Month 11-15 data has to be observed carefully, as this is the period in which the re-baselining occurred. As will be shown in the subassembly EVMS sections, the data from this time period is rather dubious. Month 16 marked the beginning of the assembly, integration, and test phase of Project R. Some of the subassemblies also continued performing more detail design activities during this time period. The team was nearly perfectly productive this month, and while they were highly productive in Months 17 and 20, they were over budget for

Months 17-21. They ended this period on schedule and over budget. Here again, the limitations of EVMS are evident since nothing in Figures 5.8 and 5.9 indicates that approval was granted by the program office to perform to a 12% larger budget. The baseline was not updated to reflect this decision; rather, the additional funds required were earmarked for another development area without being transferred.

When all was said and done, it was not the budget performance shown here that caused the restructuring near the end of Project R. It was the better-than-expected results from the interferer measurements and high projected recurring cost that drove the customer to reallocate the remaining approved funds to other areas within Project R.

Figure 5.9 Project R Overall Cost & Schedule Performance Indices



5.6 Project R Subsystem Development

Three major Antenna N subsystems were selected for deeper analysis. In the following sections, a value stream map, a value analysis, and EVMS data are presented for the Antenna N controller, radiator, and structure. The controller is broken down further into hardware and software development efforts.

The detailed subassembly VSMs use the same style and format as the detailed overall Project R value stream map shown in Figure 5.4. Since these maps are essentially sections of the horizontal bands from that figure, they are wide and short, which is not conducive to the 8 ½" x 11" paper used to print this thesis. Thus, the maps are broken up into left- and right-hand sections that are then stacked so as to fit on one sheet of paper in landscape orientation. Large arrows indicate where the left half ends and where the the right half begins, and how one flows into the other. The value analyses use the same Oehmen framework to describe the value of each task in terms of how it contributes to the product's or process's schedule, cost, and/or quality.

For the radiator and structure, the EVMS data presented is straightforward. However, the controller development EVMS data is broken further into the electrical development of two hardware components, the mechanical engineering effort, and the development of the software. Thus, there are a total of six sets of EVMS data presented throughout Sections 5.7-5.9. Each set of data was broken into work packages for preliminary design and detail design. These design phases ended with the PDR and CDR, respectively, and according to plan, each of these work packages had to have full value claimed by the review date.

For each of these data sets, three EVMS graphs present data for both preliminary and detail design. One shows the EV claimed monthly for the tasks, one graph presents monthly and cumulative BCWS, BCWP, & ACWP, and one shows the SPI and CPI data. Some additional information is presented for the controller development.

The EVMS data were collected from monthly spreadsheet reports that also included the data for hundreds of other work packages. Through a *massive* effort, the data was reorganized by work package, and reduced to only relevant information for each of the work packages. The earned value claimed for each of these work packages was based upon the aforementioned milestones and "inchstones". Thus, the EV for any given work package can be interpreted as which (and how many) of the milestones that have been completed.

There are a couple of points about this EVMS data to note before studying these graphs. Due to the re-baselining effort, the data for months 11-15 are suspect. First, the available data for Month 11 did not include any data related to the SDP (only had data regarding the design of the production model, which is based on entirely different work packages). For most data sets, the Month 11 BCWS, BCWP, and ACWP were extrapolated based upon the monthly and cumulative Month 12 data. Thus, Month 11 is indicated as "11*" on all the graphs to remind the reader of the dubious nature of this information. Additionally, while the re-baselining effort was progressing, the EVMS accounting was questionable, since BCWS and BCWP are set (after the fact) equal to the ACWP for each month that the baseline was in flux. Thus, the CPI and SPI for each of these months were steady at 1.0. Also, the earned value for each of these months was always reported as 1.0. Thus, the data for these months does is of limited use. The team leader's superiors complained that the project was "running blind" during this time period, attesting to the poor formal reporting on this part of the project. To alleviate this, team leaders manually constructed the true EVMS data and reported progress each month during the re-baselining period.

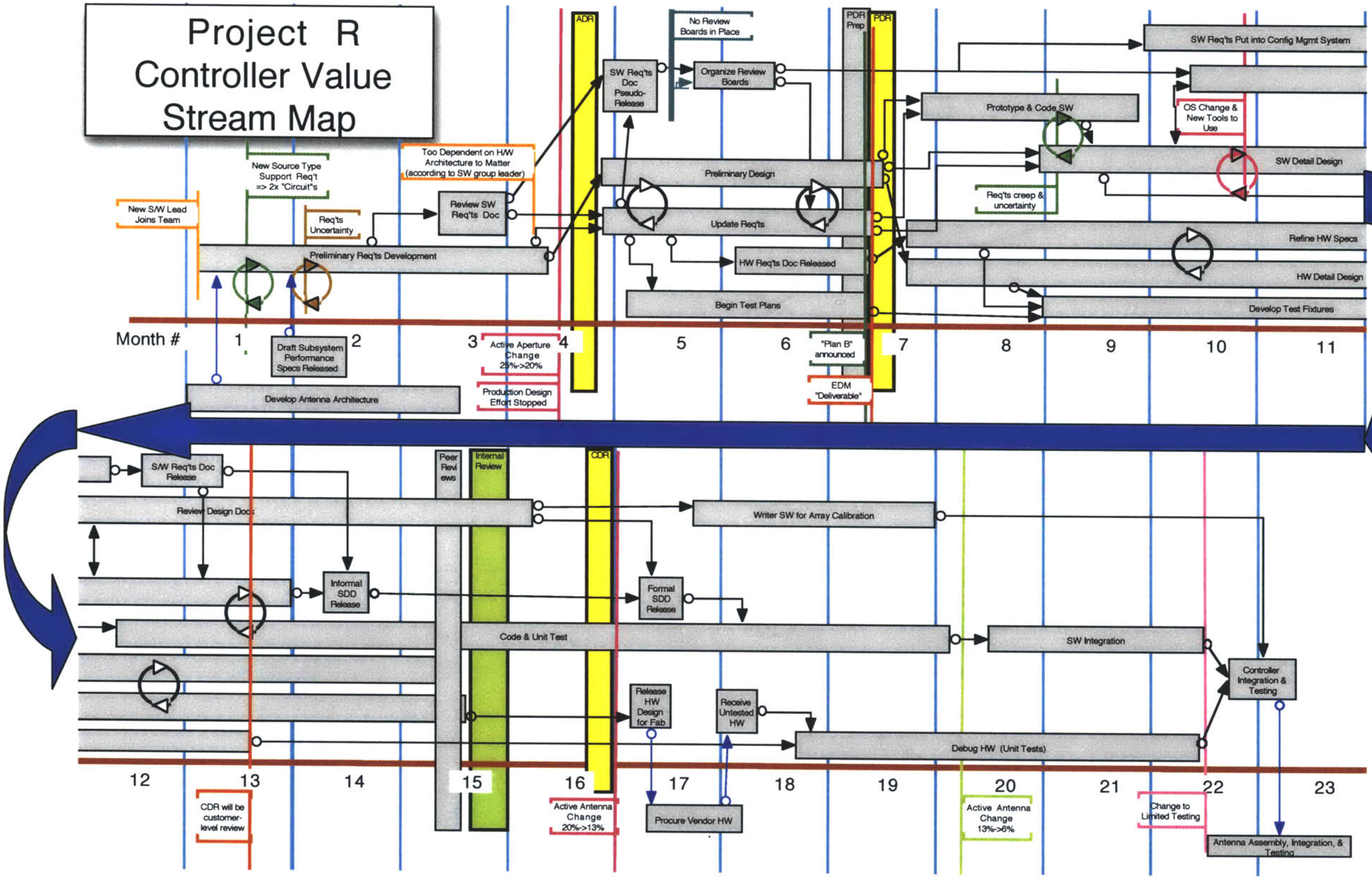
5.7 Controller Development

As previously mentioned, the controller is an essential element that "steers" the antenna to focus upon a signal. It consists of several pieces hardware subassemblies and the software to control them. Excluding the initial requirements development, most of the software development was independent from the hardware development, and the earned value for this process was reported separately. The hardware consisted of several major pieces, all of which required electrical engineering by the controller hardware group, as well as drawings and analyses by the mechanical engineering group before final release. Developing these two major components was a large enough effort was a large enough effort that each had its own work package and had its earned value reported explicitly. The earned value data for the work performed by the mechanical engineering group was reported separately as well.

5.7.1 Controller VSM

Shown below in Figure 5.10 is the value stream map for the controller development effort. Once the antenna architecture was developed and the draft performance specifications released, the controller development began. First, the overall controller requirements were developed, which then were updated after the ADR and used to create the preliminary design of both the hardware and software. Before the PDR, a peer review was held to prepare all controller preliminary design information. After the PDR, the requirements update and detail design of the various hardware components were completed independently from the software effort, which included a prototyping and configuration management cycle. Prior to the CDR, there was another group peer review, followed by an internal Company X review of all the Antenna N detail design. Before the CDR, the hardware design was released for fabrication, and the software was coded and each software unit was tested. After the hardware items were received from the vendors and tested, they were assembled and combined with the integrated software system. The controller was then tested before it was integrated with Antenna N for full antenna testing.

Figure 5.10 Project R Controller Value Stream Map



5.7.2 Controller Value Analysis

Table 5.3 Controller Value Analysis

<table>
<tr><th>Task</th><th>Product/ Process</th><th>Time</th><th>Cost</th><th>Quality</th></tr>
<tr><td>Develop Antenna Architecture</td><td>Product</td><td colspan="2">Determines work effort needed</td><td>Dictates antenna performance & general structure characteristics</td></tr>
<tr><td>Draft Subsystem Performance Specs Released</td><td>Process & Product</td><td colspan="2">Sets goals; Determines the effort req'd => Amount of time and budget red's</td><td>Establishes quality demanded</td></tr>
<tr><td>Prelim Req'ts Development</td><td>Process & Product</td><td colspan="2">Sets goals; Determines the effort req'd => Amount of time and budget red's</td><td>Determines what product will be like</td></tr>
<tr><td>Review SW Req'ts Doc</td><td>Process & Product</td><td colspan="2">Prevents excessive effort using incorrect req'ts</td><td>Makes sure req'ts are good; Ensures right product/ process plan</td></tr>
<tr><td>Advanced Design Review</td><td>Process & Product</td><td colspan="2">Check Status</td><td>Checks quality</td></tr>
<tr><td>SW Req'ts Doc Pseudo-Release</td><td>Process</td><td></td><td></td><td>Attempts to stabilize req'ts and therefore make all work value-added</td></tr>
<tr><td>Prelim Design</td><td>Product</td><td></td><td></td><td>Begins creating product component</td></tr>
<tr><td>Update Req'ts</td><td>Product</td><td></td><td></td><td>Refines what product will be</td></tr>
<tr><td>HW Req'ts doc Released</td><td>Process</td><td></td><td></td><td>Stabilizes req'ts</td></tr>
<tr><td>Organize Review Board</td><td>Product</td><td colspan="2">Creates board for reviewing changes; Allows config mgmt which prevents too many costly and lengthy changes</td><td></td></tr>
<tr><td>PDR prep</td><td>Process</td><td></td><td></td><td>Makes PDR smoother</td></tr>
<tr><td>Preliminary Design Review</td><td>Process & Product</td><td colspan="2">Checks status</td><td>Checks quality</td></tr>
<tr><td>Prototype & Code SW</td><td>Process & Product</td><td>Makes easier to code SW later</td><td></td><td>Ensures req'ts feasibility; Begins creating the SW</td></tr>
<tr><td>Refine HW Specs</td><td>Product</td><td></td><td></td><td>Further defines what product will be</td></tr>
<tr><td>HW Detail Design</td><td>Product</td><td></td><td></td><td>Creates quality component</td></tr>
<tr><td>Develop Test Fixtures</td><td>Process</td><td></td><td></td><td>Enables testing that will ensure product quality</td></tr>
<tr><td>SW Req'ts Put into CM System</td><td>Process</td><td colspan="2">Allows configuration management which prevents too many costly and lengthy changes</td><td></td></tr>
<tr><td>SW Detail Design</td><td>Product</td><td></td><td></td><td>Creates quality product component</td></tr>
<tr><td>Review SW Detail Design Docs</td><td>Product</td><td colspan="2">Prevents excessive effort using incorrect req'ts</td><td>Make sure req'ts are good; Ensures right product/ process plan; Check quality</td></tr>
</table>

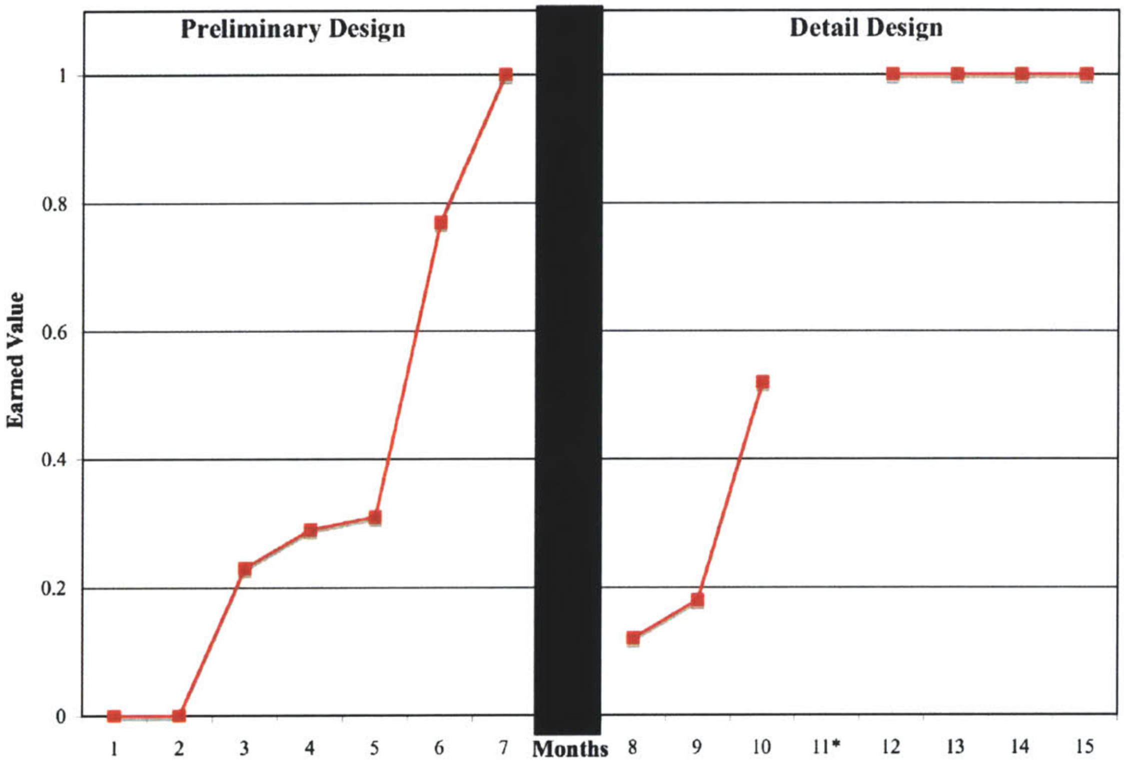
Task	Product/ Process	Time	Cost	Quality
SW Req'ts Doc Release	Process & Product			Stabilizes req'ts
Code & Unit Test	Product			Creates product component & tests its quality
Informal SDD Release	Product			Stabilizes design; Gets feedback to improve quality
Peer Review	Process			Checks quality; Prepares for CDR
Internal Review	Process			Checks quality; Prepares for CDR
Critical Design Review	Process	Checks status		Gets customer feedback; Checks quality, status, & direction
Formal SDD Release	Product			Stabilizes design
Release HW Design for Fab	Process	Allows HW fabrication to begin		
Procure Vendor HW	Process		Obtains HW	
Receive Untested HW	Process	Allows HW assembly, integration, & testing to begin		
Write SW for Calibration	Process			Enables quality testing of SW
Debug HW (Unit Tests)	Product			Improves product quality
SW Integration	Product			Unifies product
Controller Integration & Testing	Product			Unifies product; Checks quality
Antenna Assembly Integration & Testing	Product			Unifies product; Checks quality

5.7.3 Controller EVMS Data

5.7.3.1 Hardware Item #1

Figure 5.11 shows the EV reported by month for Controller Hardware Item #1 (HW1). While the data for Months 11-15 is of limited value, it is clear that no EV was claimed until Month 3 and that a considerable amount was claimed during Months 6 and 10.

