Case Studies in DSM: Utilizing the Design Structure Matrix to Improve New Product Introduction

by
Julie W. Go

B.S., Industrial Engineering and Operations Research, University of California at Berkeley (2000)

Submitted to the MIT Sloan School of Management and the Engineering Systems Division in partial fulfillment of the requirements for the degrees of

Master of Business Administration
AND
Master of Science in Engineering Systems

In conjunction with the Leaders for Manufacturing Program at the

Massachusetts Institute of Technology
June 2007

Signature of Author ______________________________
Engineering Systems Division
MIT Sloan School of Management
May 11, 2007

Certified by ______________________________
Daniel E. Whitney, Thesis Supervisor
Senior Research Scientist, Center for Technology, Policy, and Industrial Development
Senior Lecturer, Engineering Systems Division

Certified by ______________________________
Sara L. Beckman, Thesis Supervisor
Senior Lecturer, Haas School of Business

Certified by ______________________________
Donald Rosenfield, Thesis Reader
Senior Lecturer, Sloan School of Management
Director, Leaders for Manufacturing Program

Accepted by ______________________________
Debbie Berechman, Executive Director of the MBA Program
MIT Sloan School of Management

Accepted by ______________________________
Richard de Neufville, Professor of Engineering Systems
Chair, Engineering Systems Division Education Committee

(THIS PAGE INTENTIONALLY LEFT BLANK)

Case Studies in DSM: Utilizing the Design Structure Matrix to Improve the New Product Introduction Process

By
Julie Go

Submitted the MIT Sloan School of Management and
Department of Engineering Systems on May 11, 2007
In partial fulfillment of the requirements for the degrees of
Master of Business Administration and
Master of Science in Engineering Systems

ABSTRACT

This thesis describes a project that applies the Design Structure Matrix (DSM) in support of the Manufacturing Excellence (MX) program at Cisco Systems, Inc to reduce the cycle time of new product development initiatives (NPI). Because they are inherently iterative with interdependent tasks, NPIs are difficult to manage. Two case studies applying the DSM were performed and used to study the inputs and outputs of the process as well as the dependencies between the process steps.

Both case studies indicated that defining product requirements and needs upfront helped to eliminate rework later on in the process. The DSMs also showed that cycle time and standard deviation of cycle time were especially sensitive to interactions between changes in the Bill of Materials (BOM) and other tasks. In fact there was a "tipping point" where reducing the dependency between tasks could yield significant reductions in cycle time and standard deviation of cycle time.

More significantly, the case studies highlighted the large number of stakeholders involved in the process and revealed the degree to which engineering and manufacturing must work together to reduce NPI cycle times. In fact, the name "Manufacturing Excellence Initiative in NPI" is a misnomer. New Product Introduction is not just the job of manufacturing but is highly integrated between such groups as marketing, design, and engineering. If the Mx Initiative in NPI is to fully meet its potential, all of these groups must fully realize this. In addition, there is a need for process infrastructure, data infrastructure, and close examination of incentives.

This thesis thus shows that in order for Cisco's process improvement initiatives to succeed, buy-in from all relevant stakeholders must be won.

Thesis Supervisor: Daniel E. Whitney
Title: Senior Research Scientist, Center for Technology, Policy, and Industrial Development
Senior Lecturer, Engineering Systems Division

Thesis Supervisor: Sara L. Beckman
Title: Senior Lecturer, Haas School of Business

(THIS PAGE INTENTIONALLY LEFT BLANK)

Acknowledgements

This thesis would not have been possible without the support of Cisco Systems, Inc. Raj Saxena provided outstanding guidance and coaching throughout this project. Special thanks to Vah Erdekian and Jim Miller who provided help throughout the project from the Executive level and who continue to be strong supporters of the Cisco/LFM relationship. Several others, including Karen Ashley, Mike Ruddick, Mike Lydon, Mark Cavanaugh, Trina Jahnsen, Brad Warwick, Derrick Kidani, Joe Puyat, Duke Hoang, Jason Stuck, Charu Agarwal, Manny Ocampo, Carl Woodward, Stan Lim, Dave Luck, and Hoang Pham were also valuable resources for me. I was constantly amazed by how open Cisco employees are and how willing they are to help. Thank you for making my seven months at Cisco productive and fun.

I would also like to thank the Leaders for Manufacturing faculty and staff for giving me the incredible opportunity to take part in a program that leaves a sustainable legacy on the world. Thank you for your hard work making this happen year after year.

During the course of my internship and thesis, I had the privilege and honor to work with Dan Whitney and Sara Beckman, my thesis advisors. Thank you for your time and energy during this entire process as well as for constantly challenging my thoughts and logic. You have given me countless new ideas that I will continue to ponder for times to come.

Special thanks to my friends and classmates in the LFM and MIT Sloan Classes of 2007. Without you all, the times never would have seemed so good. Don't stop believin'!

Lastly, thank you to my family. Thank you Mom and Dad for always believing in me and pushing me to succeed, Eddy for giving me the tough advice and setting the bar high, Laurie for your constant optimism and support, and to Nora, thanks for everything else!

(THIS PAGE INTENTIONALLY LEFT BLANK)

Table of Contents

Table of Figures

1.0 Introduction

The product development process is difficult to predict as it involves many interconnected decision-making activities that are performed both by individuals and by groups. The output of product development activities is also variable; in many cases the output is information that can change given other pieces of information. This causes the process to be highly iterative with tasks or entire groups of tasks redone. Therefore, structure is needed to understand how to schedule tasks and product launches.

As Eppinger (2001)[1] points out, while information exchange and iteration allows for experimentation and innovation, excessive iterations are counter productive. Therefore, innovation and iterations must be carefully planned and managed.

This thesis describes a project undertaken at Cisco Systems as part of an effort to understand and improve its product development process, specifically reducing the time that it takes to launch a new product.

This thesis begins with history of Cisco System and its current market position. It goes on to describe Cisco's manufacturing model and strategy, including the New Product Introduction (NPI) process and the organizational structure. Then, the Design Structure Matrix (DSM) methodology is explained as well as and how the DSM is useful to understand and study Cisco's process. The thesis then applies the DSM methodology in two different cases using various features of the DSM tool. The goal of these case studies is first to highlight cycle time improvement activities within the NPI process and then to show the importance of the organizational structure in process improvement. Themes and learning from the case studies are then examined within the context of the organizational structure. Finally, recommendations are given to address the learning from the DSM.

The next chapter provides information on how Cisco grew up as well as its current position in the market today.

2.0 Cisco Systems History

This Chapter addresses the history of Cisco Systems as well as its position in the market. This is important to understand because its position in the market affects how tools such as the DSM can help Cisco. Cisco Systems is an industry leader in telecommunications equipment and services. After tremendous growth in employees, revenue and stock price through the 1990s, Cisco experienced a period of significantly less growth. Now, Cisco is experiencing a second period of tremendous growth. This growth will be fueled by the new products getting to market.

Cisco was founded in 1984 by a group of computer scientists from Stanford University and went public in 1990. Today, there are over 50,000 employees worldwide. Throughout its life, Cisco has been an active acquirer of and investor in innovative startups, acquiring 108 companies since 1993. However, organic development is still a large part of Cisco's culture with over $3.22 Billion spent on R&D in Fiscal Year 2005. At the end of FY07 Quarter 1, Cisco had over $8 Billion in revenues. Figure 1 shows Cisco's growth throughout the 1990s, falling with the market downturn in 2002 and growing for the past four years. Figure 2 represents Cisco's acquisitions and revenue growth. Cisco steadily acquired companies throughout the 1990s and even though it acquired fewer companies during the downturn, it remains an active acquirer.