Figure 5.11 Controller Hardware Item #1 Earned Value



Figures 5.12 and 5.13 combine to tell an interesting story about HW1. First, through Month 5, the group was far behind the work package schedule, as the BCWP fell behind the BCWS. The monthly SPI bars on Figure 5.13 indicate Months 6 and 7 were very productive. Months 5-7 were very expensive as the team caught up, but ran the work package far over budget. No actual cost was claimed until Month 4, resulting in the misleading spike in the preliminary design's cumulative CPI that month. Month 10 was more productive and less expensive than expected, as is evident from the monthly SPI and CPI.

Figure 5.12 Controller Hardware Item #1 Cost and Schedule Data

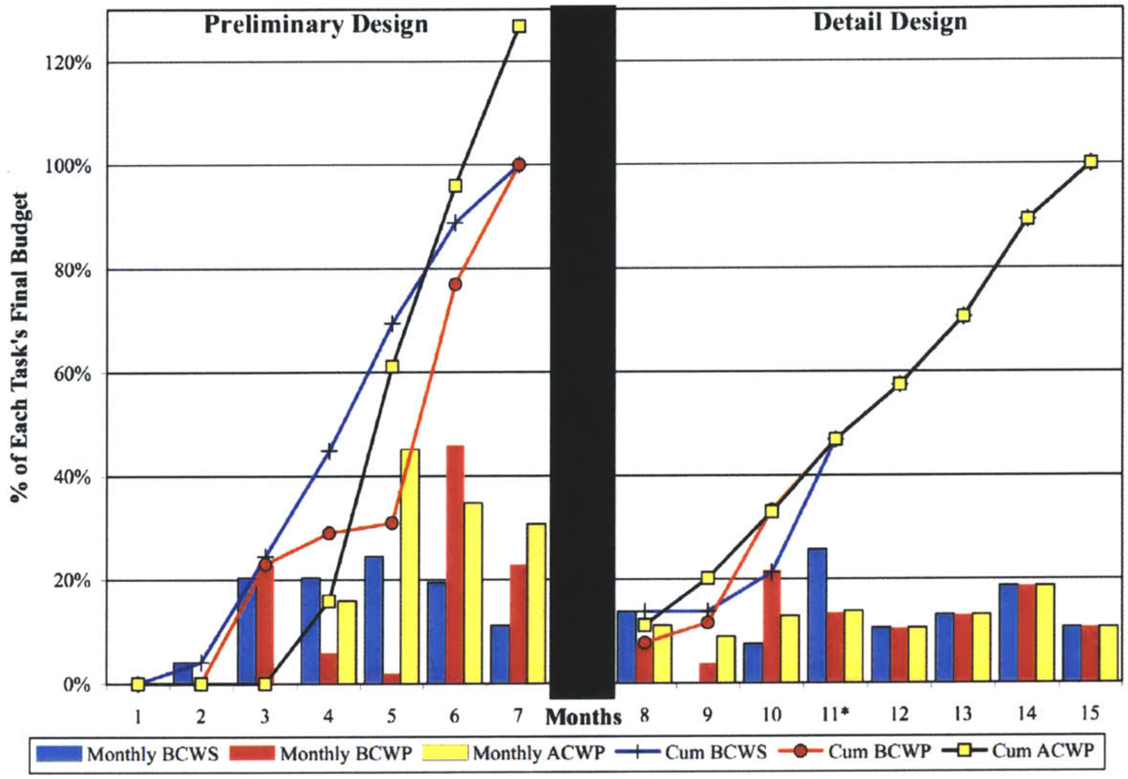
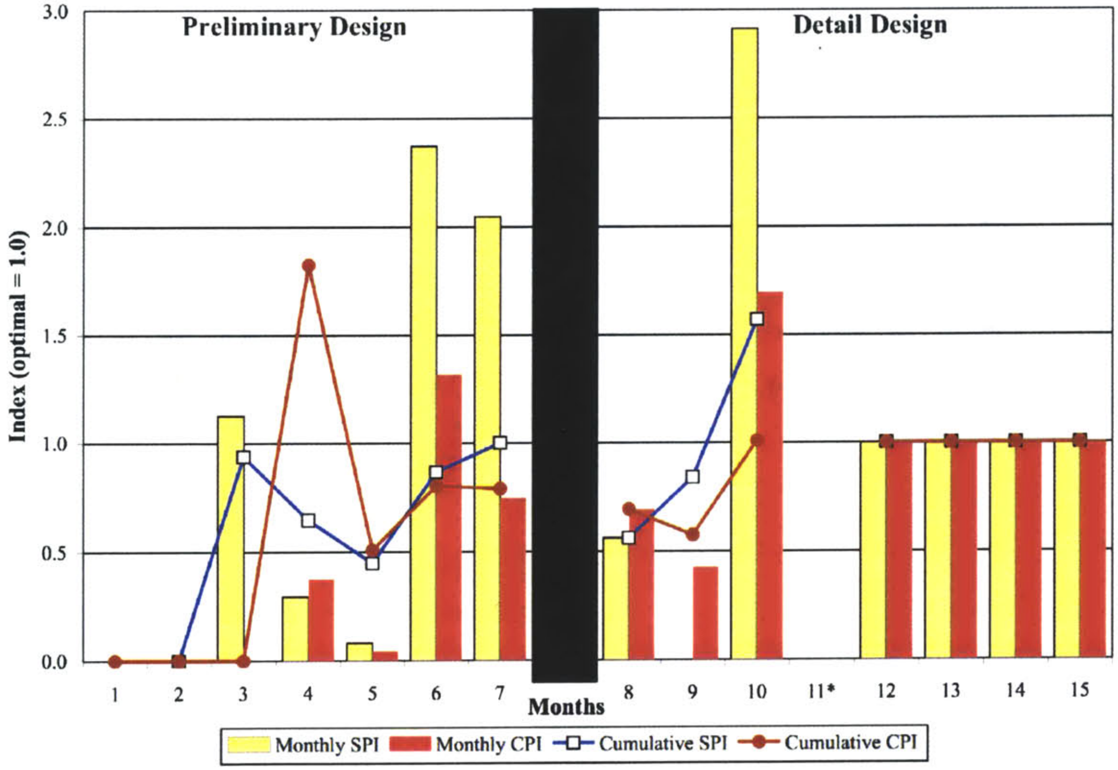


Figure 5.13 Controller Hardware Item #1 Performance Indices



This data is consistent with the fact that the scope of the baseline plan for this effort included updates to existing drawings only, no design work. The original performance requirements for Antenna N were consistent with a design used on another program, but when the additional capabilities were added by the System D systems engineering group, it became necessary to do an entirely new design from scratch. No additional budget was allocated to cover this effort. When the added scope is taken into account, the Controller hardware design team actually achieved productivities slightly better than the typical values achieved for this type of design.

For the preliminary design, the work package was 27% over budget, but met its schedule. One cannot be certain about the relative performance of the detail design from this design since it was "running blind". However, the fact that more than 40% of the final budget was spent after Month 12, when the EV claimed first reached 1.0 (and thus that the work package should have been finished) suggests that the work package ended up far over its original budget. The final budget after Month 15 was 156% of the budget planned as of Month 10.

5.7.3.2 Hardware Item #2

The graphs below for Controller Hardware Item #2 (HW2) tell a story very similar to that of HW1. The preliminary design was behind schedule, until the very productive Months 6 and 7, in which the team caught up to schedule. The one glaring difference is that the ACWP for HW2 did not over shoot the BCWS during this catch up period. In fact, the work package ended up being about 30% under budget, as opposed to the 27% over budget for the similar HW1 preliminary design work package. The spike in detail design cumulative SPI is deceptive, and is an artifact because there was no BCWS reported for Months 9 and 10. A similar 40+% of the final budget was spent after Month 12, when the EV claimed first reached 1.0 (again, which would mean that the work package should be completed), so one might assume that the HW2 detail design was also over budget. Also similar to HW1, the final Month 15 budget was considerably higher than the Month 10 budget; the former was 212% of the latter!

Here again, the explanation lies in the fact that scope increases always result in unfavorable comparisons against the baseline. In this case, more stringent antenna requirements and a request by the customer for demonstrations of additional Antenna N features forced the design of a second version of HW2.When this unbudgeted scope is accounted for, the productivities of the Controller hardware team reach levels consistent with that achieved on other large programs.

Figure 5.14 Controller Hardware Item #2 Earned Value

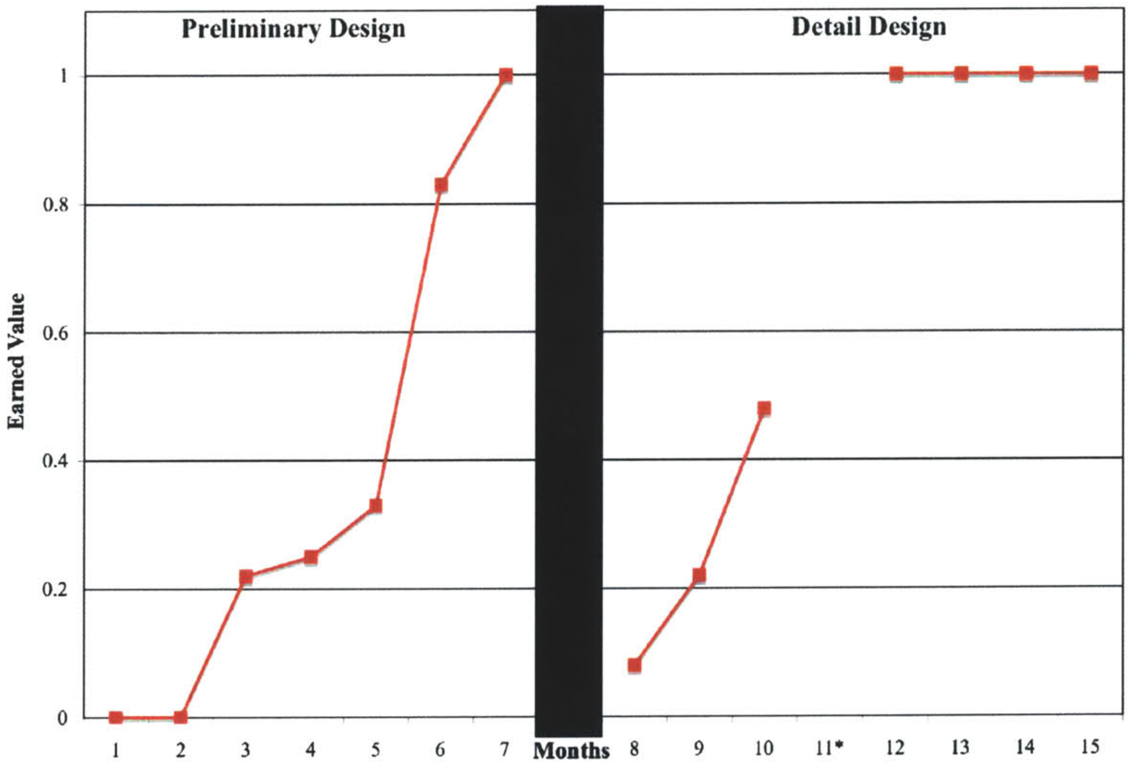


Figure 5.15 Controller Hardware Item #2 Cost and Schedule Data

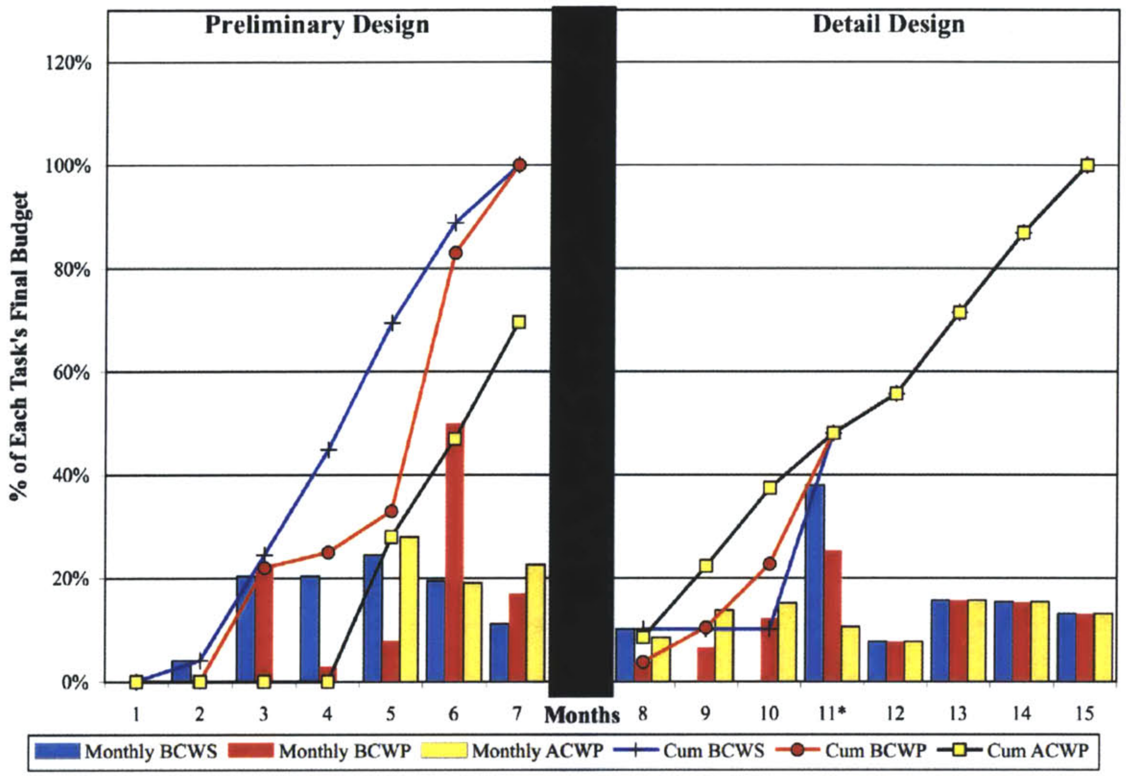
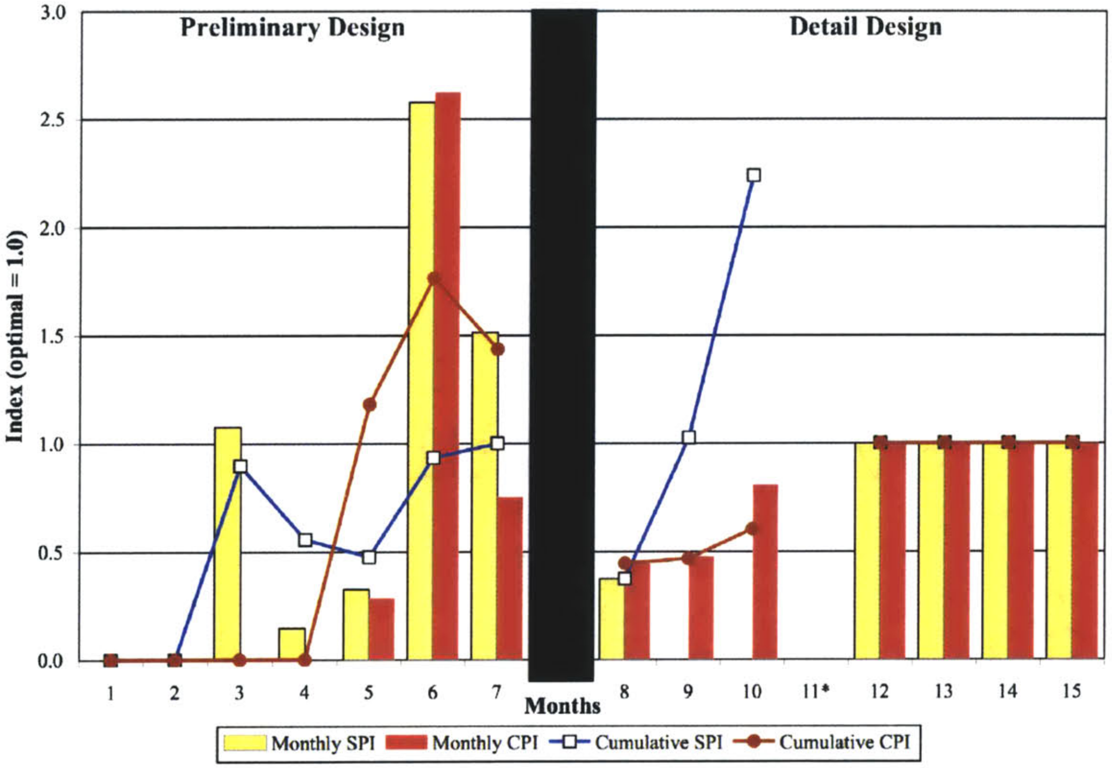


Figure 5.16 Controller Hardware Item #2 Performance Indices



5.7.3.3 Mechanical Engineering Effort

The data from the controller mechanical engineering effort (MEE) in Figures 5.17-5.19 tells a simple story. Like HW1 and HW2, this work package started off behind schedule and the engineers had to accomplish much in the later months of preliminary design. During Months 5-7, the team was able to catch up rather cheaply, and the preliminary design effort ended up more than 50% under budget! For the detail design, the story is rather murky. In the first month, the group accomplished more than was expected far more cheaply than expected, but the next two months, money was spent and very little earned value claimed. The extrapolated numbers from Month 11 indicate that the group was highly productive before the re-baselining period. 38% of the detail design final Month 15 budget (which was 119% of the Month 10 budget) was spent after Month 12, when the earned value claimed first reached 1.0. Again, this indicates that this work package ended over budget.

Figure 5.17 Controller Mechanical Engineering Effort Earned Value

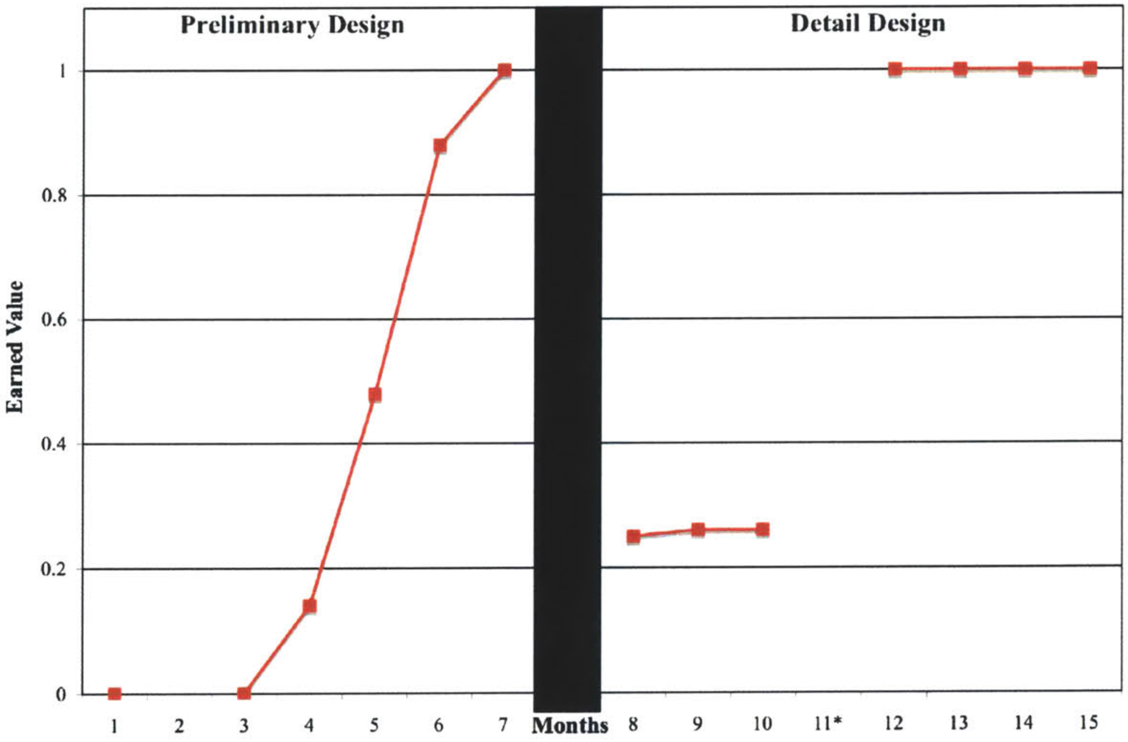


Figure 5.18 Controller Mechanical Engineering Effort Cost and Schedule Data

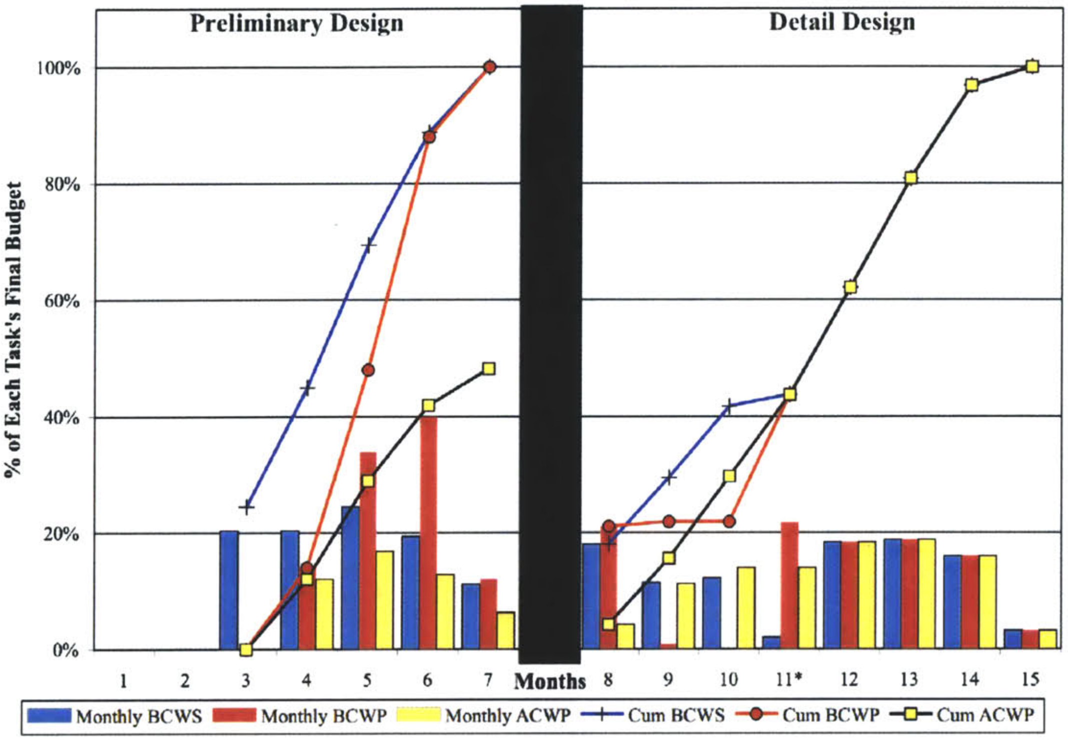
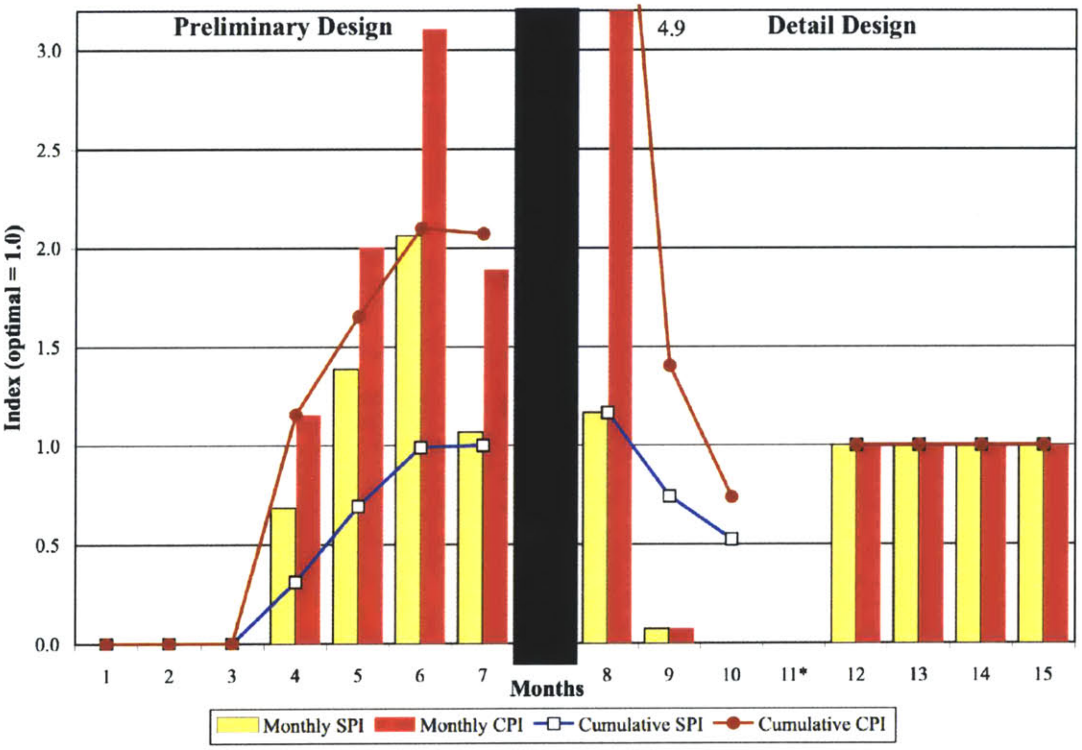


Figure 5.19 Controller Mechanical Engineering Effort Performance Indices



5.7.3.4 Software Development

The EVMS data for the controller software in Figures 5.20-5.22 indicates that the software (SW) preliminary design proceeded much like those of the rest of the controller work packages in that it started off behind schedule, and had to catch up in Months 5 and 6. Like HW2 and MEE, it ended under budget. It is interesting to note that the Month 3 BCWP was greater than 20% but no ACWP was claimed; however in Month 4, no BCWP was claimed, but the ACWP was about 20%. The detail design phase was behind schedule for Months 8-10, but the extrapolated Month 11 data indicates that they were very productive this month, and caught up to schedule. Just like the other work packages, Months 12-15 drove up the final budget (to 137% of the Month 10), as about 40% of the final budget was spent after Month 12, when the EV first reached 1.0.

Unlike the other controller work packages, the SW development continued after the re-baselining was complete, and continued with a new work package in Months 16-21. The schedule for this second detail design phase expected the group to achieve 68% of the schedule in the first month. While the group was behind after this month, they were able to catch up and be on budget by Month 18. However, in finishing the last 8% of the earned value, the group spent 83% of the final budget, resulting in the work package being 71% over budget.