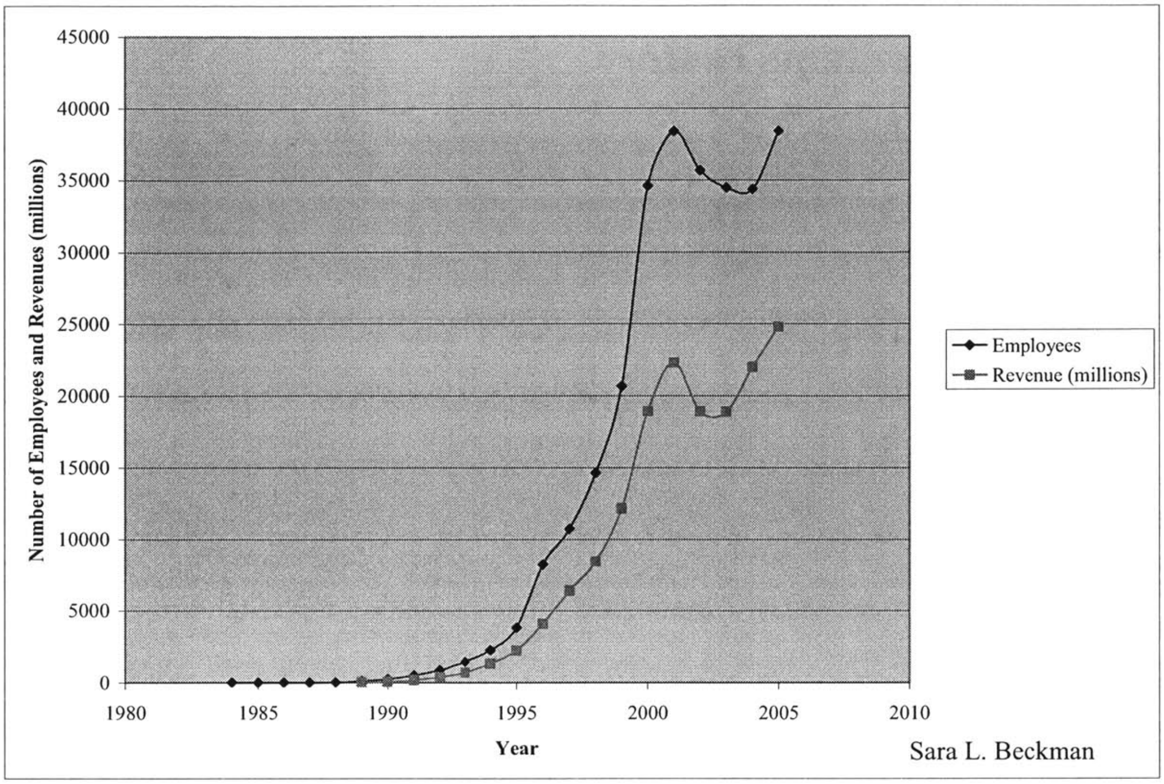


Figure 1: Cisco growth in employees and revenue

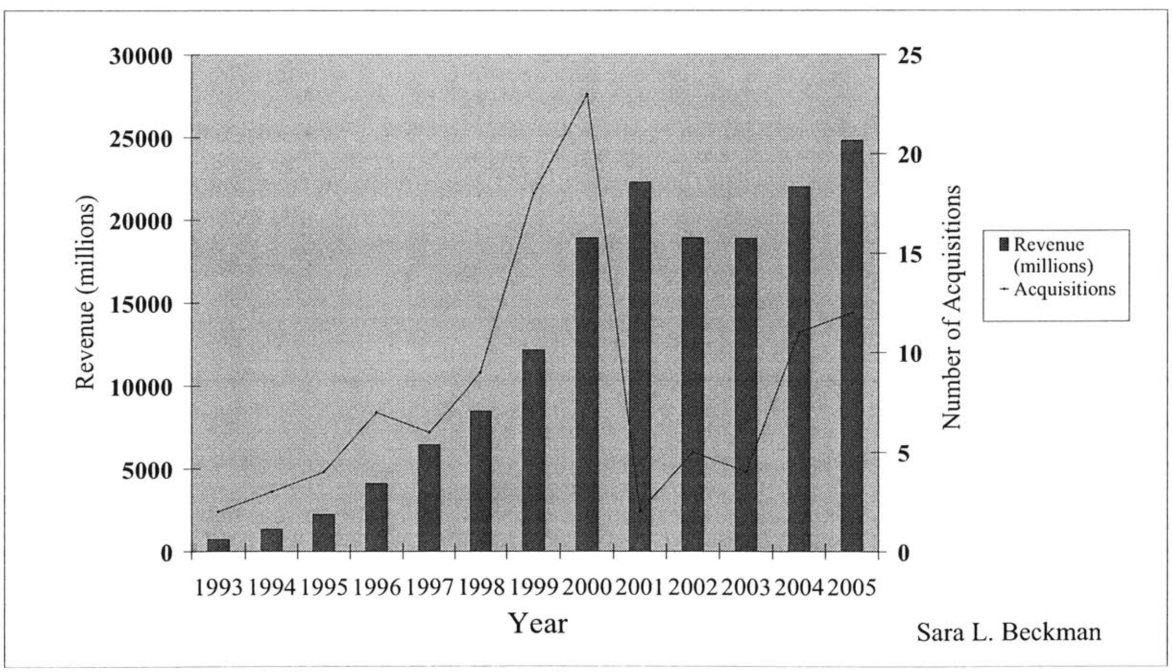


Figure 2: Cisco's acquisition and revenue growth

2.1 Market Position

Cisco is one of the leading suppliers of network switches and routers in the telecommunications industry. In 2005, the company had a market share of about 70% and 73% in the internet switches and routers segments, respectively. In 2004, Cisco's revenue was more than the combined revenue of its main competitors, which include Avaya, Juniper and Nortel Networks. Other major competitors are 3Com, Alcatel, Avaya, D-Link Corporation, Dell, Fujitsu, HP, Lucent Technologies, Nokia, Nortel Networks and Siemens, among others. Cisco is playing in several different markets, and some of these competitors are only in a few of those markets.

Currently, the company has more than 200 offices in around 60 countries worldwide. In addition, the company has a stable financial position. Its cash flow from operations increased from $7.0 billion in fiscal 2004 to $7.6 billion in fiscal 2005. Currently, the company has zero bad debts. To date, Cisco has been successful in aligning its business with the new market opportunities, and continues to gain share in new and established markets. Because of its dominant position in the market, Cisco has significant competitive advantages including economies of scale, a broad product portfolio, a large customer and installed base and large research and development resources.

In this section, we explored Cisco's current situation and concluded that while they have a storied young history and are currently in an enviable position, there is still work that needs to be done to maintain this position. In the next section, we look more closely at Cisco's manufacturing strategy and how it fits in with the overall business strategy.

3.0 Cisco's Manufacturing Model and Business Strategy

This chapter explores several concepts that relate to outsourcing and competitive dynamics in order to have a frame of reference with which to study Cisco.

3.1 Focusing on the Core, Outsourcing Context

This section covers the idea of core and context and how it pertains to the Cisco's outsourced model of manufacturing.

Most of Cisco's products are manufactured by outside contract manufacturers. This is driven by the idea of "core" and "context" activities. According to author Geoffrey Moore, the word "core" is what distinctively differentiates you in the eyes of the customer. By focusing on the core, a company puts as much human and financial capital behind processes that have this impact. Everything outside of this is defined as "context." This is not to say that context is not extremely important to the success of the company. If a firm fails to fulfill a context task, it could easily lose a customer, be out of compliance with regulation or law, or alienate an employee, a partner, or the community at large.[2] At Cisco, much of the direct manufacturing and assembly work is considered to be "context," while activities such as managing the supply chain and design engineering are considered "core." The idea is to work closely with partners and support them where needed.

As Cisco CEO John Chambers expressed, "...Moore asserts—and I agree—that by focusing on what is core in your business, outsourcing context activities, and leveraging your competitive advantage, your company can achieve maximum shareholder value. And in this (or any) economic environment, managing for shareholder value is critical for success."[3]

Similarly, Fine and Whitney (1996)[4] argue that companies should retain not only those skills directly involving product or process but also those skills that support the very process of choosing which skills to retain.

3.2 Outsourcing and Clockspeed

In the past several decades, outsourcing has transformed from transactional partnerships to more collaborative partnerships. Understanding what and when deciding what and when to outsource is an important decision for any firm. A few concepts are useful to use as a lens in which to think about firm resources and capabilities and when to outsource. These are explained in the next sections.

3.2.1 Clockspeed and its Implications on Product Development

In Charles Fine's book Clockspeed,[5] he defines Clockspeed as the rate at which an industry evolves. This industry Clockspeed is based partially on the product, process, and organizational Clockspeed. An example of a fast Clockspeed industry would be the motion picture industry where movies could have lives measured in weeks or days. On the other end of the spectrum is the airline industry with its Clockspeed in the decades or even more.

There are two main drivers of Clockspeed: technological innovation and competitive pressure. In a situation with high innovation and pressure, there tends to be the fastest Clockspeed. This makes sense; in an industry such as semiconductor manufacturing, there is both high innovation and high competition and firms have been following Moore's Law for the past four decades.

A firm such as Cisco is considered to have a fast Clockspeed with changes in the telecommunications industry happening at rapid rates. In industries like this, firms have more opportunities to disrupt the industry and gain advantage. Therefore, it is even more important to make rapid decisions.

Because Cisco outsources its manufacturing, the supply chain complexity is even higher. Coordination in the supply chain design is crucial and development of the capability has very direct effects on the speed of its product development process.

3.2.2 Capability chains

One central theme in Clockspeed is the idea of a capability chain. The idea is that no capability exists in a vacuum and that the linkage between capabilities is where value lies. Analogously to a mechanical system, a chain is only as strong as its weakest link. In fact "no capability is more critical than that of superior design of one's capability chain – from the final consumer all the way upstream to the sources of raw materials and new technological concepts." [6]

Firms must understand the importance of building robust capability chains as well as having mechanisms for understanding of their capabilities. In an outsourced model of operation such as Cisco's, capabilities must include those of its contract manufacturing partners and the mechanisms by which the firms' capabilities are linked.

Fine also makes the case that no advantage is forever. Even though Cisco is currently dominant in the market, this can quickly change.

3.3.3 Temporary Advantage

Applying the Clockspeed logic, firms can only hold temporary advantages, and the faster the Clockspeed, the faster advantages turn over. Cisco must continue to evolve in order to maintain this position. Fine cites such examples as the IBM of the 1970s and the General Motors of the 1950s and 1960s as examples of firms that once seemed unassailable but needed to adjust in the face of temporary advantage. Another example is photolithography and how the dominant firm has changed with each new generation of product technology.

Similarly, Teece, Pisano, and Shuen (1997)[7] argue that firms develop and sustain competitive advantage through development of dynamic capabilities - the ability to adapt competences to changing market conditions and the ability to rearrange skills, resources, and functional competences to match the changing market.

In summary, Cisco is a company in a high Clockspeed industry, meaning technological innovation and competitive pressure is high. Because of this, their competitive advantage may be even more temporary and tenuous. Further, because Cisco utilizes an outsourced model of production, not only are Cisco's and their contract manufacturer's individual capabilities important, the link between their capabilities is also important. Therefore, even though there can be organizational inertia that prevents a firm from moving forward, it is important for the managers to keep doing something to stay ahead.

4.0 The New Product Introduction Process

In Chapters 1-3, Cisco's history, current market position, and manufacturing strategy were discussed. This section looks more deeply at the NPI process including its process steps and the organizational structure surrounding it. The Manufacturing Excellence (Mx) Initiative in NPI and its role will also be explained. In Chapters 6 and 7, case studies focused specifically on the NPI process will be performed.

4.1 The NPI Process

Cisco's NPI process is structured in a Phase-Gate Model. There are seven gates: Concept Commit, Execution Commit, Prototype Builds, Pre-Pilot, Pilot, First Customer Ship, and Total Time to Quality and Volume, At each gate, there is a review with the major stakeholders and the project must meet certain criteria to pass.

Concept Commit is an event where marketing product management presents a conceptual draft of customer requirements to engineering for new product development or existing product enhancement in the form of a Product Requirements Document (PRD). Concept Commit can be done in one-on-one settings or as a cross-functional meeting with representatives from marketing, engineering, manufacturing, and customer advocacy present. Marketing usually drives the Concept Commit.

The next step is Execution Commit. This is where the senior level, cross-functional management team and product teams agree to allocate resources and fund the development as described by the Program plan and the System Functional Specification. The System Functional Spec is the engineering of the requirements specified in the PRD. The New Product Program Manager is responsible for completing the Initial Manufacturing Plan for Execution Commit.

After Execution Commit, design is done along with Bill of Materials (BOM) risk analysis and supply chain definition. Many products are derivative products that build on existing designs. However, the phase-gate process is the same for both derivative and brand new products.

Once designs have passed design for manufacturing guidelines, prototypes are built. This is the longest part of the process. Cisco manufacturing works with the CMs to start prototyping the products. Each prototype is sent back to Cisco for testing and design changes. Typically, several iterations are done before a product is ready for Prepilot.

Prepilot is the last prototype build before a product is readied for launch and is meant to test the new products through final production processes to reduce risk at Pilot. This is the last chance to work out the kinks before Pilot, the first production build.

If all goes well during the Pilot and there are waiting customers, the first product is shipped (First Customer Ship). After FCS, the product is ramped; the product passes through the Total Time to Quality and Volume (TTQV) gate. At this point, the product is no longer considered NPI and is passed to the Manufacturing Operations team for full production.

Within the NPI process, many of the steps require tight coordination between both smaller groups and larger organizations. Because manufacturing is outsourced, this coordination is necessary between companies as well as within companies. As a result, the attitude toward contract manufacturers tends to be that of "partners" rather than suppliers. Figure 3 indicates which groups are involved at each stage of the process.

	Groups Involved			
NPI Step	**Marketing**	**Engineering**	**Cisco Manufacturing**	**Contract Manufacturing**
Concept Commit	X	X	X	
Execution Commit	X	X	X	X
Prototype Builds		X	X	X
Prepilot		X	X	X
Pilot		X	X	X
First Customer Ship	X	X	X	X
Total Time to Quality and Volume	X	X	X	X

Figure 3: Groups involved in the NPI process

4.2 Product Development Organization Structure

In any product development, the organizational structure plays a large role in how work gets done and how it is coordinated. This section details the roles of the main players in new product development.

Cisco's organizational structure is a classic functional organizational structure. At the senior executive level are Operations, Sales, Development, and functional areas such as Finance. Manufacturing sits inside the Operations organization. Figure 4 shows a general organization chart for Cisco.

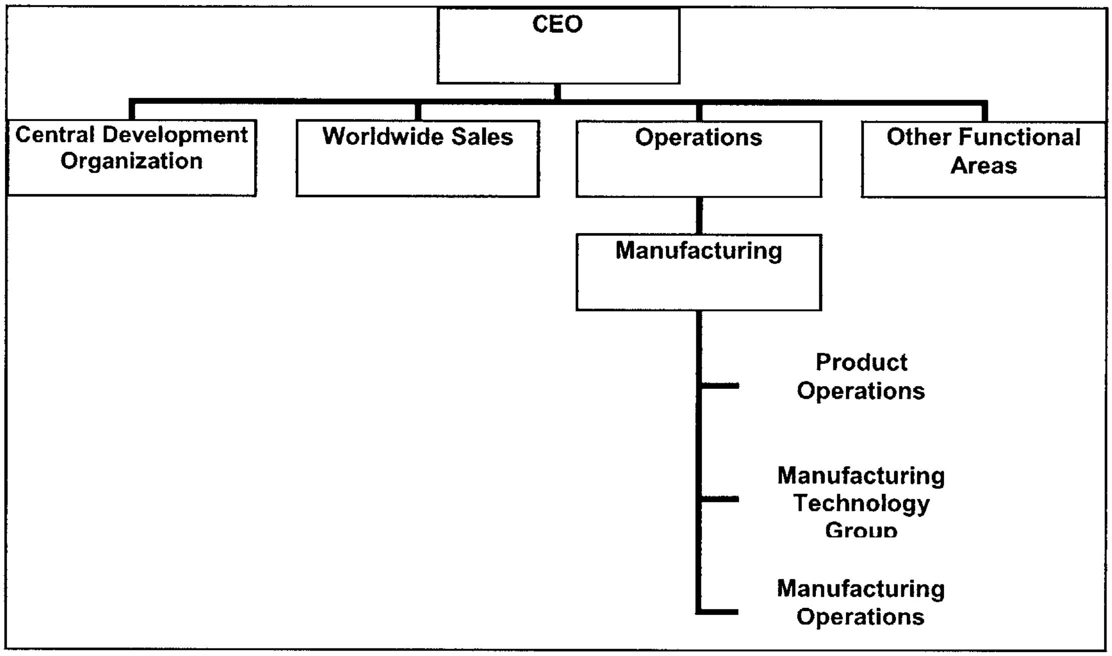
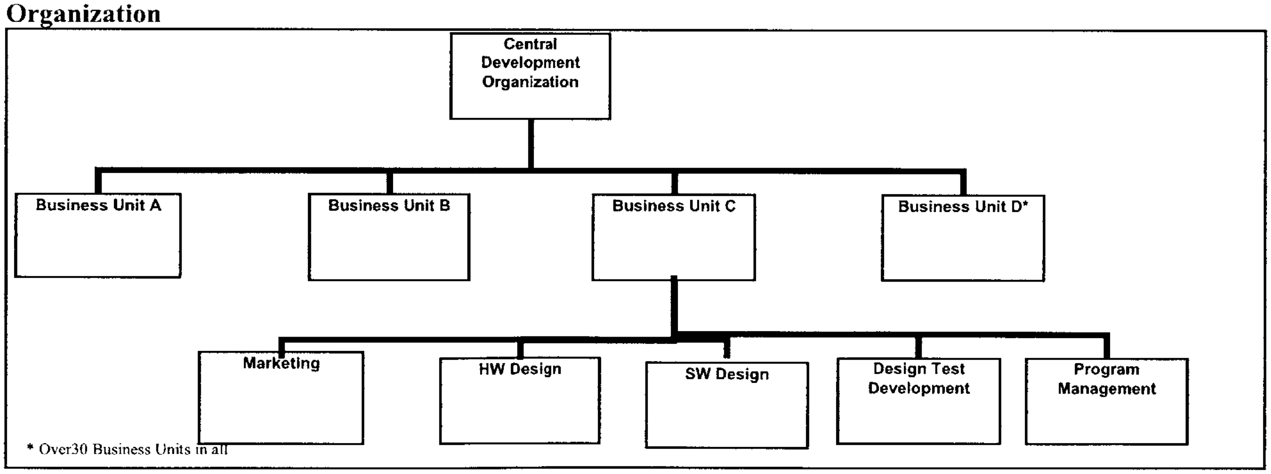


Figure 4: Cisco Organizational Structure

Within the Central Development Organization (CDO), over 30 Business Units (BU) are responsible for the design and marketing of products. BUs are organized according to product family. Within design, there are both software and hardware functions. Figure 5 shows how the CDO is organized.

Figure 5: Central Development Organization



Each BU interfaces with Manufacturing through the Product Operations group. The Product Operations group is organized into groups that specifically interact with the

BUs. For example, each Product Operations group will interface with a specific BU on its products. This allows for the BUs to have a single point of contact within manufacturing.