Figure 5.20 Controller Software Earned Value

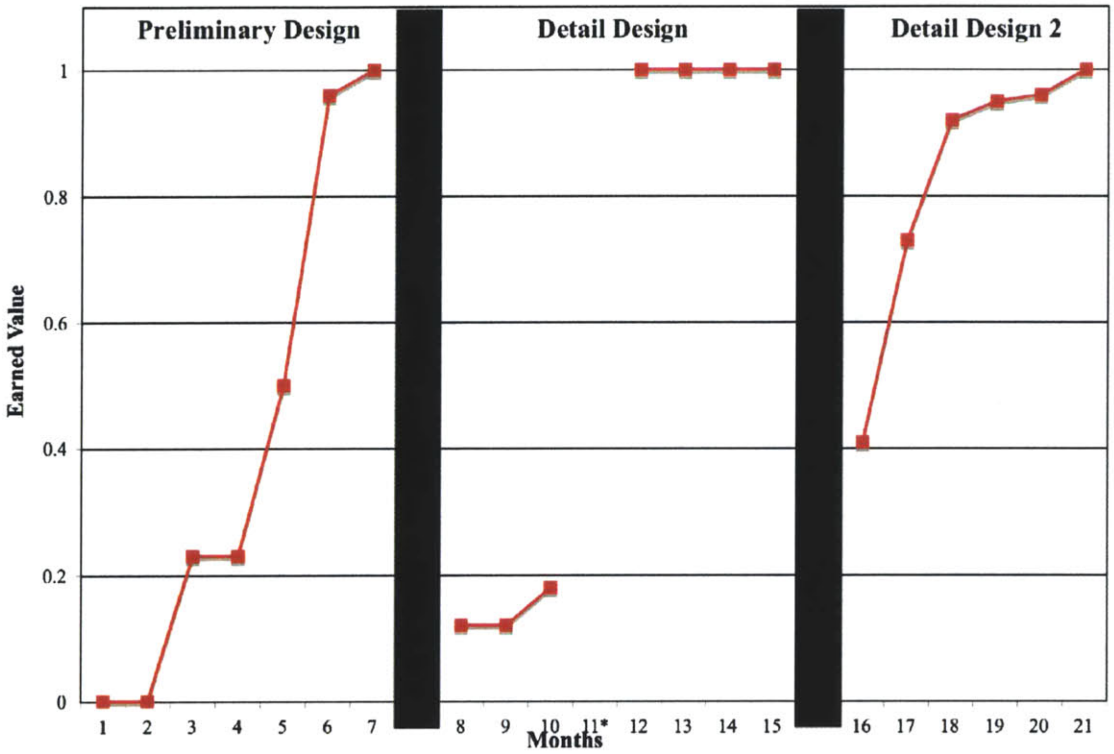


Figure 5.21 Controller Software Cost and Schedule Data

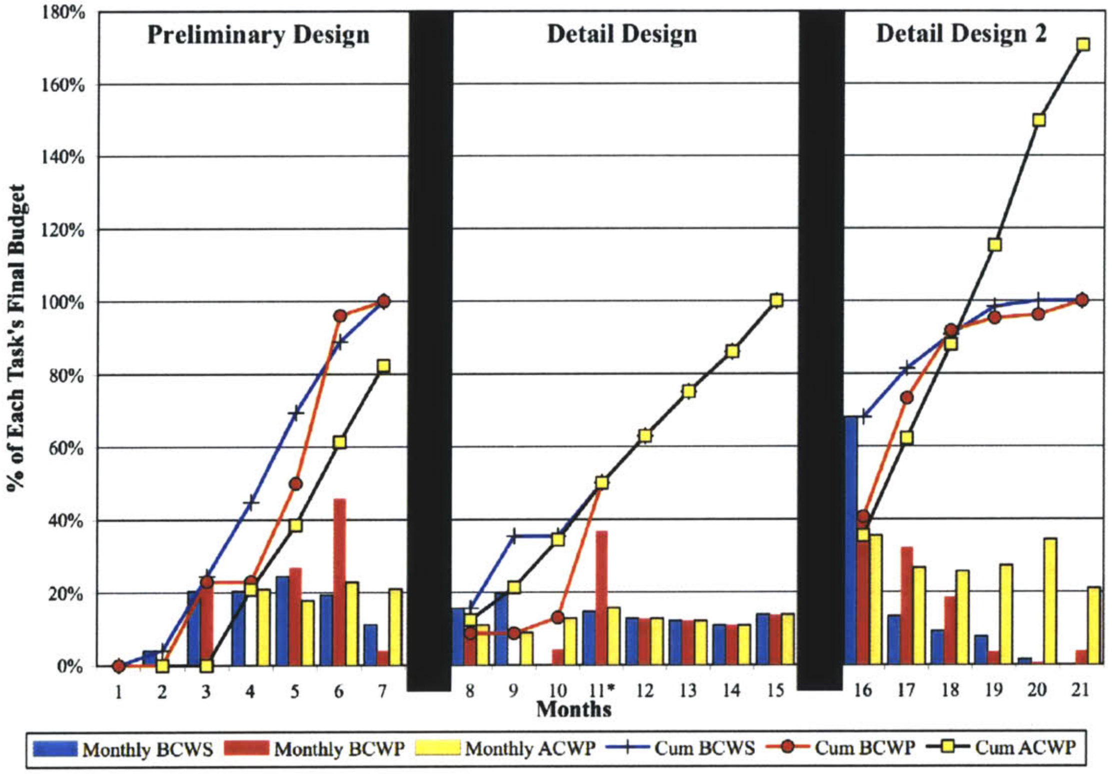
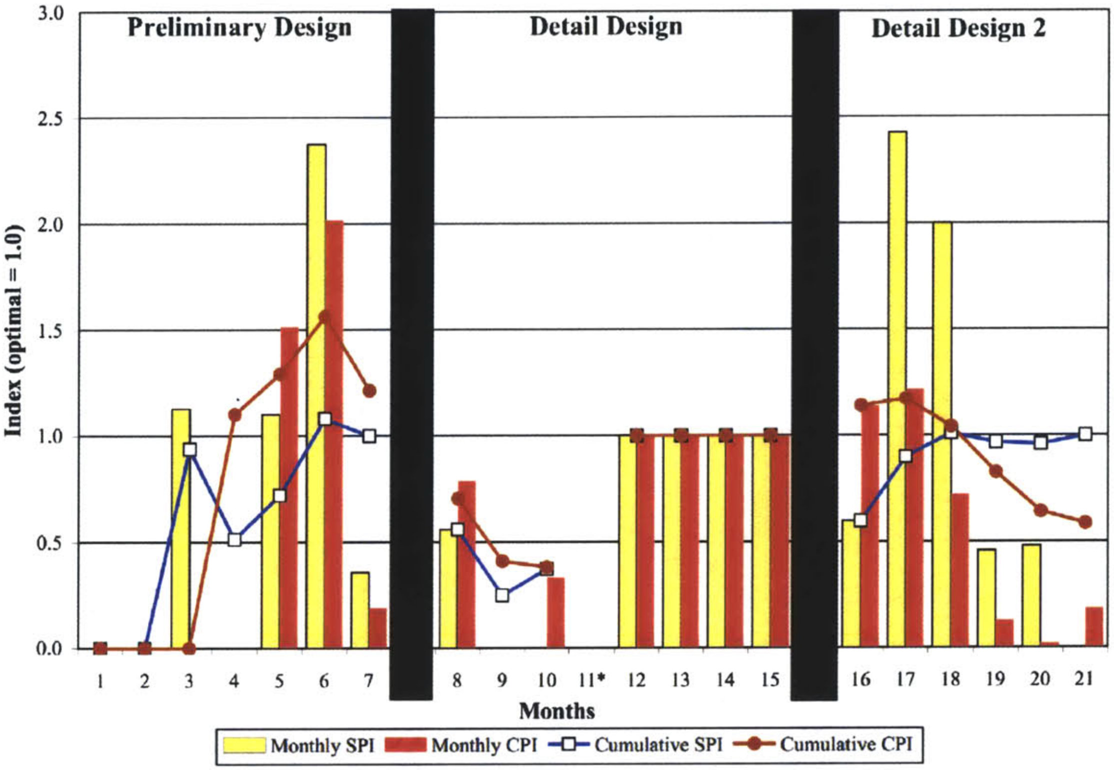
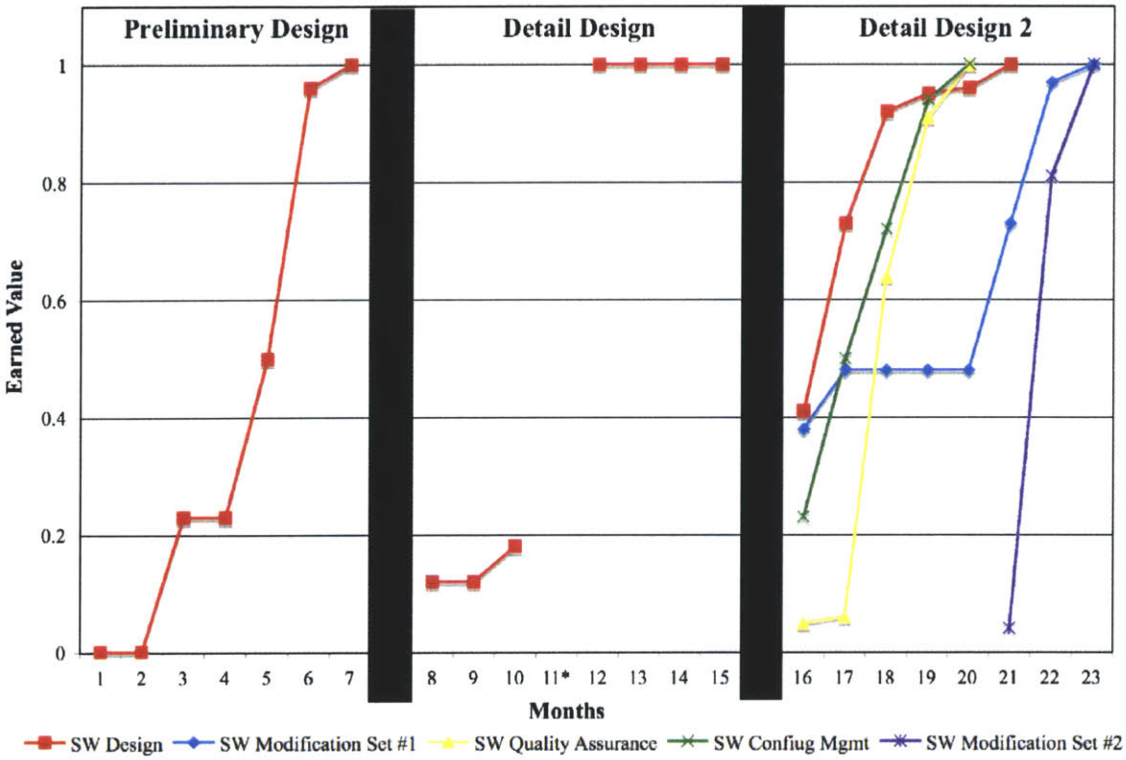


Figure 5.22 Controller Software Performance Indices



In addition to the preliminary and detail design work packages that all other parts of the controller development had, the SW development also had several additional work packages that dealt with quality assurance, configuration management, and two specific sets of modifications to the software. The earned value for these work packages is shown below in Figure 5.23 along with that of the preliminary and detail design work packages. This information offers a deeper look at the software development process. As is evident from the graph, the software Modification Set #1 and Quality Assurance tasks did not take very long once they were begun, but the Configuration Management task took five months of steady progress to complete. Modification Set #2 took the longest, and had a strange three-month time span in which no EV was claimed.

Figure 5.23 Earned Value for All Controller Software Work Packages



5.7.4 Controller EVMS & PDVSM Comparison

By creating both the PDVSM and associated value analysis, I have presented an outsider's opinion of the value of each of the tasks in Project R. From studying the EVMS data, one learns how Company X values the various phases of the development. Comparing the two sets of data illustrates how well the PDVSM corresponds to the actual process and how well the Earned Value Management System tracks the addition of value in the process. The EVMS data presented in the previous sections have been condensed, and just the earned value for each step is superimposed over the PDVSM for the process.

The controller is by far the most complex of the subassemblies studied, and thus this comparison is rather long and difficult. As was presented, the controller consisted of the major work packages associated with the preliminary and detail design of HW1, HW2, SW, and MEE. Figure 5.24 presents the PDVSM and EVMS comparison for the overall controller development, showing the EV for each of these four sets of work packages. Since the PDVSM does not specifically identify any task as either HW1 or HW2 development, and since the EV for these two developments is very similar for each month, the analysis will focus on "controller hardware" as a general term to encompass both. Additionally, it is known that the mechanical engineering group performed analysis tasks and created the detailed drawings needed for the hardware development, but there are no tasks on the value stream map that specifically call out this group. Thus, the comparison for this mechanical data is loose guesswork at best, and is included with the hardware development. While the software development is shown in this figure to illustrate its relation to the hardware development and mechanical effort, it is not analyzed in the same section as the hardware development. The SW preliminary and detail design are analyzed with the other SW work packages, presented together in Figure 5.25. For both the hardware and software comparison below, the EV for each month is analyzed in relation to the value of tasks worked upon that month (according to the PDVSM).

Figure 5.24 Project R Controller Work Packages EVMS and PDVSM Comparison

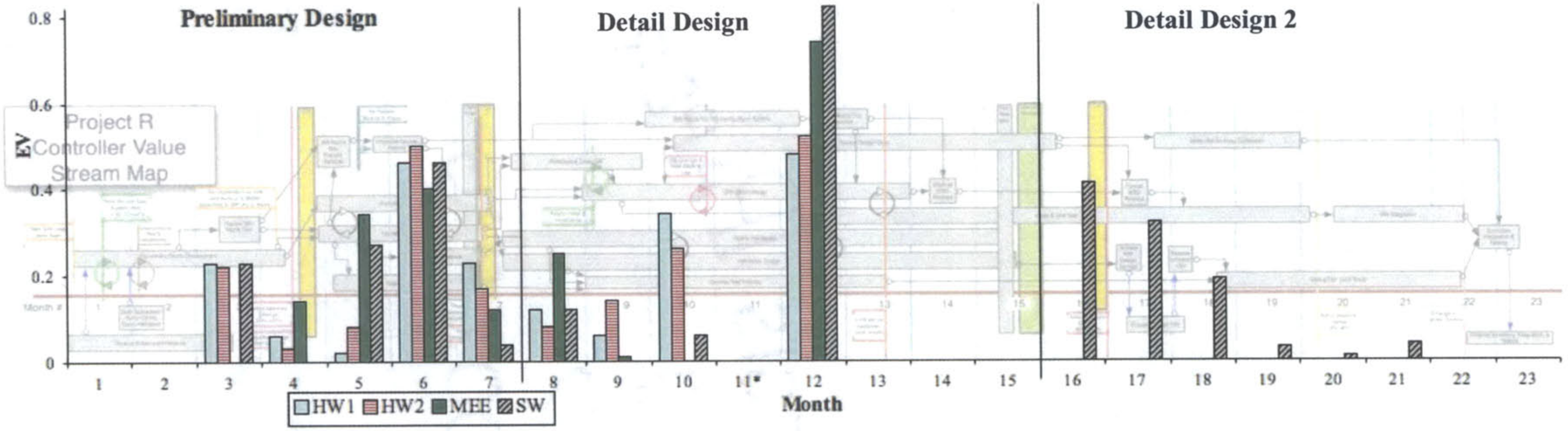
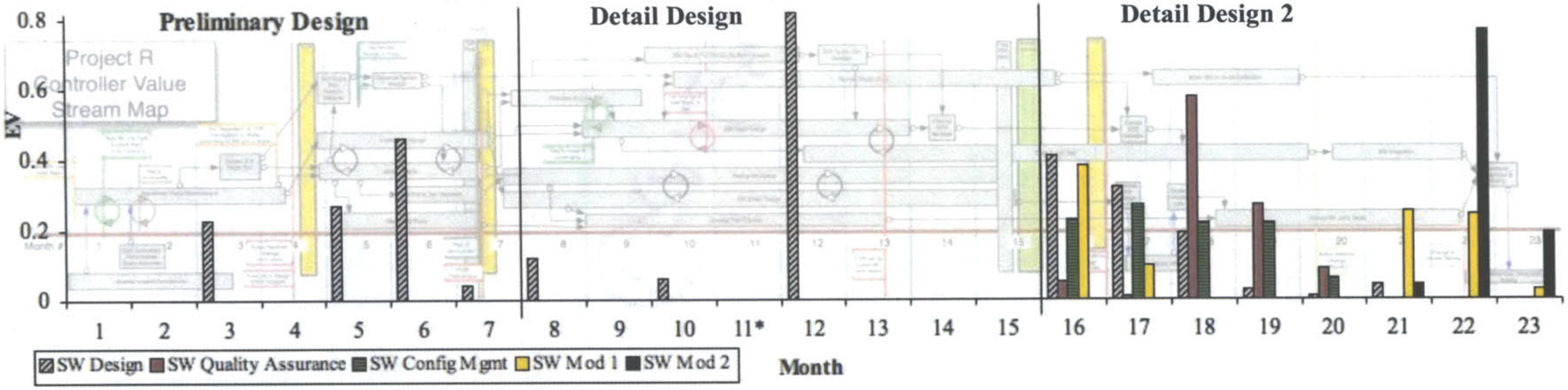


Figure 5.25 Project R Controller Software Work Packages EVMS and PDVSM Comparison



5.7.4.1 Controller Hardware Development Comparison

The story of the preliminary design for each of the hardware and mechanical work packages presented before is well illustrated in Figure 5.24. In Months 3 and 4, the requirements were being developed. Each of the HW work packages totaled about 25% EV for these two months, but there was no MEE EV claimed. Since creating the requirements frames the rest of the development and determines what the controller will be like, this 25% number for each HW development makes sense. It also makes sense that there is no mechanical EV in Month 3, but the 14% claimed in Month 4 is incongruent with the VSM. In Month 5, the preliminary design, requirements update, and test planning tasks began. There was very little EV claimed for the HW work packages, but there was a considerable amount (34%) for the MEE. The former seems to make sense, as during the beginning of the preliminary design, there is a great deal of uncertainty about the design, and thus it is hard to claim any EV. The mechanical EV claimed does not make much sense relative to the VSM since there are no tasks on the map that seem to indicate the work of this group. This could represent a shortcoming of the PDVSM. In Months 6 and 7, a great deal of EV was claimed, as the controller hardware development caught up to schedule. The HW tasks claimed an averaged of 48% EV in Month 6 and 20% in Month 7. This was when the tasks begun in Month 5 were finished, the HW requirements document was released, and PDR was held. During these months, the MEE claimed 40% and 12% earned value, respectively. The EV claimed in these months indicates that a lot of the milestones and "inchstones" were reached during Month 6 and that the preliminary design wrapped up in Month 7 as it should be according to the PDVSM. The tasks completed in these months helped to reduce the uncertainty in the product and improve its quality by creating the design and stabilizing the requirements; it makes sense that so much EV was claimed.

The detail design paints a cloudy picture of the value. Each of the HW tasks claimed about 20% EV total for Months 8 and 9; the mechanical group totaled 26% (almost all in Month 8). According to the PDVSM, the tasks begun during this period included the actual detail design, requirements updating, and test fixture development. While these are important tasks that contribute greatly to defining the product, they are not complete at this point. Thus it is no surprise that little EV was claimed, as was the case in the beginning of the preliminary design. In Month 10, the HW tasks produced a great deal, claiming an average 30% of the work packages' value, but there was no EV for the MEE work package. The tasks in this month were the same as in the previous ones, and one can only assume that at this point, the design and requirements had matured such that more milestones could be achieved. Under this assumption, the EVMS and PDVSM correspond for the HW tasks; in the case of the MEE, it corresponds for this month independent of the assumption. The EVMS data for Months 11-15 offer very little insight, as the re-baselining effort clouded when the value was actually added (it appears Month 12 was productive while Months 11, 13, 14, and 15 were a vacation for the engineers). It can be clearly seen, however, that during this time period much EV was claimed for each group as they finished the tasks begun in Months 8 and 9. The remaining value of these tasks was finally claimed, as the HW groups claimed an average of 50% work package EV in this period; the mechanical group, 76%! This amount of earned value is congruent with the value analysis base on the PDVSM.