There are several key manufacturing roles on the NPI team and they are explained below. Figure 6 shows how they are organized within the Product Operations Group

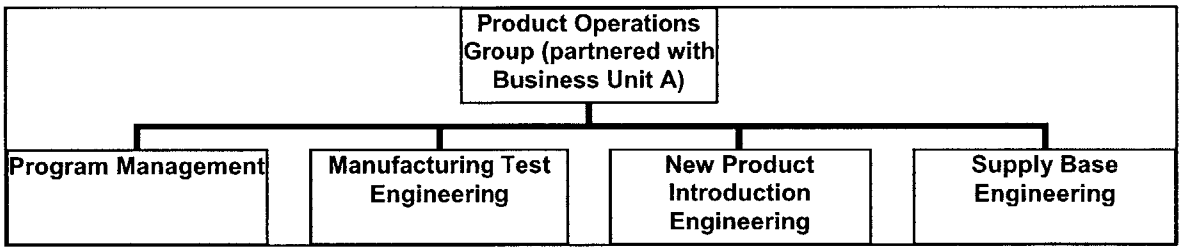


Figure 6: Product Operations Structure

The New Product Program Manager (NPPM) is Manufacturing's representative to the Business Unit and on the Engineering Product Team. The NPPM leads the NPI team ensuring that all necessary representatives are included and all tasks are completed. This includes setting up the meeting times for the team, setting the agenda, and performs follow up on problems.

New Product Introduction Engineer (NPIE) owns interaction with Engineering to ensure the technical manufacturability of the product. This includes mitigation of assembly risks, BOM risks, technology risks, and design for manufacturing. The NPIE also works closely with the materials organization to make sure the parts are on order.

The Manufacturing Test Engineer works with the BU test engineering to write test scripts that automate the testing process in manufacturing. They typically take software diagnostics and automate them. They are also required to be in close communication with the diagnostics engineers in the event of bugs or other issues that need to be resolved. They are also in communication with the test engineers who work for the contract manufacturer

The Supply Base Engineer (SBE) owns the ongoing process assessment, process development and process capability feedback associated with the Contract Manufacturers. SBEs work closely with the suppliers to perform assembly and inspection qualifications, communicate related Design for "x" issues, manage documentation needs and monitor yields and drive improvements.

Outside of the Product Operations team, there are other players within manufacturing who have important roles. Within the Commodities group, the NPI Buyer/Planners are responsible for mitigating BOM risk sourcing issues and managing the supply chain partners during the proto build and pilot process. A member of the Demand Planning organization, the Demand Planning Manager is responsible for setting up Oracle so that demand can be placed against the product and then for loading the production demand based on forecasts from the Business Unit.

The BU obviously also has many players involved in the process including a BU program manager, HW Design Engineers, Development Test Engineers, and SW Design Engineers. Manufacturing integrates tightly with them during the design and manufacturing process. Even though some members of the Business Units were interviewed during the course of this project, efforts of the Mx initiative were much more focused on manufacturing. Through this project however, it was found that collaboration and partnership with the design organization in the process improvement is critical.

The Contract Manufacturer (CM) is responsible for the actual assembly of product and shipping it to customers. During the NPI stage, the CM builds all prototypes and partners with Cisco in the development of the process.

4.3 The Manufacturing Excellence (Mx) Initiative

The Manufacturing Excellence Initiative is coordinated through the Manufacturing Technology Group at Cisco; generally, this is the group that is involved in process changes and improvements. The strategic intent of the Mx Initiative is to deliver world class process, product, and leadership excellence, such that Cisco is the undisputed leader in supply chain management, guaranteeing Cisco's sustained innovation and competitive advantage. New Product Introduction (NPI) is one of 6 Focus Areas within Mx and aspires to develop best-in-class processes and capabilities for introducing new products to the market. These initiatives are manufacturing-centric. Some pull in members of various engineering teams but this is ad hoc. The engineering teams have a process improvement team that works loosely with the Mx NPI team but they do not have formal channels in which to partner.

The Mx NPI initiative is divided into four tracks or sub-initiatives, under which 8 projects were created. Each project is staffed with a Program Manager, a Technical Leader, and subject matter experts (SME) from across Cisco's organization. The DMAIC Process is being used as a framework for all process improvements, and there is a Master Black Belt to facilitate and coach the teams on the process. This project was associated with the Mx NPI initiative to reduce cycle time between Concept Commit and Total Time to Quality and Volume.

Mx is using a DMAIC framework that is being deployed along with targeted six sigma black belts and master black belts being utilized in specific areas. The Mx NPI area has a master black belt and 2 black belts supporting this effort, all of whom are relatively new to the company.

In this chapter, the NPI process at Cisco was described as a phase gate model. Many of the different players in the organization were explained as well. Cisco's NPI process has a large number of players who reside in various groups and organizations. In addition, as new products move through the NPI phases, various groups become more or less involved. In the next chapter, the Design Structure Matrix will be explained along with how it can be used to learn more about and improve the NPI process. In the two chapters following that, two case studies will be used to demonstrate how the DSM was used to reduce cycle time in the NPI process.

5.0 The Method – The Design Structure Matrix

The Design Structure Matrix (DSM) offers a method to figure out the tasks necessary to complete a project and also which tasks depend on which tasks. In this project, the DSM was used to understand the new product development process at Cisco.

The NPI process at Cisco is generally understood to be a series of interdependent tasks. However, the DSM gives a simple framework to clearly map what those tasks are and how their relationships affect the whole process. In this chapter, the history and applications of the DSM are detailed as well as the mechanics of how the DSM works. In Chapters 6 and 7, the DSM is used in two case studies.
Also included in this chapter are short explanations of alternate methods.

5.1 DSM history and applications

There are four types of DSM models: component based, parameter based, activity based, and team based. In a team based DSM, organizational analysis and design based on information flow among various organizational entities. Individuals and groups participating in a project are the elements being analyzed. A component-based DSM documents interactions between elements in complex system architecture. Parameter-based DSM captures dependencies between decisions on design parameters. The DSM used in this project is an activity based DSM which focuses on input/output relationships of tasks.

In the literature, there have been several applications in the DSM. Many of these ideas rely heavily on the work of Whitney and Eppinger, et al. First, the DSM gives the ability to visualize the complex system and manage iteration formally. For example, decoupling of tasks can ultimately speed up the product development process by cutting out steps and iteration. However, there is a tradeoff between speed and the level of innovation; less iteration can result in a suboptimal product by not taking advantage of the information exchange. Therefore, firms must make strategic decisions about which factors are most important and work that into their strategies.

Second, the DSM can give insights into restructuring the organization. In an example, Eppinger (2001) uses the DSM to improve communication across product development teams at an automobile manufacturer. The result was to reorganize its 4 product development teams into 4 overlapping teams with an additional systems integration team as a structural mechanism to ensure overall work.

Another useful application is linking product design and organizational design. Many studies have used the DSM to study product design or organizational design. However, Sosa, et al (2004),[8] integrates product architectural knowledge with the organizational design of product development teams. They show through DSM analysis that the likelihood of misalignment is greater across organizational and system boundaries and that indirect interactions are an important coordination mechanism within boundaries.

The DSM also highlights areas of improvement in both complex and simple situations. In Eppinger (2001),[9] the DSM was applied to the extremely complex process of semiconductor manufacturing. He found that most of the major steps had many iterative steps including a significant number of unplanned iterations. For example, a poor outcome on a manufacturing test could result in an entire product redesign. The value of DSM to the organization lies in pointing out these unplanned iterations and giving the firm an opportunity to change the process in order to mitigate or eliminate this iteration. A case study to be described later on in this thesis is an example of a simple DSM.

5.2 DSM- The Mechanics

This section explains the mechanics to the DSM. Later on, these steps are used in two case studies. The following are the steps to develop an activity based Design Structure Matrix

1) Define the system and the scope
2) List all system elements
3) Study information flow between systems elements and build a matrix to represent information flow between the elements

4) Validate matrix with subject matter experts
5) Partition the matrix
6) Use optimized matrix to make recommendations

Note that the steps 2-4 can be iterative with changes being made as more subject matter experts lend their thoughts to the DSM.

Task relationships based on informational dependencies as parallel (no information flow), sequential (A feeds information to B) and coupled (A and B are mutually dependent on each other for information). In the example in Figure 7, all tasks are listed on both the vertical and horizontal axes. The vertical axis receives information while the horizontal axis provides information. The presence of a "1" in the box means that there is a dependency between two tasks. In this example, "A" provides information to "B". Any dependency marks above the diagonal signify feedback where a step relies on information that occurs after it is first performed. The dependency between steps "C" and "A" is an example of this. "C" provides information to "A" but "A" is performed before "C." This means that under these conditions, "C" could cause rework of "A."

Receiving information	Providing Information							
	Name	A	B	C	D	E	F	G
	A	1		1		1		
	B	1	2					
	C	1		3				
	D				4			1
	E			1		5		
	F						6	
	G			1		1	1	7

Figure 7: Simple DSM Example

Once a DSM is created, there are software macros available to partition and band the matrix. To band the matrix means to separate the tasks steps into "chunks" that can be done concurrently. The macros also reorder the tasks into optimal order. Figure 8 represents the matrix from Figure 7 in a banded form. For example, in Figure 8, tasks B, C, and D can be done concurrently and tasks E and F can be done concurrently.

Receiving Information

Name	A	B	C	D	E	F	G
	Providing Information						
A	1	1	1		1		
B	1	2					
C	1		3				
D				4			1
E			1		5		
F						6	
G			1		1	1	7

Figure 8: Banded Matrix

In Figure 9 below, the matrix is partitioned into "blocks." The DSM does not have to be banded before it is partitioned. These blocks represent a section of the process that loops back on itself.

Receiving Information

Name	A	B	C	E	F	G	D
	Providing Information						
A	1	1	1	1			
B	1	2					
C	1		3				
E			1	5			
F					6		
G			1	1	1	7	
D						1	4

Figure 9: Partitioned Matrix

The DSM provides a tool for easily seeing interdependencies between tasks as well as beginning to think about the how tight the coupling of tasks is.

In addition to using the tool, the process of creating the matrix can also bring insights to a larger number of stakeholders. Much of the value-add can come from gathering the data from stakeholders and raising the awareness of task and group interdependencies.

5.3 Other Methods

The DSM is not the only method that could be used to study the NPI process. In fact, several other methods or combinations of methods could have been used and are discussed below. Practically speaking, many of these tools are quite useful and should be considered in product development. DSM was chosen over these tools because it is very simple and intuitive way to study dependencies.

5.3.1 Value Stream Mapping

Value Stream Mapping provides a template for determining the steps that a product goes through during the process. It also provides for determining the inputs and outputs of the different steps. Using the Value Stream Map (VSM), the user can also see the where waste occurs. The VSM also provides a coherent and intuitive device for users to understand the process. Like the DSM, it also hinges on many users coming together to gather the data and inputs and can also be used for process simulation. With a VSM however, the tendency can go towards creating a linear path – while it is possible to utilize it to look at interlinked steps, the map can become quite messy.

5.3.2 Critical Path Method

Like DSM, the Critical Path Method (CPM) lists all tasks required to complete the project, records the duration of each task and also the dependencies between the tasks. Using these values, CPM calculates the start and end times for each task and determines which activities are on the critical path. The critical path is the sequence of tasks that determine the shortest time possible to complete the project. Put another way, any delay of an activity on the critical path directly impacts the planned project completion date. A project can have several parallel critical paths.

5.3.3 Theory of Constraints

According to the Theory of Constraints (TOC), any system has a constraint or a bottleneck and these constraints determine the performance of the system. There are 5 Thinking Steps in TOC:

1. Identify the bottleneck or constraint.
2. Decide how to exploit the constraint. This entails making sure that the constraint is being utilized to maximum efficiency and potential.
3. Align all other processes to the constraint.

4. Increase capacity of the constraint if required.
5. If the constraint has now moved, repeat steps 1-4.

In this chapter, the basics of DSM were explained and several alternate methods were discussed. In Chapters 6 and 7, this knowledge is applied to a case study.

6.0 DSM Case Study 1: Manufacturing Test Development Process

This section explores the first case study using the DSM at Cisco. In the case study, the steps to build a DSM are used and applied.

6.1 Manufacturing Test Development Process

This case study is intended as a way to show Cisco a simple tool that highlights the interdependencies and connectedness of its NPI process steps. The Manufacturing Test Development Process was chosen because it requires a cross section of several groups including the contract manufacturer, the engineering organization, and several areas of Cisco manufacturing. Also, because the test case development process is a manageable chunk of the process, there is the ability to clearly validate the process and learning.

Manufacturing tests are conducted on the manufacturing line at the contract manufacturer but are developed by Cisco engineers in manufacturing and design. Occasionally, design errors are found during these tests.

Manufacturing test development begins after Execution Commit and must be completed for Pilot. Software Diagnostic engineers write diagnostics (diags) to test functionality of the product. This occurs for both the system and for components. A Manufacturing Test Engineer (TE) will take these diags and automate them by writing scripts. Because one key piece of software development is finding and correcting bugs, the manufacturing test development process can be a very iterative one. During the Pre-pilot stage, the test scripts are released to the contract manufacturer and bugs can emerge here as well. This creates another area where iteration comes about during this process.

Since each of Cisco's business units follows a slightly different process, one business unit was chosen to be the case study. The Internet Systems Business Unit (ISBU) was chosen because it contains several of Cisco's flagship products which account for a significant percentage of Cisco's revenue. Because of this, even a small improvement could have far reaching impacts.

6.2 Manufacturing Test Development DSM

In this section, the DSM steps described in Chapter 4 are applied in a case study about Manufacturing Test Development.

Step 1: Define the system and the scope

For our case study, the test development process was chosen as a manageable chunk of the process. In addition, the process is centered in the manufacturing organization (specifically manufacturing test) but requires inputs from several other organizations including marketing, engineering, and the contract manufacturer. Even though the process occurs during many other process steps, only those with information or inputs related to manufacturing test development are used in the DSM.

Step 2: List the system elements

The system elements were determined first by a rough cut based on specs and other documentation. Several subject matter experts including software diagnostic engineers, manufacturing test engineers, hardware engineers, new product program managers, and supply base engineers were assembled to refine the system elements. This step was revisited during Step 3 and Step 4 as more subject matter experts were engaged.

Step 3: Study information flow between systems elements and build a matrix to represent information flow between the elements

The dependency marks were placed first by rough cut based on specs and other documentation. This rough draft was given to subject matter experts to refine the dependency marks as well as change system elements where necessary.

Step 4: Validate matrix with additional subject matter experts

This step utilizes a larger population of subject matter experts in a series of one on one meetings. These were mainly manufacturing test engineers and others who were intimately involved in the test process. When there were disagreements, the majority opinion was placed in the model. This including tweaking the system elements by adding and subtracting rows and columns as well as adjusting marks.

Step 5: Partition the matrix

This step involves running the DSM software to optimize the system. The software program reorders the task steps in order to decrease rework. It also localizes rework loops into blocks.

A high level view of the entire partitioned model can be found in Appendix A.

It became clear from the DSM that we can articulate the test development process as four main "blocks" that consist of task steps that are interdependent.

1) understand requirements
2) plan and budget
3) write test scripts
4) utilize scripts in a build

Understanding these four main steps was useful in being able to break down the process to find areas for improvement. Next, these four blocks and the learning from them are examined more closely.

Understand Requirements

Figure 10 is the block for the Understand Requirements

		understand product requirements (PRD level)	understand schedule	understand priority	understand HW specs/schematic (how to	on whether the ASIC is being exercised correctly	understand previous tests on similar products	understand central test strategy	understand EDVT test strategy
		1	12	13	2	3	4	5	6
understand product requirements (PRD level)	1	1							
understand schedule	12		12						
understand priority	13			13					
understand HW specs/schematic (how to implement the prod requirements	2	1			2	1			
understand ASICS - understand the test case on whether the ASIC is being exercised correctly	3	1			1	3	1	1	1
understand previous tests on similar products	4	1			1	1	4	1	1
understand central test strategy	5	1			1	1	1	5	1
understand EDVT test strategy	6	1			1	1	1	1	6

Figure 10: Block 1, understand test requirements

This block shows how the various test strategies are interlinked and the importance of understanding this upfront. It also showed how many different test groups there are and how coordination between the various groups could cut out rework time and effort later in the process.

Plan and Budget

The block shown in Figure 11 shows the interactions between the various test plans and the budget. Overall, there were no surprises here for the team.

		write test plan	communication needs of diags for mfg test	test plan review and approval	budgeting for new equipment
		8	**9**	**10**	**14**
write test plan	**8**	8			1
communicate needs for diags to mfg test	**9**	1	9	1	
test plan review and approval	**10**	1	1	10	
budgeting for new equipment	**14**			1	14

Figure 11: Block 2, Plan and Budget

Write Test Scripts

Figure 12, below, is the block that shows the process for writing test scripts.

	receive boards	write diags	test functionality of board	understand diag readiness	write scripts	report issues with diags	diag eng works on bugs	write and test new diag	understand new diag features	TE tweaks based on changes	design and layout boards
receive boards	16										1
write diags	1	18									
test functionality of board		1	19		1						
understand diag readiness		1		20							
write scripts	1	1		1	21				1		
find and report issues with diags		1	1		1	22					
diag eng works on bugs			1	1	1	1	23				
diag eng write and test new diags							1	24			
understand new diag features								1	25		
TE tweaks based on changes					1	1		1	1	26	
design and layout boards										1	27

Figure 12: Block 3, Write test scripts

This block is the "meat" of the Test Development process and it was also where most of the interaction with the Business Unit occurs. This block showed the Test Engineering (TE) team how interconnected the Software Diagnostic and Test Engineering teams are. While this was known generally, the block clearly demonstrated how both teams contributed to this process.

Utilize Scripts in a Build

Figure 13, below, represents the fourth block, where the scripts are ported to the Contract Manufacturers and utilized in a build.

	scripts ready to go	port script to CM	send test package to CM	CM build	CM tests funcationality of diags	CM uses scripts	CM finds and reports issues	tweak scripts for new build cycle
scripts ready to go							1	1
port script to CM	1							
send test package to CM		1						
CM build			1					
CM tests funcationality of diags				1		1		
CM uses scripts					1		1	1
CM finds and reports issues						1		
tweak scripts for new build cycle							1	

Figure 13: Block 4, Utilize scripts in a build

This block represents the point when the CM comes into the picture. Here, the test scripts are sent to the CM and used. When the CM finds issues or bugs in the program, iteration occurs. This was an obvious point to the Test Engineering group.

Step 6: Use optimized matrix to recommend improvements

Perhaps the most difficult task in utilizing the DSM is "selling" the results. In the case of test development at Cisco, the DSM results were validated by the subject matter experts. It also brought to light the blocks within the process – something that the experts inherently knew but did not fully consider until seeing the model. Subject matter experts describe tasks the way that they believe they are currently done but sometimes this clashes with what management believes is happening. This can open the conversation around what truly happens during the process. In this case, this occurred and both the people doing the work as well as the managers were able to learn more about the process. While this discovery is quite common in many companies, it is still important to highlight as an issue.