Overall the controller hardware development EVMS data is congruent with the PDVSM developed, with a few exceptions. The PDVSM does not have any indication of how the MEE relates to the rest of the work packages. The EVMS data's lack of resolution in Months 11-15 that obfuscates the comparison. During the beginning of both design phases, the work packages were behind schedule, and very little EV was claimed. This fact seems to suggest that either the expectations of the group (BCWS) were excessive, the group underperformed (BCWS-wise), or (most likely) that the milestones do not adequately address the beginning of each design phase (they are mostly end-loaded).

5.7.4.2 Controller Software Development Comparison

The controller software development preliminary was similar to that of the hardware development: more than 20% EV was claimed in Month 3, much less in Month 4, a ramp-up in Months 5 and 6 to catch up to the schedule, and much less in Month 7. The Month 3 EV makes sense for the same reason that the hardware development did: the initial requirements development is crucial for shaping the final product, but does not remove a considerable amount of uncertainty. The lack of Month 4 EV correlates well with the controller PDVSM, in which the only task for that time was the execution of the ADR. According to the controller software group leader, when reviewing the software requirements document, it became apparent that the software requirements were heavily dependent upon the hardware architecture, and that the document produced in Month 3 could not be considered to have much value until the hardware architecture was settled. Thus, perhaps Month 4 was spent waiting for the hardware architecture to be established while creating a draft version of the requirements document. During Month 5, this unstable requirements document was released while requirements were being updated, and preliminary design began. In order for the software requirements to be officially released, it needed to be entered in to the configuration management system, which begins with a review board of various high-level engineers and managers. A board should have been in place before, but it was during this time that the group leader had to scramble to assemble such a review board. The group claimed 27% earned value this month (much more than either hardware group but less than the mechanical group), which seems about right for the number of valuable tasks performed. Like all the other controller work packages, Month 6 was when the most software EV was claimed as the group caught up to schedule. In Month 7, the EV tailed off (from 46% to 4%) as the remaining preliminary design and requirements updates were finished in preparation for the PDR.

The software detail design started off behind schedule, and the group did not achieve much until the confusing re-baselining period. In Month 8, as the software was being prototyped and the detail design begun, 12% EV was claimed. During Month 9, no EV was claimed, and this could be a result of having to perform rework due to requirements creep and uncertainty. In Month 10, the group claimed 6% EV as the requirements were finally entered into the configuration management system, the detail design was being performed, and its results were being reviewed. This small amount of earned value seems to be another case of the lack of early design milestones. In Months 11-15, a whopping 82% of the earned value for the software detail design was claimed. During this period, the detail design was performed and reviewed, the requirements made it into the configuration management system and were officially released, an informal design document release was made, and the CDR preparatory meetings were held. Each of these tasks adds considerable value to the design process, and it makes sense that such a considerable portion of the EV was earned in this period. If this period had better EV resolution, it would have enabled better comparison with the PDVSM since the software portion had good resolution.

The software development was not finished by the end of the re-baselining effort, and a second detail design work package was created. Most of the EV this work package was claimed in Months 16-18, and with some being claimed until Month 21. This fact makes sense, as Months 16-19 were when the software code and unit test was being performed and the Software Design Document (SDD) was formally released.

Also during this period after the re-baselining, four other smaller work packages related to the software development were completed, as is shown in Figure 5.23. Configuration management began in Month 16 and continued at a relatively steady pace through Month 21. There is no reflection of this work package on the PDVSM. In fact, the only reference to configuration management was during the first detail design period; obviously, these are not one and the same. A work package for quality assurance had minimal earned value claimed on it in Months 16 and 17, but had an incredible amount during Months 18 and 19 (with a minor amount in Month 20). While there is no PDVSM task associated with this work package, its spike in earned value coincides with the tailing off of the second software detail design work package and the last part of the "code and unit test" task. This fact suggests that the later part of the "code and unit test" task, testing was performed for quality assurance purposes. Also, there were two sets of software modifications performed in this period. Software Modification Set A dealt with the controller calibration effort. Strangely enough, this task reported about half of its EV in Months 16 and 17 but then claimed no more EV until Months 21-23. This fact suggests that something is amiss, as one would expect that the task would not stop and then continue after a three-month hiatus. Also this gap in the EV disagrees with the PDVSM, which has a task named "Write SW for Array Calibration" which occurred mostly during Months 18 and 19. Finally, Modification Set B was completed in Months 21-23, with the majority of the work being performed in Month 22. This work package is not reflected in the PDVSM.

The software development effort was similar to the hardware development for the preliminary and detail design phases, in that each phase started behind schedule and ended with a spike in effort at the end. Also, the comparison was invalidated during the re-baselining period. However, the software development was different in that it had a second detail design phase and several other work packages. Of these, only the quality assurance and the second software detail design packages are reflected in the PDVSM.

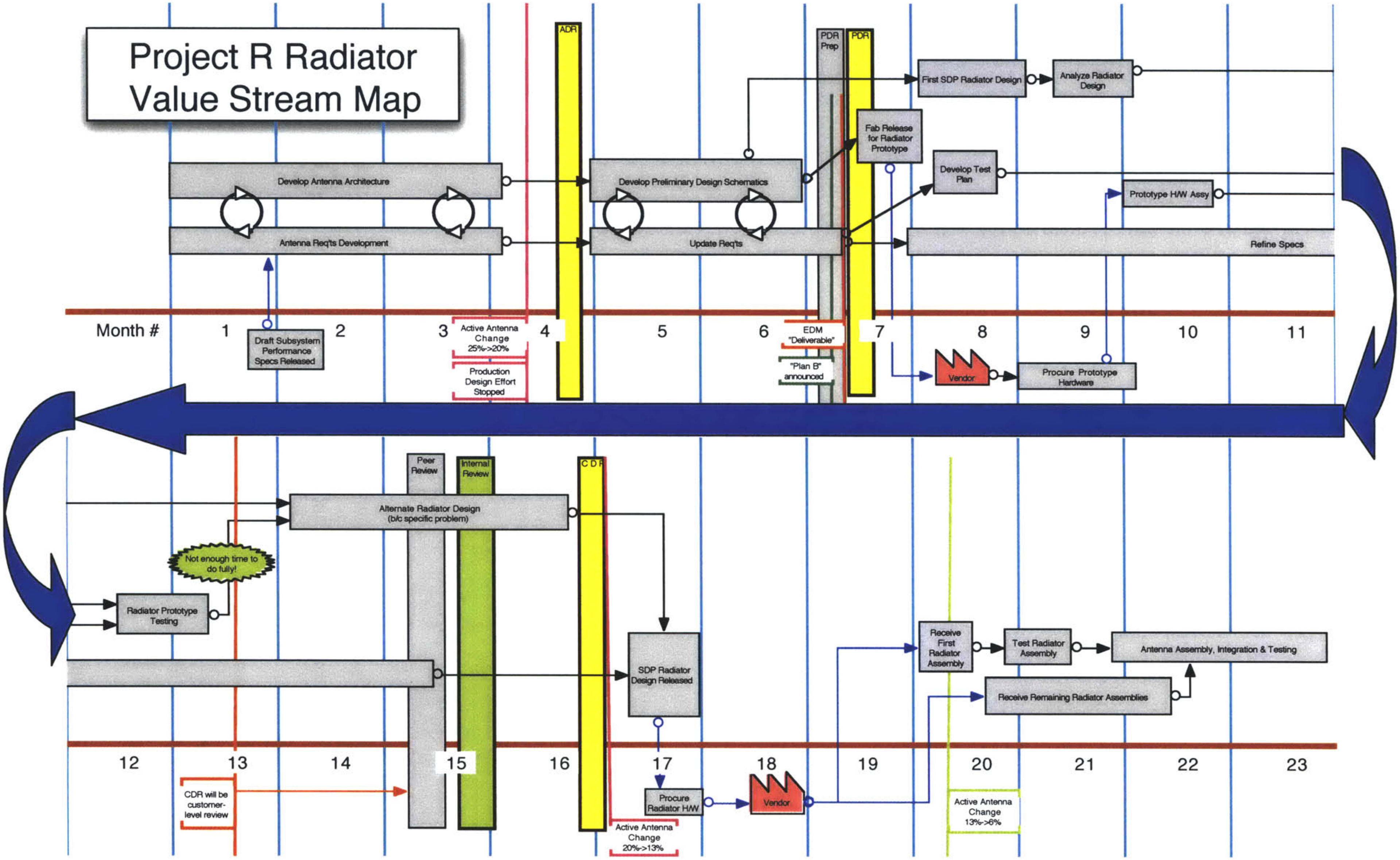
5.8 Radiator Development

The radiator is an integral part of the antenna design. Thousands of radiators cover the antenna's surface and allow it to properly interact with electromagnetic radiation so that it can perform its intended function. A considerable amount of electronics sits between the radiators and the controller that "steers" the antenna by activating certain sets of them. The design of the radiator is crucial to the performance of the antenna, and was performed by the A/RF electronics group.

5.8.1 Radiator VSM

Shown below in Figure 5.26 is the value stream map for the radiator development effort. It shows how the development of the antenna architecture and requirements preceded the ADR, and that subsequently the requirements were updated and the preliminary design schematics were developed. After the PDR, information was released to the vendor for creating a radiator prototype based upon these preliminary schematics. During the prototype build, the radiator design was more fully fleshed out and then analyzed. Once the prototype was assembled and partially tested, it became obvious that there was a specific problem with the radiator design; consequently, a new design was developed before the CDR. After the CDR, this new design was released for vendor fabrication. As the vendor sent back the first of the assemblies that contain the radiators, it was tested. Once all radiator assemblies were tested, they were integrated to make Antenna N.

Figure 5.26 Project R Radiator Value Stream Map



5.8.2 Radiator Value Analysis

Table 5.4 Radiator Value Analysis

Task	Product/ Process	Time	Cost	Quality
Draft Subsystem Performance Specs Released	Process & Product	Sets goals; Determines the effort req'd => Amount of time and budget red's		Establishes quality demanded
Antenna Req'ts Development	Process & Product	Sets goals; Determines the effort req'd => Amount of time and budget red's		Determines what product will be like
Develop Antenna Architecture	Process & Product	Sets goals; Determines the effort req'd => Amount of time and budget red's		Determines what product will be like
Advanced Design Review	Process & Product	Check Status		Checks quality of product and process
Develop Prelim Design Schematics	Product			Begins creating product
Update Reqt's	Product			Refine what product will be
PDR Prep	Process			Makes PDR smoother
Prelim Design Review	Process & Product	Checks status		Check quality of design
Fab Release for Radiator Prototype	Process	Enables prototype fabrication		
First SDP Radiator Design	Product			Creates product
Analyze Radiator Design	Product			Checks product quality
Develop Test Plan	Process	Determines test cost and schedule		Figures out how to proceed;
Refine Specs	Product			Further defines what product will be
Procure Prototype Hardware	Process	Enable testing to begin		
Prototype HW Assembly	Process	Enable testing to begin		
Radiator Prototype Testing	Product	Learn if redesign necessary→ Whether or will be costly and require more time		Checks quality of radiator assembly architecturalconcept
Alternate Radiator Design	Product			Creates new product to avoid problems of the original design
Peer Review	Process			Check quality; Prepare for CDR
Internal Review	Process			Check quality; Prepare for CDR
Critical Design Review	Process	Checks status		Gets customer feedback; Checks quality, status, & direction
SDP Radiator Design Released	Process & Product	Enables fabrication		Fully defines the product
Procure Radiator HW	Process	Enables assembly		

Task	Product/ Process	Time	Cost	Quality
Receive First Radiator Assembly	Process	Enables unit testing & eventual antenna assembly, integration & test		
Test Radiator Assembly	Product			Checks product quality
Receive Remaining Radiator Assemblies	Process	Enables unit testing & eventual antenna assembly, integration & test		
Antenna Assembly, Integration, & Test	Product			Assembles final product; Check its quality; Final proof of quality

5.8.3 Radiator EVMS Data

Similar to the controller SW development, the radiator detail design was not complete when the re-baselining took full effect after Month 15, and thus had a second detail design work package. The radiator EVMS data in Figures 5.27-5.29 combine to paint a bleak picture in which the budget was exceeded by all three of the design phases. The radiator preliminary design was similar to that of HW1 and other work packages in that the team did not accomplish much at first, and had to be highly productive in the final months to catch up. In Figure 5.28, one can see that Months 5-7 were very expensive, and drove the work package to almost 50% over budget. The Month 3 CPI spike resulted because little money was spent, but much EV was claimed. Also, the SPI spikes in Months 6 and 7 are due to the team catching up after being behind. The first three months of the detail design phase, the team achieved as much as planned, but spent too much doing so. The data from Months 12-15 were similar to all the others, only indicating that more than 50% of the final budget was spent after Month 12, when the EV claimed was 1.0. The final budget for the detail design was more than twice as much as the budget from Month 12, which itself was three times as much as the original Month 8 budget. Thus, this phase of the design was probably over budget. The second detail design phase was front-loaded and also ran 40% over budget. About 80% of the EV was claimed in the first month of this work package (at a reasonable cost), but the final 20%

cost 55% of the final budget and sent the work package to 40% over the final budget. In all, the radiator development was far over budget.

Figure 5.27 Radiator Earned Value

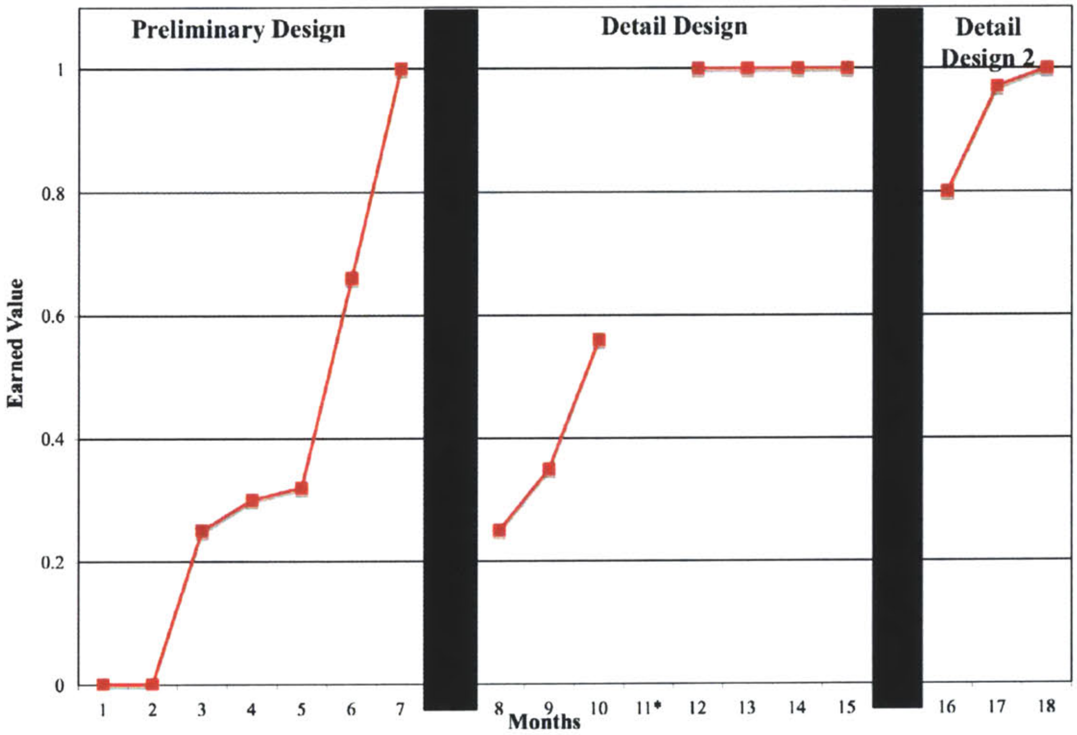


Figure 5.28 Radiator Cost and Schedule Data

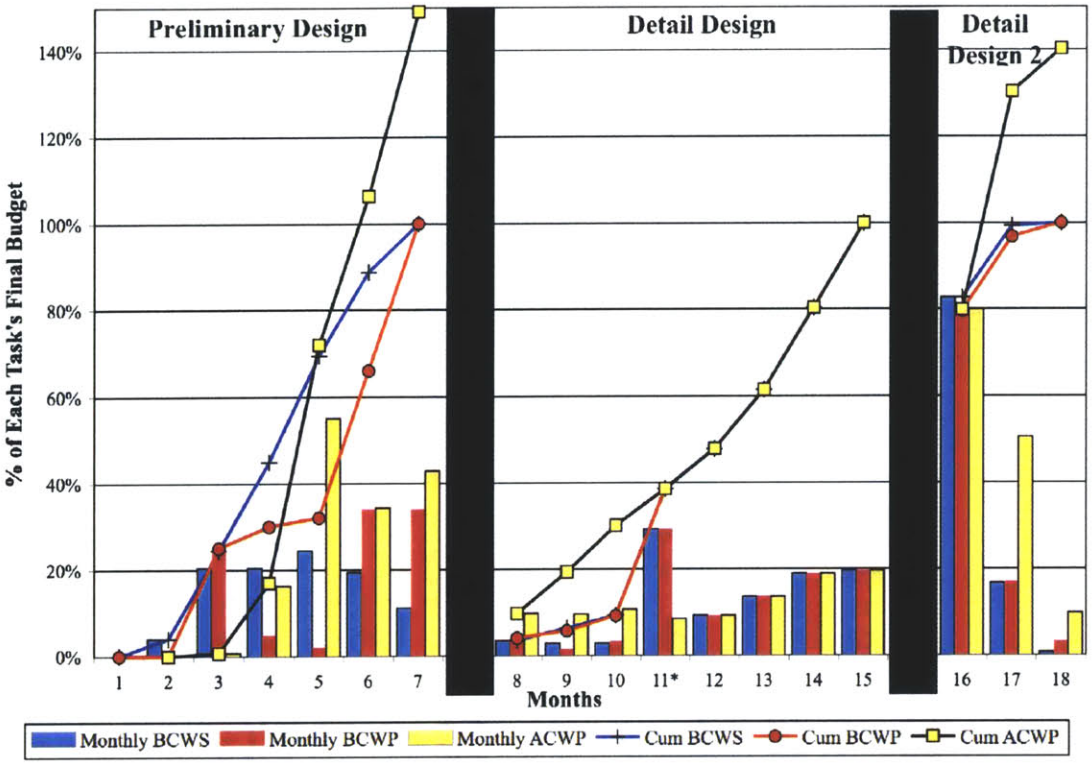
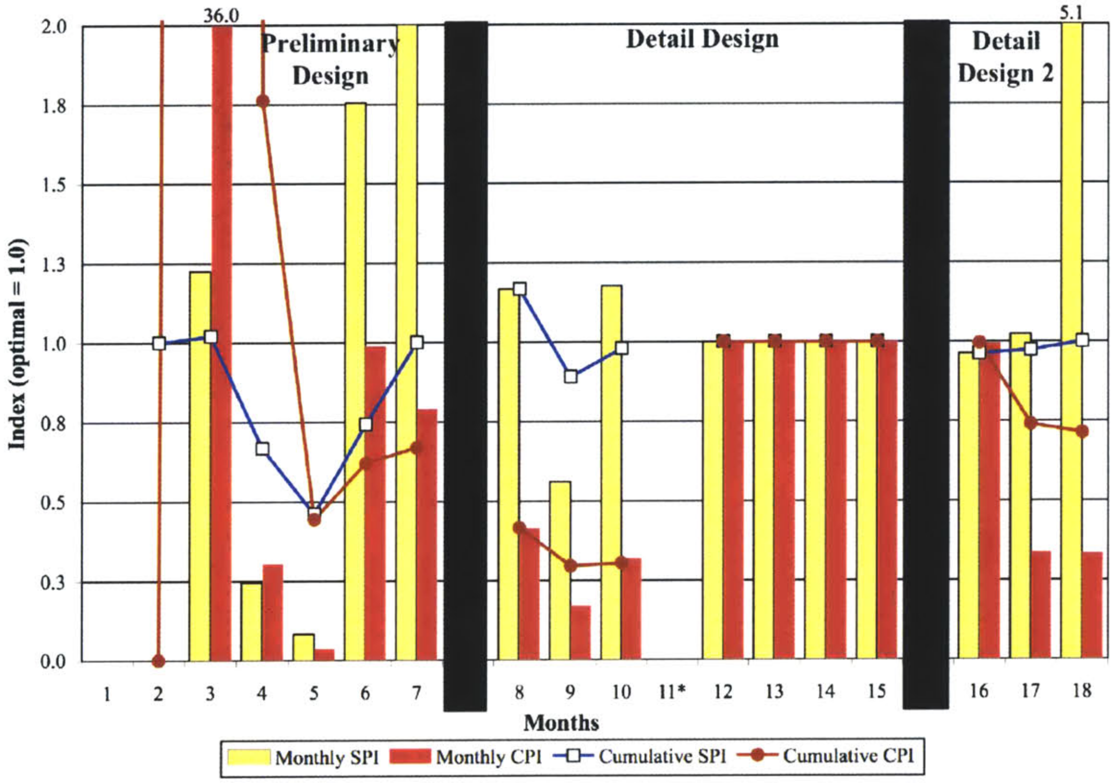


Figure 5.29 Radiator Performance Indices

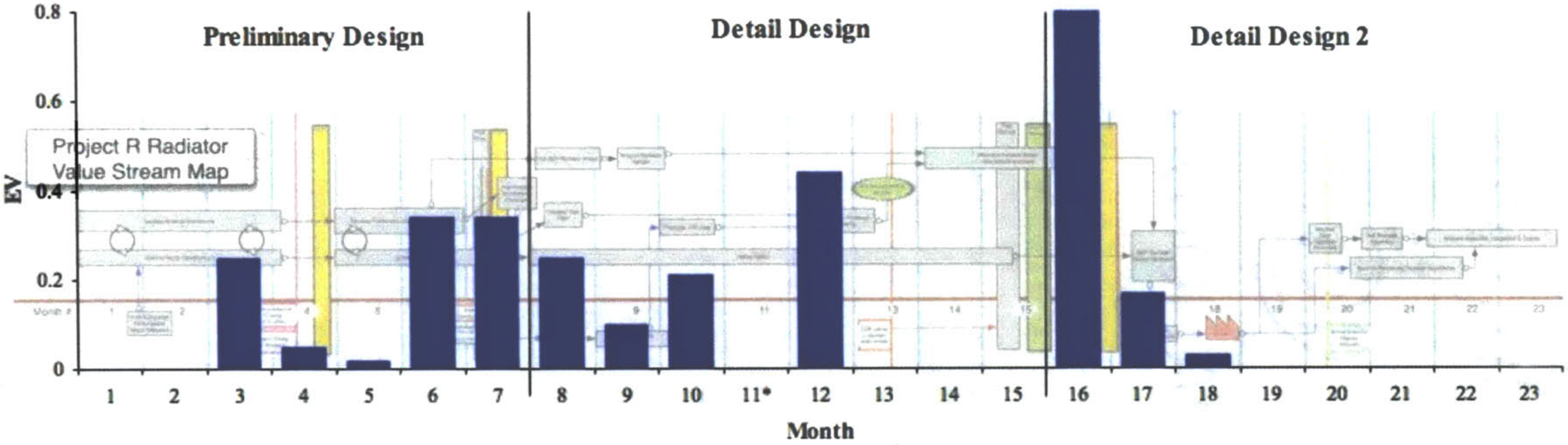


5.8.4 Radiator EVMS & PDVSM Comparison

The radiator EVMS data is easily comparable to the PDVSM. During the preliminary design, the team fell behind and had to catch up during Months 6 and 7 as the requirements were updated, the preliminary schematics were completed, and the PDR was held. The EV claimed during the first month of the preliminary design aligns with the development of the antenna architecture and the initial antenna requirements development. While these tasks were not exclusively related to the radiator, the radiator is a considerable component of the antenna; thus, the 25% EV claimed makes sense. In the detail design, the 25% claimed the first month coincides with several tasks: the radiator prototype design release for fabrication, the first SDP radiator design, the test plan development, and the refinement of the requirements. All these tasks have a strong value component, so it make sense that so much EV should be claimed. Based on the pattern of the controller, it is likely that little EV was claimed relative to the requirements updating; much of the 25% from Month 8 is likely related to the radiator design and the test plan development. In Months 9 and 10, the radiator design was analyzed, the requirements update was continued, and the prototype was procured and assembled. During this time, nearly 31% of the work package EV was claimed. This fact makes sense, as the prototype build and analysis contribute to the reduction of uncertainty in the product quality.

As was the case with the controller, the re-baselining period hides when EV was claimed in the later portion of the detail design. Only 44% of the work package's EV was claimed in this time period in which a relatively large number of tasks were completed: the requirements were settled, the radiator prototype testing executed, the alternate radiator design was begun, and the CDR preparations done. The PDVSM and value analysis indicate that more value should have been added, and therefore disagree with the EVMS data. Like the controller software, the radiator also had a second detail design phase. A whopping 80% of the EV was claimed in Month 16, with Months 17 and 18 combining for the remainder. The incredible amount of EV claimed in Month 16 is logical, as it was during this time period that the alternate radiator design was completed and the CDR was held. The 15% EV claimed in Month 17 can be attributed to the release of the radiator design for fabrication and the procurement of the radiator (which continued into Month 18). Overall, the radiator design EVMS matches the PDVSM, but has some points of disagreement.

Figure 5.30 Project R Radiator EVMS and PDVSM Comparison



5.9 Structure Development

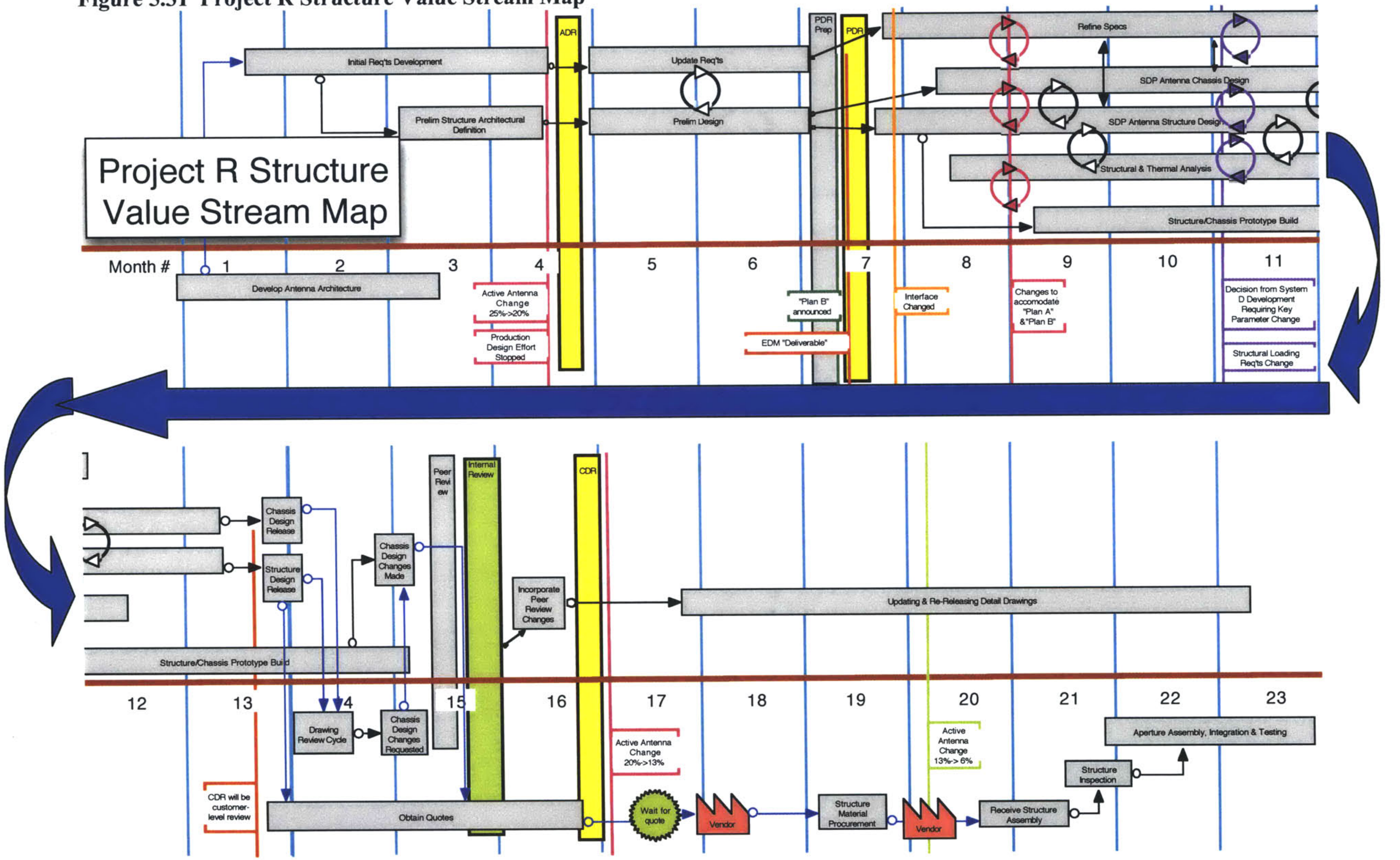
The structure is the physical housing for all the electronics, radiators, controller, and power system for Antenna N. For Project R, the overall structure developed was the size of a full antenna, but the SDP version of Antenna N only had 25% (then 20%, then 13%, then 6%) of its surface covered with active radiators. A smaller structure had to be designed that could hold the active radiators and associated electronics within the much larger overall structure. This smaller structure, known as the chassis, caused several problems for the development effort. Reviewers determined that the chassis design had to be changed before it could be released for fabrication. Additionally, the structure itself had to be obtained from a vendor. Before it was ordered, the overall structure had to be changed to fit either the Plan A or the Plan B design. Because the structure is such a large item, its procurement had a long lead time, and in between its order and receipt by Company X, requirements changed, but the structure could not be changed. This led to a discrepancy between the chassis and structure, and thus the chassis design was adjusted.

5.9.1 Structure VSM

Figure 5.31 illustrates the value stream map for the Project R structure development. After the antenna architecture was established, the requirements and architecture for the structure were created. Between ADR and PDR, the requirements were updated and the architecture was turned into a preliminary design. After PDR, several major changes created problems for the structure development process. First, when the "deliverable" status was determined, it required that drawings and documents be far more detailed than they were and that everything be very well documented. At the PDR, it was determined that the chassis was needed for the structure to properly interact with the rest of the SDP testing. This requirements change led simultaneous execution of the chassis and structure design, requirements update, and analyses. Since using the chassis required special connections and since it had not been part of the original design, a prototype for the chassis was created between PDR and CDR. Additionally, the invocation of Plan B led to design changes for the structure. Instead of creating a separate design just for Plan B, the original design was changed to accept either the Plan A or Plan B system architecture. Further along during the detail design, two major changes to the requirements of the System D design caused a lot of rework in the Antenna N structural design. These two changes dictated that a key parameter in the structure's design must be changed and at the same time that the structure would have to bear a great deal more of the load of the electronics than had been previously planned. Not only did these changes cause a great deal of rework for the structure and chassis design, but also limited the value of the previous chassis prototype (which had been designed to the earlier requirements).