This chapter walked through a case study of developing and using a DSM for the Manufacturing Test Development process. In the course of the case study, both management and the subject matter experts were able to take a step back and look at their process more objectively. There were two main takeaways. First, defining customer and product requirements was a necessary first step that could derail the entire process if not done thoroughly. Second, the Manufacturing Test and Software Diagnostics teams also came to realize how interconnected the Manufacturing Test Engineering, Hardware Design Engineering, and Diagnostics Engineering teams are. It was valuable insight to begin to start looking at how these groups could work more closely in order to improve both the speed and quality of the process.

For example, the Manufacturing Test Engineering group used the DSM model as leverage to try to convince their partners in engineering and marketing to engage early in order to avoid rework later in the process.

7.0 DSM Case Study 2: The Entire NPI Process

During the first case study, several subject matter experts expressed the sub-optimality of addressing just one portion of the process. Therefore, a second case study was used to study the entire NPI process. This DSM represents the process extending from Concept Commit to Total Time to Quality and Volume. This DSM better illustrates the linkages between all groups involved in the process and helps the users to understand the capability chains that Cisco needs in order to be successful in its new product development.

This case study follows the same steps as outlined in Chapter 5 and that were used in the previous case study. In addition, for this case study, simulation and sensitivity analysis features of the DSM were applied.

7.1 Case Study 2: DSM Build

Step 1: Define the System and the scope

In this example, the NPI process from Concept Commit to Total Time to Quality and Volume is studied. Since the various Business Units do things differently, this case study is limited to the Internet Systems Business Unit. Later on, this model can be modified to study other Business Units.

Step 2: List the System Elements

In the first case study, one of the difficulties was decoupling the test development activities from those that did not affect test. In this case study, the difficulty was in figuring out the proper level of detail in which to represent the process. As a result, the task steps in this case study are more general than those in the previous one. The processes of finding the appropriate system elements required several iterations to make sure that all steps were significant enough to use.

Step 3: Study information flow between system elements and build a matrix to represent information flow between the elements

Determining the dependency marks was a process that required a several rounds of back and forth. A small subsection of New Product Program Managers (NPPM), New Product Introduction Engineers (NPIE), Supply Base Engineers (SBE), Test Engineers (TE), and members of the Manufacturing Excellence (Mx) team were utilized to determine the dependency marks.

Step 4: Validate matrix with additional subject matter experts

A larger group of NPPMs, SBEs, NPIEs, and TEs were utilized for validate the matrix. This process also required a great deal of clarifying the steps and dependency marks.

Step 5: Partition the matrix

A full partitioned matrix is available in Appendix B. When the analysis was run, one large block was discovered. This was the prototype stage, shown in Figure 14.

		product requirements	schedule	proto allocation	risk rating	BOM	parts qualification	mfg plan	DFM feedback pre proto	DFM feedback mitigation pr	valor run and results	schematic	diag readiness	system specification	mechanical tooling strategy	fabout custom mechanicals	second sourcing	PCB layout	EDVT - proto	MDVT - proto	dev test - proto	approved vendor list	RDT	CM builds proto	bring up proto	CAD work	fabout package CAD PCA r	DFM post proto feedback	design change
		1	2	3	4	5	6	7	8	9	10	11	12	13	15	16	19	20	21	22	23	24	27	29	30	39	41	44	50
product requiremer	1	1																											1
schedule	2	1	2																										1
proto allocation	3		1	3																									
risk rating	4				4	1																							
BOM	5	1				5	1							1			1		1	1	1	1						1	
parts qualification	6					1	6										1												
mfg plan	7			1		1		7																					
DFM feedback pre	8					1			8		1			1	1									1					
DFM feedback miti	9								1	9				1	1									1					
valor run and resul	10										10	1														1			
schematic	11	1										11						1								1			
diag readiness	12					1							12	1											1	1			
system specificatio	13	1												13															
mechanical tooling	15													1	15														
fabout custom mec	16														1	16													
second sourcing	19				1	1											19					1							
PCB layout	20					1								1				20											
EDVT - proto	21			1		1							1	1	1				21		1				1				
MDVT - proto	22			1		1								1	1					22	1				1				
dev test - proto	23			1		1								1							23				1				
approved vendor li	24				1	1																24							
RDT	27					1							1	1									27						1
CM builds proto	29							1		1	1				1	2								29			1		1
bring up proto	30																							1	30		1		
CAD work	39													1				1								39			
fabout package CA	41																	1									41		
DFM post proto fee	44																							1				44	
design change	50																		1	1	1		1					1	50

Figure 14: Block representing prototyping

Step 6: Use optimized matrix to recommend improvements

The matrix clearly shows that the largest contributor to NPI cycle time is the prototype stage. This conclusion is also supported by empirical evidence. One key finding is that some of the early tasks, including nailing down product requirements, provide inputs to many other crucial steps such as BOM changes. This shows the importance of doing the due diligence in the beginning of the process to reduce the need for rework later on. This is a common theme from the first case study.

One difficulty with fully understanding this large block is that it encompasses tasks from the BUs and the CMs as well as the manufacturing organization. However, the DSM has highlighted it as a clear issue and the model can be used to gain leverage on the BUs to at least start studying the process and beginning to understand why it takes so long to get through the prototype stage.

The DSM also demonstrates that the Mx team must make connections outside manufacturing into the CMs and BUs in order to drive the most change. This is not a new finding to the Mx team knows this. However, in its process improvement activities, the Mx team relied on the New Product Program Managers, New Product Introduction Engineers, and other members of the Product Operations group to gather inputs from the Business Units and act as "ambassadors" for the process improvement work. Since the Product Operations groups already had working relationships with the Business Units, this seemed like a good idea. However, this was not enough to gain full engineering partnership. The DSM showed how interdependent the organizations are and the importance for the Mx team itself to personally make inroads into gaining engineering support.

7.2 Simulation and Sensitivity Analysis

The second case study showed how improving the entire NPI process depended greatly on a partnership between the BUs, Manufacturing, and CMs. Gathering support though, will be difficult and it is helpful to first get an idea which areas might be high value areas to target first. This chapter builds on the second case study and uses the simulation capabilities of DSM. Sensitivity analysis of the NPI process to probability of rework is also performed. These models can be used to simulate the process as well as be modified to test various process improvements.

7.2.1 Simulation

One DSM feature that is helpful in this case study is the simulation capability of DSM. However, in order to simulate the process, several factors in addition to task dependencies are required.

First, the duration of tasks must be determined. As Cisco did not have this information readily available, this was an opportune time for stakeholders to come to agreement on a "ballpark" estimation of task duration. This was done using a triangular distribution with a Best, Worst, and Most Likely duration determined for each task.

Second, the probability of rework is a critical factor. This is only relevant for tasks that are dependent. For example, if Figure 15 is the original matrix and Figure 16 is the probability rework matrix, only in the places where there are dependency marks in Figure 15, will there be probabilities of rework in Figure 16. The probability of rework is the probability that one task will cause another to be reworked. For example, if "A" is reworked, there is a 20% chance that "B" will be reworked and a 10% chance that "C" will be reworked.

		Providing information						
	Name	A	B	C	D	E	F	G
Receiving information	A			1		1		
	B	1						
	C	1						
	D							1
	E			1				
	F							
	G			1		1	1	

Figure 15: Original Matrix

		Providing information						
	Name	A	B	C	D	E	F	G
Receiving information	A			.6		.7		
	B	.2						
	C	.1						
	D							.9
	E			.3				
	F							
	G			.2		.5	.4	

If A is reworked, there is a 20% chance that B will also be reworked

Figure 16: Probability rework matrix

Second, the impact of the rework is examined. This is the percentage of the task that is performed if the task is reworked. In the example in Figure 17, if C causes A to be reworked, then 80% of A will have to be reworked. If A is reworked, 50% of C will have to be reworked.

		Providing information						
	Name	A	B	C	D	E	F	G
Receiving information	A			.8		.4		
	B	.3						
	C	.5						
	D							.7
	E			.2				
	F							
	G			.1		.2	.9	

Figure 17: Rework impact matrix

Learning Curve is another important part of the puzzle. This refers to the percentage of the duration of the task the second time. For example, a learning curve of 90% means that 90% of the task must be done the second time and 81% the third time (90% of the time of the second task).

With these factors known, the model can be simulated to estimate the time the entire process takes. This functionality can also be utilized to simulate potential process improvements.

7.2.2 Sensitivity Analysis

The DSM also has capability to run sensitivity analysis to understand which tasks have the most impact on process time and standard deviation. The probability of rework was the lever used.[10] By modifying an Excel macro that Cory Welch developed for his 2001 LFM thesis, sensitivity analysis was performed on the probability of rework. This meant that it was possible to simulate the process times given a reduction in the probability of rework. Put another way, if we tighten or loosen the coupling of tasks, how does this affect the process?