After the structure and chassis design were considered complete, they were released to a management review cycle. The overall structure design was approved and entered into the procurement process nearly immediately, which was beneficial since the large and complex structure required a lengthy time period for vendor delivery. While the structure made it through the review cycle, the reviewers demanded that changes be made to the chassis design. The changes were made, and the design was approved for procurement release before the Peer Review and Internal Review. During the Internal Review, several issues were discovered that required the design information be updated before the customer CDR. About seven months after the structure design was released, Company X received and inspected the structure assembly. The structure was subsequently assembled with the other hardware items, and the full Antenna N was integrated and tested. During this time, the mechanical engineering group spent several months updating and re-releasing drawings so that they would perfectly match what was actually delivered for Antenna N. This drawing update effort was required because of the "deliverable" status decision.

Figure 5.31 Project R Structure Value Stream Map



5.9.2 Structure Value Analysis

Table 5.5 Structure Value Analysis

Task	Product/ Process	Time	Cost	Quality
Develop Antenna Architecture	Product	Determines work effort needed		Dictates antenna performance & general structure characteristics
Initial Req'ts Development	Product			Dictates what structure must do
Prelim Structure Architectural Definition	Product			Dictates structure format
Advanced Design Review	Process & Product	Checks status; Determines Structure & Chassis Needed =>more effort		Checks status; Determines structure & chassis Needed
Update Req'ts	Product			Refine what structure must do; Defines what chassis must do
Prelim Design	Product			Creates Initial Design
PDR Prep	Process			Makes PDR smoother
Prelim Design Review	Process & Product	Checks schedule Performance	Checks cost Performance	Checks Quality
Refine Specs	Product			Improves knowledge of what structure & chassis must do
SDP Antenna Chassis Design	Product			Defines the chassis
SDP Antenna Structure Design	Product			Defines the structure
Structural & Thermal Analysis	Product			Checks the quality of the design
Structure/Chassis Prototype Build	Product			Checks feasibility of the structure/ chassis concept
Chassis Design Release	Process			Allows for quality check
Structure Design Release	Process			Allows for quality check
Drawing Review Cycle	Product			Checks quality of design
Chassis Design Changes Requested	Product			Dictates how to improve design quality
Chassis Design Changes Made	Product			Improves design quality
Obtain Quotes	Process	Determines lead time for materials	Determines cost for materials	
Peer Review	Process			Prepares for the internal review
Internal Review	Process			Prepares for CDR
Incorporate Peer Review Changes	Process			Improve design and presentation

Task	Product/ Process	Time	Cost	Quality
Critical Design Review	Process	Checks status		Gets customer feedback; Checks quality, status, & direction
Updating & Re-Releasing Detail Drawings	Process			Done because of "deliverable" status; Did not have much effect on the design; Created better drawings for production antenna design to start from
Structure Material Procurement	Process	Determines when materials can be assembled	Determines how much materials will cost	
Receive Structure Assembly	Process	Allows inspection & assembly to begin		
Structure Inspection	Product			Checks the quality of the parts
Antenna Assembly, Integration, & Test	Product			Creates Antenna N; Tests Performance of Antenna N; Prepares for Integration with SDP I

5.9.3 Structure EVMS Data

Shown in Figures 5.32-5.34, the Structure EVMS data paints the structure development in a light similar to all the other component developments. The development was behind schedule for the first four months of preliminary design, but caught up in Months 5-7. Due to modest expenditures and worthwhile productivity in Months 3 and 4, deceptively high CPI values were reported. In Months 6 and 7, however, the ACWP was very high and drove the work package over 14% budget. During the first three months of detail design, the team was nearly on schedule, and was actually under budget. During the re-baselining period, 36% of the overall budget was spent after the detail design EV was claimed as 1.0, resulting in the final budget being 148% of the Month 10 detail design budget. Thus, the detail design phase was similar to the preliminary design phase in that they both went over budget.

Figure 5.32 Structure Earned Value

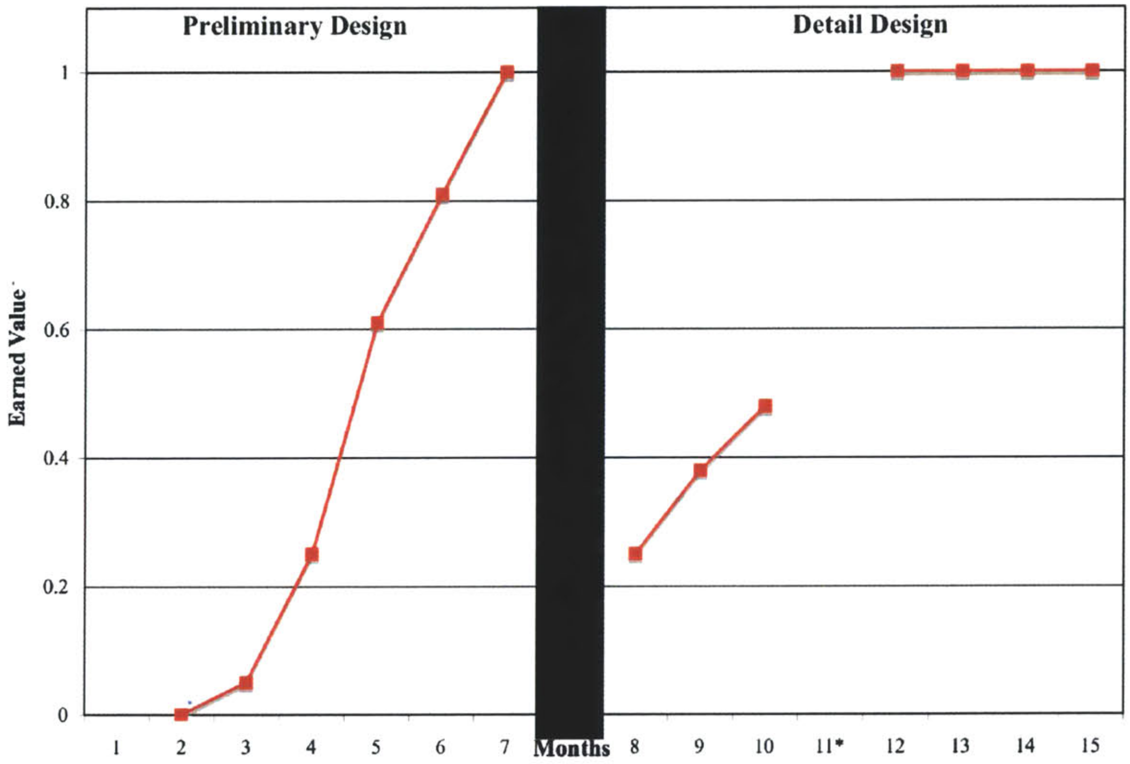


Figure 5.33 Structure Cost and Schedule Data

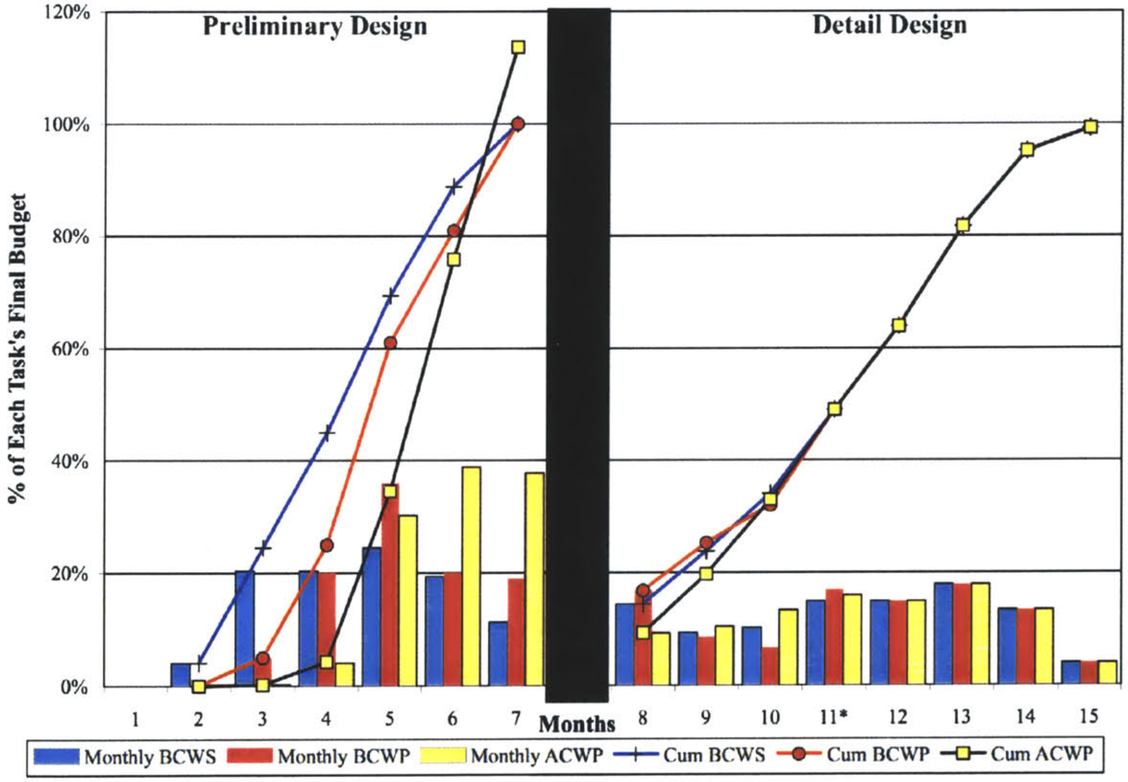
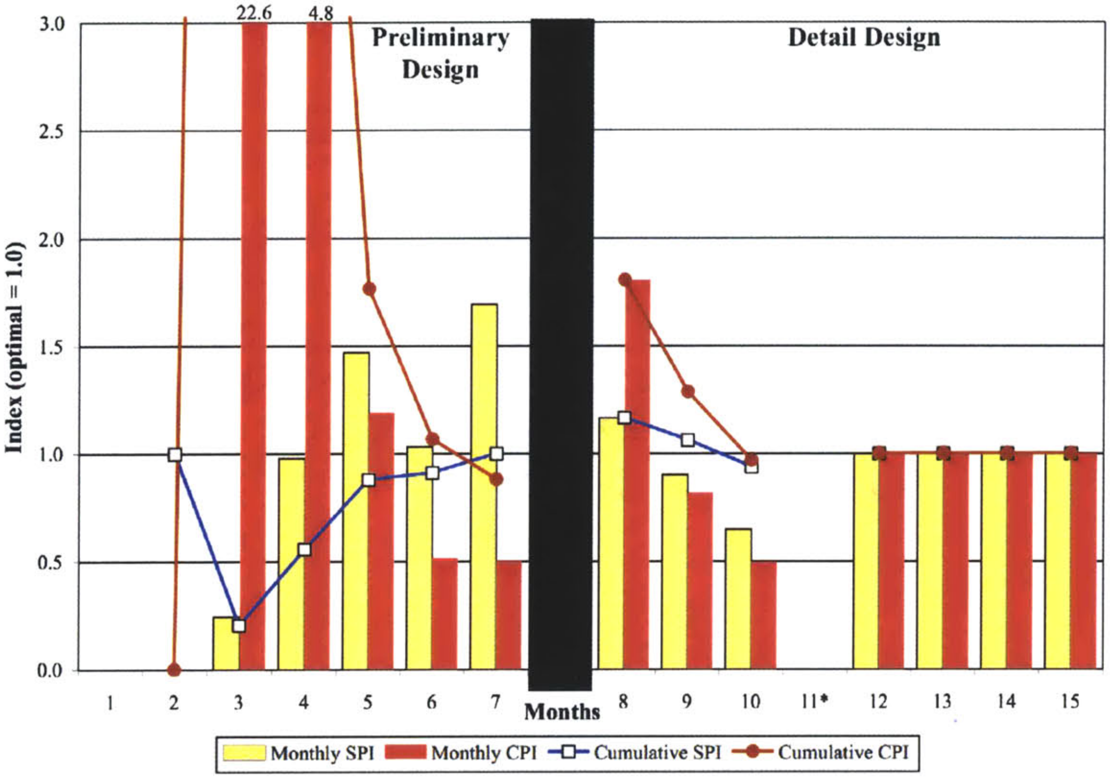


Figure 5.34 Structure Performance Indices

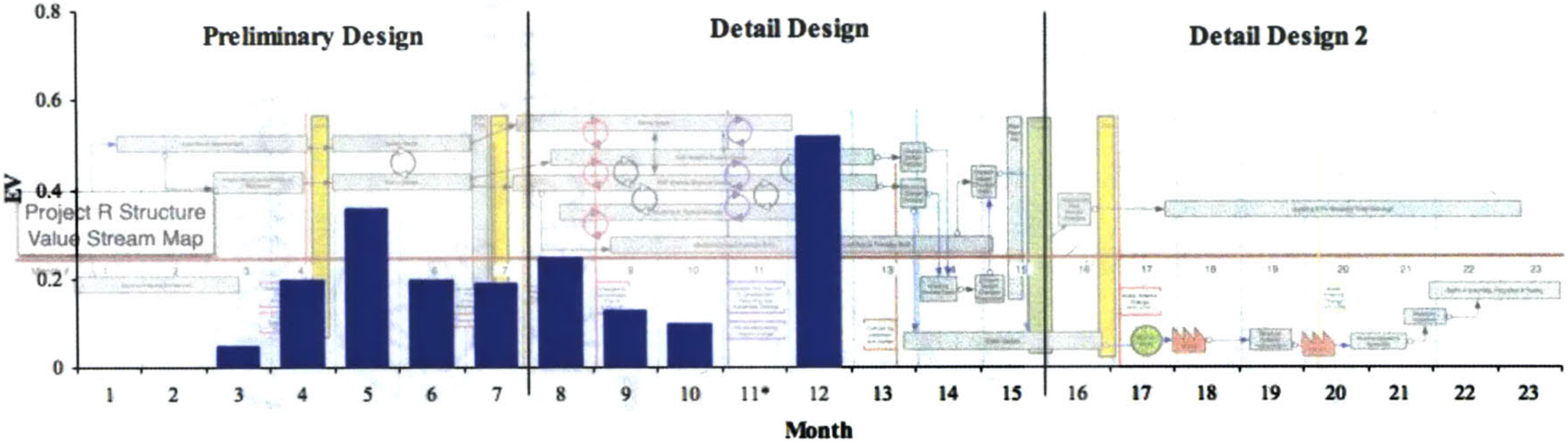


5.9.4 Structure EVMS & PDVSM Comparison

The structure EVMS data and PDVSM align well as is shown below in Figure as each of the monthly EV claimed makes sense with respect to the value analysis. It stands to reason that little EV was claimed in Month 3, as the structural architectural definition was just beginning. The initial requirements development and architecture were complete by the ADR in Month 4, justifying the 20% EV that month. 36% of the work package EV was claimed in Month 5, at the beginning of the preliminary design and requirements update. Months 6 and 7 averaged 19% EV claimed as the preliminary design was wrapped up and PDR held.

The structure differs from the other subassemblies in that a great deal of EV was claimed during Month 8. This suggests either the group was highly productive this month or the EV milestones were better aligned with the tasks being performed than they were for the other subassemblies. During this month, the requirements were updated, the chassis and structure design were begun, and the structural and thermal analyses began. Also during Month 8, changes to the interface between Antenna N and the rest of the SDP had to be dealt with. The EV claimed this month seems to be excessive for the work seen on the PDVSM. The EV from Months 9 and 10 totaled 23% as the tasks from Month 8 were continued and better defined. The EV for each of these months makes more sense than does that of Month 8. The structure is no different than the rest of the Antenna N development in that it faced the obfuscation of the earned value data by the re-baselining during Months 11-16. Looking at all the tasks completed during this period on the PDVSM, the 52% EV claimed during this time is less than what the PDVSM and value analysis suggest. The completion of all the tasks from Month 8, coupled with all the other tasks, should add more value. Overall, the structure development followed a different pattern than the other subassemblies in that it was more front-loaded, but unfortunately suffered the same confusion due to the re-baselining.