After performing sensitivity analysis, it was then possible to create a Pareto analysis of the tasks that had the most impact. Not surprisingly, tasks that had many dependency marks seemed to have the most impacts. In this particular DSM, the BOM Change task seemed to have the most impact. This intuitively makes sense; changing parts in the middle of the process not only drives wait time for the part, but also possible design and testing changes. Figure 18 is a sample of the output of the model when sensitivity analysis is run. The baseline mean cycle time and the baseline standard deviation are the values when running the simulation based on the original inputs for task duration, probability of rework, and amount of task rework. The new mean and standard deviation are based on the original input for task duration and amount of task rework, but a reduced probability of rework. This means that in the probably of rework matrix, all values in a specific task's row are reduced by the "percent reduction." The "Percent Reduction" box records what this amount is. In each row, the task is listed along with the new cycle time mean and standard deviation based on the new probability of rework. It also lists the percentage cycle time mean and standard deviation.

	Baseline Mean	Baseline Std Dev	Percent Reduction	Runs per Simulation
	377	168.6533	0.05	1000
Task	New Mean	New Std Dev	% Mean Change	% Std Dev Change
product requirements	379	167	0%	-1%
schedule	375	171	-1%	1%
proto allocation	386	169	2%	0%
risk rating	334	156	-11%	-8%
BOM Changes	105	13	-72%	-92%
parts qualification	266	126	-29%	-25%
mfg plan	384	171	2%	2%
DFM feedback pre proto	372	161	-1%	-4%
DFM feedback mitigation pre proto	375	167	-1%	-1%
valor run and results	368	165	-2%	-2%
schematic	379	171	0%	2%
diag readiness	372	167	-2%	-1%
system specification	383	170	2%	1%
functional test script readiness	365	163	-3%	-4%
mechanical tooling strategy	382	168	1%	0%
Fab out custom mechanicals	376	167	0%	-1%

Figure 18: Sample output sensitivity analysis

In this example, we can see that the most significant changes occur when reducing the probability rework of BOM Changes and doing Parts Qualification. This makes sense as changes to the BOM change a number of other tasks including various tests and sometimes the design and layout of products.

Figure 19 shows a Pareto analysis of the mean sensitivity and standard deviation sensitivity to the probability of rework. BOM Changes are found to be the most significant.

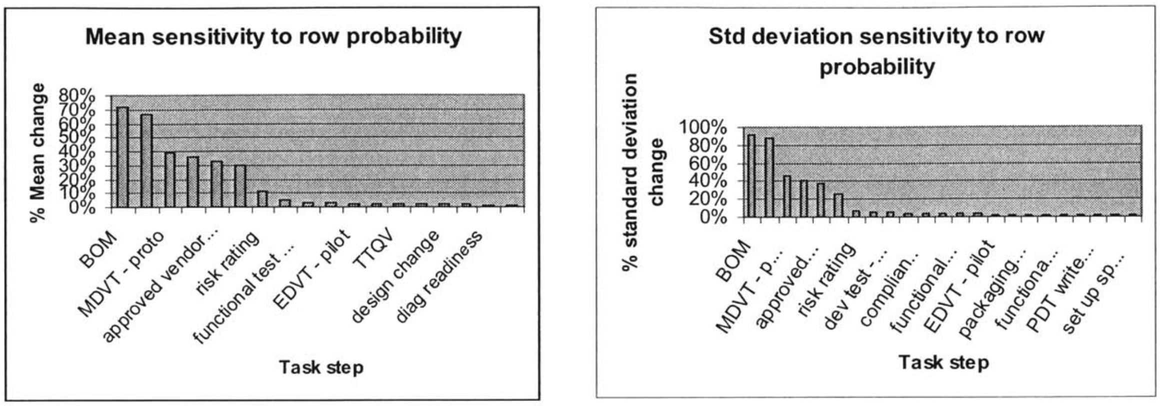


Figure 19: Pareto analysis of sensitivity to probability of rework

7.2.4 A Tipping Point?

After studying the sensitivity of the process to the probability of rework, further research was done to see if there was a "tipping point" in which a small change in the probability of rework caused by BOM Changes could greatly affect the process. This is helpful in understanding if there is a point of diminishing returns in certain areas of improvement. The model suggests that at a certain point, the amount of rework "tips the system" such that larger blocks of the system are reworked causing a greater cycle time of the process. In Figures 20 and 21, the graphs demonstrate that at approximately 30% likelihood of rework due to BOM Changes, the sensitivity of both the mean cycle time and standard deviation increases significantly.

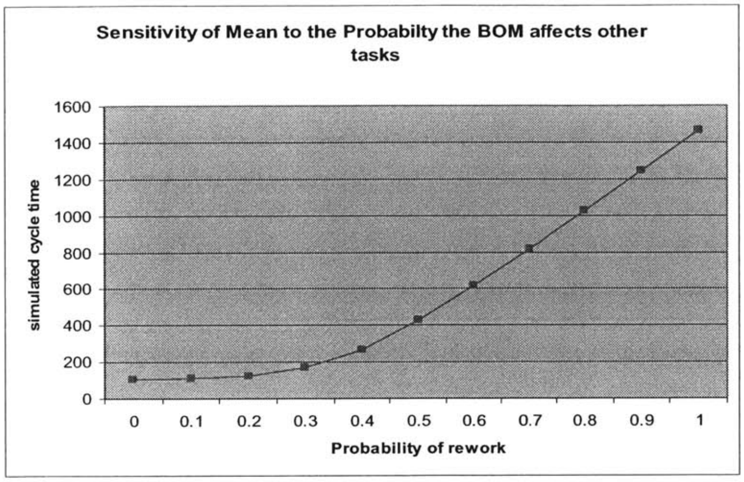


Figure 20: A Tipping Point, sensitivity of mean cycle time

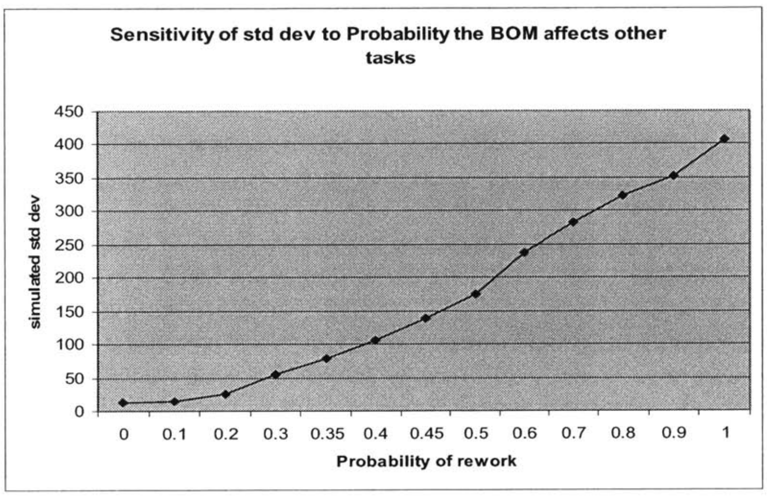


Figure 21: A Tipping Point, sensitivity of standard deviation

While this is just an example, running this analysis on "high value targets" can streamline the process improvement by giving the team a guideline for when further improvements may not be as useful.

7.3 Limitations of the DSM Models

The models in these two case studies make several assumptions in order to simplify the problem.

First, the models in the case studies assume that there are enough resources allocated. In real situations, lack of resources is frequently discussed as a limiter. While this is an available feature in the DSM package, it was left out of this model for simplicity. Figure

22 represents the Wheelwright and Clark (1992) research suggesting that productivity peaks at around 2 projects per engineer.

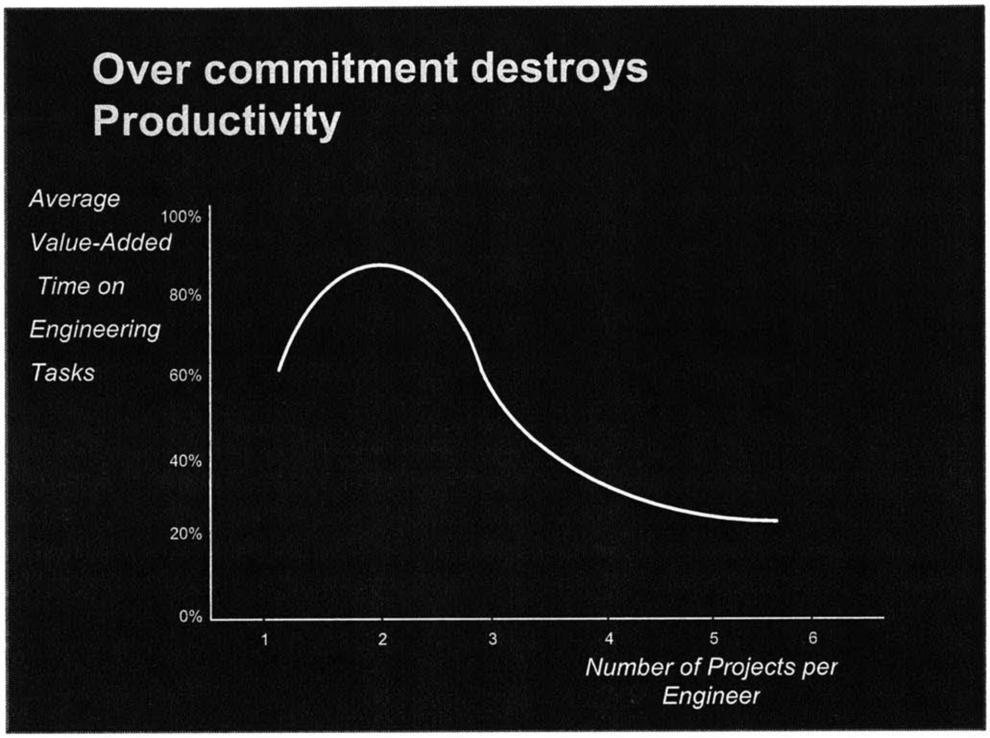


Figure 22: Effects of Over commitment[11]

Second, these models do not take into account product complexity or where the product stands in the product lifecycle. For example, the product duration and probabilities of rework are the de facto measures of product complexity and because the data is not readily available at Cisco, these were determined in an ad hoc manner during the case studies. In order to address this issue, the task times were adjusted according to product complexity; more mature products had reduced task times.