Figure 5.35 Project R Structure EVMS and PDVSM Comparison



5.10 Project R Conclusions

Project R was a complex hardware prototype development effort, and the complex system context that it operated within created issues that handicapped the effectiveness of the project. The requirements uncertainty haunted the entire development effort, causing each group on the team to keep working on requirements long after they should have been settled. The uncertainty associated with the extended period of project re-baselining (resulting from an increase in the product's scope) affected the engineer's efforts and hampered the team's use of the Earned Value Management System. Also due to outside influences, the project funding was reallocated before its product, the prototype antenna, could be properly tested within the SDP context.

This complex process was represented by a large Product Development Value Stream Map, which enables one to see the flow of information between tasks. In order to qualitatively analyze how value was created in Project R, this large intricate map was both abstracted into a simpler high-level map and also broken down into detailed subassembly map sections. The tasks from each of these maps were analyzed with the Oehmen value framework, and the EVMS data for each subassembly was collected and studied. The EVMS data was then compared to the PDVSM for each of the subassembly maps. These comparisons revealed some issues general to all of the subassemblies, and some specific to each one. It was obvious that comparing the EVMS data to the PDVSM is not only *feasible* but also *valuable*, for it helps improve understanding of the project and offers a useful means for analyzing how value was added during the project. However, this comparison does not tell the whole story, since the EVMS data is of limited use when scope changes occur, and the PDVSM (in this case) is somewhat uncertain due to its construction after the design effort, rather than concurrent with it.

All of the subassemblies tended to follow the same pattern within each of their respective work package phases: they fell behind, and then caught up to schedule near the end of the phase (often overspending to do so). This fact could be due to many factors, but it is suggestive of external influences common to the project and design team as a whole. This is supported by the global impacts of the higher level architecture changes, delayed requirements, excessive up-front expectations, and uncertainty over the "deliverable" status of the antenna. Another consideration is that for discrete milestones, earned value is often achieved and claimed in a back-loaded manner. That is, a lot of work is done up front to support the design before any tangible milestones can be claimed, but this groundwork allows the milestones to be claimed rapidly near the end of the task. By contrast, the engineering team spends at a constant rate, so the ACWP data is essentially linear over time. This results in a schedule variance at the beginning of a task, and a correction at the end. Also, the re-baselining effort created confusion and clouded the EVMS data for all of the subassemblies. As mentioned above, this turbidity of the EVMS data severely limited the usefulness of the data, and considerably hindered comparison with the PDVSM, over what was a particularly value-adding portion of the project.

While most of the issues that affected Project R were out of the hands of the Project R team, perhaps the team could work to identify better up-front milestones that reflect how the early stages of a work package add value to the project. Also, increased pressure should be put on the System D program office to move more quickly with the re-baselining, as it was in the six months between its beginning and implementation that the team was "running blind" and had to create the EVMS data manually in order to use it as a management tool.

6 Research Summary and Conclusions

The concepts of value and value stream are crucial to the philosophy of Lean, and a better understanding how these concepts relate to product development is essential for the creation of a Lean PD strategy. While others have looked at how different aspects of Lean relate to PD in various ways, I focused on learning more about value by looking at two case studies and two different value perspectives: Value Stream Management and Earned Value Management. By looking at these projects at Company X with two different value lenses and comparing them, a great deal was learned about each and about how they relate to value and incorporating Lean into product development.

First and foremost, both projects showed that the two value perspectives are comparable. The superposing of the EV data on top of each PDVSM proved to be a valuable approach, as it enabled an easy visual comparison of the project with the official value measurement. The comparison also made misalignments between the perspectives visible, and illustrated that similar levels of resolution are necessary for EVMS data to be adequately compared to PDVSMs.

Creating each PDVSM was a difficult task, requiring many iterations and feedback from the teams involved. Determining what really constituted a task worthy of representation and, similarly, what level of detail to map the value stream at was a constant issue on both projects. Neither project lent itself well to strictly using the format proposed by McManus (2004), and I employed several new symbols in my maps (including ones for reciprocal information flow and requirements changes). Using Morgan's (2002) suggestion of having a timeline on the maps ultimately made the comparison to the EVMS data more feasible. His other suggestion of using different horizontal bands for different functional groups (commonly referred to as "swim lanes") proved essential for making the Project R PDVSM understandable. The special circular representation for the requirements reuse analysis and development task in Project S was essential in properly representing this process. The PDVSMs reflected how value was created in the projects, helping me understand the processes and proving of interest to the team leaders. They illustrated the flow of information in the projects and enabled the value analysis of each task using the Oehmen framework (Oehmen, 2005). By looking at the various value analyses I made, it's evident that each project focused far more on product quality than on time or cost. Also, the tasks focused more on creating product value than process value. Whereas the high-resolution maps that Kato (2005) made enabled the identification of value, the maps I made did not lend themselves to waste identification very well.

The comparisons indicated that EVMS is not adequate for measuring value in the early stages of development. Several other problems with EVMS were identified in this research. In the case of Project S, using the number of "shalls" in the SRS book as a measure of earned value led to a loose usage of EVMS and ultimately proved to be inadequate, as it made it seem that the project was more fully complete that it really was. Also, after it was realized that too much EV had been claimed, there was no way for the Project S team to "unearn" value, who thus had to finish the project without using EVMS. In Project R, limited work package resolution and the extended re-baselining process (both beyond the control of the project team) limited the usefulness of the data for understanding how Company X thought value was earned and for comparison with the PDVSMs. Additionally, the milestones and "inchstones" used to claim value in the early part of each Project R work package were not conducive to measuring the value that was added, and new ones are needed. In both cases, the explicit link in EVMS between budget and value motivated the behavior of the teams.

These two case studies are similar and yet diverse enough that their lessons seem to be applicable to other product development processes. The projects are similar in that they were both performed by employees of the same large company, at the same facility, and for the same customer. Their products were both technologically demanding and closely related to the core competency of Company X. Each project was on the same time scale, lasting more than a year. Both used the EVMS system to monitor progress, and both had issues related to the re-baselining of the budget and schedule. Both projects fell behind schedule early and never really recovered. Outside influences from the complex project contexts impacted each project, and ultimately neither was executed as planned. Both ended over budget and behind schedule. Requirements were crucial to each project, as Project S was creating them, and Project R was significantly impacted by uncertainty and changes to its requirements. Actually, the change to the reuse basis in Project S almost could be considered a requirement change for the SRS process, and thus, these projects were even more similar. At the very least, the lessons from these cases should be of interest to Company X, but many of these issues are common to other projects within the aerospace/defense industry.

The differences between the projects make them diverse enough (within the realm of aerospace PD) to offer lessons that might be applicable to many other projects. One project dealt with hardware, the other software. Project R was very complex and had many groups involved; the other only was simpler and had but one small group of engineers. Two different phases of the PD process were covered by these projects: design and testing of a major prototype and requirements development. A spiral development context surrounded one project, but Project S lay within a more standard PD context. Project R had a very specific task breakdown; the other did not. These two projects encompass both sides of many PD issues, and thus, their lessons may be more widely applicable than expected.

The PDVSMs and value analyses presented in this paper are representations of my perception of the projects, and the comparisons presented based upon them cannot be taken as a golden standard. The framework proposed in the thesis is not a panacea for best utilization of EVMS in deploying Lean, and the true value of the comparisons I made between the PDVSMs and EVMS data was more conceptual than quantitative. The overlaying of EVMS information on top of the PDVSMs allowed them to be easily compared, but did not enable a comparison with the value analysis. Either by using a different, more quantitative value analysis method or by being creative with the information I had, I could have created a method for visually representing the value contribution of each task on the map. If this were possible, it would have made the overlapping comparison even more valuable. Moreover, the empirical testing and validation will reveal the real value of use of this comparison framework.

One of the key lessons from this research was the realization that EVMS does not really measure the value of process tasks. It merely defines outputs from sets of tasks and the *costs* associated with them, but not the *value* of them. It aspires to be a surrogate for value. As explained above, EVMS suffers several problems that other process measurements methodologies do. The metrics often mismatch the process and can motivate undesired behavior. It is inflexible in dealing plans that must change. Also, determining the proper level of resolution for recording and reporting the information is important.

Perhaps there are ways to improve EVMS. A combined use of the EVMS and PDVSM could hold great potential to both measure value and serve as a more nimble management tool than standard EVMS. To make such a combined system, a project team could start with a high-level WBS provided by the customer, and analyze the major tasks therein for *how* they add value. This analysis could be done using the Oehmen framework (2005) or some other value system. Once the top-level tasks have been analyzed, the budget could be distributed among them more appropriately than they are in current practice (which is based upon historical task cost performance).

After this allocation, the lower levels of the WBS could divide the work into smaller tasks, and a PDVSM could be made to indicate the information flow between the tasks. By analyzing the value of each task in this map, the project leaders could more effectively assign task weightings (to improve EV fidelity to reality) and allocate budget among the tasks. Also, a comparison could be made for each task to weigh its cost up against the value it adds. If a task is determined to not be worth its budgeted cost, another way to add the type of value it adds should be found, or the task should be dropped outright. If neither of these options is possible, and a task is deemed unavoidable, the task should become the focus of a kaizen event to improve its individual performance. Such an event would need to be performed independently of the implementation and usage of this hybrid system.

By continually comparing the EVMS data and PDVSM, leaders should be better able to track the progress of a project and make better decisions about how to make agile course adjustments. Perhaps having a superimposed EVMS/PDVSM comparison that was updated weekly or monthly would prove useful for visual analysis and for identifying when action was needed. Such a system would enable leaders to see when there was a stretch where little value of one type was added and then adjust accordingly. This hybrid value management system also would let them ensure that all types of value needed on the project were added.

This proposed combination system that encompasses EVMS and PDVSM is far from reality. The vision as presented is similar to the research being done by Oppenheim (2004) which uses PDVSMs to help project planning. Such a system could ultimately help in the creation of the Lean PD display proposed by Graebsch (2005).

To make this system, we need to better understand value, its flow, and how to measure the two. Getting to this state requires more research to be done. Other perspectives on value need to be explored. Chase (2001) and Browning (2003) sought to understand value by focusing on risk reduction. Graebsch (2005) presented how information creation can be considered valuable. Exploring each of these perspectives on value (along with others) would help create a better overall understanding of value.

Research should also be done into how to more accurately analyze value. While the Oehmen (2005) framework has proven useful for this first analysis, perhaps subsequent research will use another qualitative value framework, such as Chase's (2001). Also, research should be done on how to turn such frameworks more quantitative. Perhaps this could be as simple as relatively weighting the contributions of each task. The value analysis could also be much more strictly quantitative such as what Browning (2003) proposed or something akin to Multi-Attribute Utility Theory. Ultimately, exploring how each of these methods expresses task value will help the application of Lean to PD.

To help make this system, the PDVSM process and its representation techniques need to be further refined. Specifically, figuring out how to represent the value of a task (from any of the above analytical techniques) would prove very helpful for visual management. If the PDVSM used in this hybrid system could illustrate the analyzed value of a task and display it next to the EVMS value measurements, it would make this system far more usable and enable better decision-making about how to redirect the project.

This proposed hybrid system also requires research into making forward-looking future VSMs. One of the original goals of this research was to try to make such realistic predictive maps, but due to changes in the research plan, this was abandoned. These maps are not the same as the idealistic future state VSMs that are suggested by McManus (2004). These maps would be realistic and based on the current state, but would be made before a project's inception. As the project progresses the initial map could be compared to subsequent versions of the map for future analysis. There has been little work done on making predictive VSMs, and research into how to make them would prove valuable.

Perhaps existing IT systems could be leveraged to help make the VSMs from the WBSs. This would require detailed research, but might even enable linkage to critical chain analysis (which would further enable good project management). Further research should be done in determining to how to easily make electronic PDVSMs, and how to tie this software to that used for EVMS entry and tracking. Regardless of how one must go to get to a future Lean state of product development and management, for now managers should try to use VSMs to improve their EVMS usage.

7 References

Bauch, Christoph. *Lean Product Development: Making Waste Transparent,* Massachusetts Institute of Technology and Technical University of Munich, 2004.

Browning, Tyson R., *Modeling and Analyzing Cost, Schedule, and Performance in Complex System Product Development,* Doctoral thesis in Technology, Management and Policy, Massachusetts Institute of Technology, December 1998.

Browning, T. R., *On Customer Value and Improvement in Product Development Process,* Systems Engineering, pp 49-61, 2003.

6

Chase, James P., *Value Creation in the Product Development Process,* Masters thesis in Aeronautics and Astronautics, Massachusetts Institute of Technology, December 2001.

Graebsch, Martin, *Information and Communication in Lean Product Development.* Diploma Thesis, Massachusetts Institute of Technology and Technical University of Munich, 2004.

Joglekar, Nitin R., Whitney, Daniel E., *Where Does Time Go? Design Automation Usage Patterns during Complex Electro-Mechanical Product Development,* presented at the LAI Product Development Winter 2000 Workshop, Folsom CA, January 26-28, 2000.

Kato, Jin. *Development of a Process for Continuous Creation of Lean Value in Product Development Organizations,* Masters thesis in Mechanical Engineering. Massachusetts Institute of Technology, 2005.

Martin, James, *Systems Engineering Guidebook: A Process for Developing Systems and Products,* CRC Press, New York, 1997.

McManus, Hugh L., "Outputs of the Summer 1999 Workshop on Flow and Pull in Product Development," The Lean Aerospace Initiative Working Paper Series WP00-01, January, 2000.

McManus, Hugh, *Product Development Value Stream Analysis and Mapping Manual (PDVSM)-Beta Draft,* Cambridge, MA, MIT Lean Aerospace Initiative, 2004.

Millard, Richard L., *Value Stream Analysis and Mapping for Product Development,* Master's thesis in Aeronautics and Astronautics, Massachusetts Institute of Technology, June 2001.

Morgan, James, *High Performance Product Development: A Systems Approach to a Lean Product Development Process*, Doctoral thesis in Industrial and Operations Engineering, University of Michigan, 2002.

Murman, Earll, et al., *Lean Enterprise Value: Insights for MIT's Lean Aerospace Initiative*. New York: Palgrave. 2002.

Oehmen, Josef. *Approaches to Crisis Prevention in Lean Product Development by High Performance Teams and through Risk Management,* Massachusetts Institute of Technology and Technical University of Munich, 2005.

Oppenheim, Bohdan, *Lean Product Development Flow,* Systems Engineering, Vol. 7-4, 2004.

Sawhney, M. and Parikh, D., *Where Value Lives in a Networked World,* Harvard Business Review, 2001, pp 79-86.

Shook, J., Rother, M., *Learning to See. Value Stream Mapping to add value and Eliminate Muda,* Brookline, MA: The Lean Enterprise Institute, 1999.

Slack, Robert A., "Application of Lean Principles to the Military Aerospace Product Development Process," Masters thesis in Engineering and Management, Massachusetts Institute of Technology, December 1998.

Womack, James & Jones, Daniel. *Lean Thinking: Banish Waste and Create Wealth in Your Corporation.* New York: Simon & Schuster, 1996.

Womack, James, Jones, Danie, and Roos, Daniel, *The Machine that Changed the World: The Story of Lean Production*, HarperCollins, New York, 1990.

Young, Marvin, "Engineering Idle Time Metrics," presented at the LAI Product Development Winter 2000 Workshop, Folsom CA, January 26-28, 2000.

9/17.05