Similarly, each model only addresses one specific product even though it could be modified for any additional products. For products at varying places in the product lifecycle, the model does not take into account the product portfolio in which the product lies. While product differentiation often is a competitive advantage, a large variety of products in the product portfolio can divert resources. Companies tend to spend the least R&D money on incremental upgrades to derivative products. A lack of funding for certain projects can adversely affect its cycle time. When doing a DSM analysis, it is important to keep this in mind and create different DSMs according to the need.

This chapter discussed a second DSM case study. This time, the entire NPI process was studied. Once the matrix was partitioned, it became clear that there was one significant block in the process- the prototype phase. Similar to the first case study, a major highlight was the interconnectedness of the various groups inside Cisco. This was a clear signal that for the Mx Team to be successful, it would have to engage with groups outside of manufacturing and gain their support.

To better understand the prototype block, simulation and sensitivity analysis were run on the DSM. To do this, additional information was needed including the probability of rework, the amount of rework, and the learning curve. The analysis found that changing the probability of rework for certain tasks, such as BOM Changes, had a great effect on the simulated mean cycle time and standard deviation. These high value targets are something that manufacturing can begin to focus on when approaching BUs for help. The model also suggested that there might be a "tipping point," a place where there might be diminishing returns for improvements.

This chapter also discussed some of the limitations of the models built for the case studies. Some of these, such as not taking into account resource considerations, were omitted for simplicity but are available features of the DSM software package. Others such as not accounting for product complexity had to be adjusted through other variables in the model.

One of the main takeaways from Chapters 6 and 7 is the interactions between different groups in the organization. Chapter 8 delves more deeply into the organizational challenges and offers a few new concepts with which to think about organizations.

8.0 Putting the DSM into practice and use

One of the largest challenges of any improvement project is making it a sustaining part of the product development culture. However, there are many organizational challenges that can make this difficult. In the next sections, some of these challenges are addressed.

8.1 Organizational Challenges

Because the NPI process is so interlinked, the linkage between organizations is also of interest. In particular, this section address the role of the Mx NPI team as well as underlying issues about culture, social networks, and group communication.

8.1.1 The Mx NPI Team: Outsider-Insiders

The Mx NPI team is a small group working within a specific process role inside the Manufacturing Excellence Initiative. In her book, True Change,[12] Janice Klein describes a concept of Outsider-Insider, employees who are either outsiders to the business process but who can provide compelling 'inside' viewpoints, or insiders who deliberately look outside the business problem for advice and inspiration.[13] The Mx NPI team fits the description of Outsider-Insiders. While all members are trained in six sigma/DMAIC methods, the levels of expertise in DMAIC varies with some members having gone through large transformations and others much more localized ones. These true outsiders provide fresh insights through their work in different industries and apply the frameworks to Cisco. The others on the team have been at Cisco for several years and have worked through the system. Seeing that there were changes to be made they consciously took roles outside the system where they can focus on improvements.

Even though this team has an excellent ability to generate workable ideas for change, there are challenges because the group's incentives do not completely line up with those who are working inside the process. For example, throughout the project, several mentioned that the incentive structures for the NPI Process seemed to put more emphasis on firefighting and making the "diving catch" rather than on methodical and organized projects that proceeded smoothly. By contrast the Mx team is measured on its success in driving improvements that create a more methodical and organized process.

One issue is that of "pulling change." Klein likens pulling change to pulling inventory (as opposed to pushing it). The theory behind pull inventory is that a workstation will only produce when a customer needs it. Analogously, the idea here is to figure out what challenges the customer is facing that can not be worked out internally and work towards a change. Pushing change often results in wasted effort and frustration because the customers do not see why an external group is involved. In Cisco's case, strong market position and lack of a burning platform prevented some of this "pull" for change. In addition because of its product oriented structure, a pull needs to come from several of the stakeholders. For example, a pull from manufacturing can not make up for lack of a pull from the Business Unit (BU). In this case, manufacturing would need to show the pull for change to the BU.

8.2.2 Experience-based organizations versus technocratic

In Klein's view, there are generally two extremes of organizations: technocratic and experience-based with most companies lying somewhere in the middle. Experience-based firms are those that tend to use age, seniority, and company longevity as a basis for legitimacy and credibility.

On the other hand, a technocratic organization is typically one where one must first build credibility with demonstrated technical competence. Data is king in these types of organizations. Cisco does not lie on either extreme but does tend to exhibit tendencies of a technocratic firm. For example, during the course of this thesis, many attempts were made to collect data across the varied product lines but because there was no easily searchable data infrastructure, the lack of data sometimes served as a roadblock to progress. In additional, because each Business Unit tended to collect data differently, the wide array of products made it difficult to create a "level playing field" in which to compare situations. This created a situation where the data infrastructure was not set up to evaluate the success of programs and also not able to help create an impetus for change. While not the only reason, the lack of data helped to prevent the creation of a very strong case for change.

The DSM helps to remedy this by providing a blueprint for a data infrastructure. For example, in order to run a robust DSM simulation, task duration, rework probability, rework impact, task overlap, and learning curve parameters need to be known. Building a data infrastructure to record these variables would help to provide data for future process improvements.

8.2.3 Relationship between Design and Manufacturing – Social Networks

At Cisco, much of the coordination and interaction is virtual. For example, many of the key players are not co-located in the same buildings and many of the coordination meetings are done remotely on a conference call. The contract manufacturers are often in other states or even countries.

Eppinger (2001) gives us the idea of communication drivers and communication barriers.[14] Communication drivers are those that motivate information transfer between interacting team members while communication barriers are factors that hinder the process of exchanging information. The strength of task interdependence determines the extent of communication coordination needed. There are several factors that drive communication including organizational ties or affiliation and the interdependence of tasks. Factors that are communication barriers include physical distance, non-overlapping working time and cultural/language differences. Using this idea, we can see that Cisco has some serious communication barriers to overcome.

Similarly, Batallas and Yassine (2006) discuss three types of communication in product development organizations: [15]

1) Coordination-type: refers to technical information exchange (e.g., parameters) necessary to integrate final product's assembly.

2) Knowledge-type (innovation): refers to new knowledge created and shared during a new PD process.

3) Affirmative: team members and managers communicate for motivation and inspiration needs.

At Cisco, this "virtual" communication makes it more difficult to engage in all three types of communication. Using a DSM does not help to address the Affirmative communication but does help facilitate Coordination-type and Knowledge-type communications.

This chapter discussed the organizational challenges that Cisco faces. First, the Mx team is a talented group of individuals who work passionately for change. However, their primary role needs to be to "pull" change. That is, find out what the people doing the work feel are the major problems and attack those issues. This means getting to know all the major stakeholders, not just those in the manufacturing organization.

Next, the difference between technocratic and experience-based firms was discussed. Cisco tends to be a technocratic firm where data reigns. Ironically though, Cisco's data infrastructure is not as searchable as it could be. Because Cisco is currently doing well in the market and there is not burning platform for change, it is difficult to make a strong case for change without data.

Finally, social networks were looked at. At Cisco, much of the work that gets done is done virtually. This virtual communication can make it difficult to engage in coordination-type, knowledge-type, and affirmative-type communications.

In the next chapter, the learning from the case studies along with ideas from Chapter 8 are used to understand some of the themes and find ideas for improvement.

9.0 Themes and Lessons from the Case Studies

By using the DSM to study the NPI process at Cisco, we uncovered several themes. Even though most of these are not directly related to DSM, the use of the DSM highlighted these issues at Cisco. This Chapter discusses these themes and how Cisco can improve.

9.1 Provide the infrastructure for change

In True Change, Janice Klein points out that Outsider-Insiders and a pull for change are not enough. The last piece of the puzzle is providing a support infrastructure in which change capability is nurtured. This includes not only management support for change management initiatives but also support for the right data sources.[16]

One challenge in gathering support and champions lies in the lack of searchable data at Cisco. In the absence of a robust data infrastructure, the data collection becomes somewhat ad hoc with data residing in forms such as meeting minutes, emails, and word documents as opposed to searchable databases. This requires more time and effort to gather the necessary data. While this may be passable in actually getting the new products out the door, it does not help to optimize processes across the company or help Cisco to learn from previous mistakes. Cisco's product oriented organization can also make it difficult to distill learning across the product groups.

Not only is data a concern but an organizational infrastructure must also be built. This means setting up an organization in which the NPI can be studied and improved upon with inputs from all stakeholders. There are limits to what the Manufacturing organization can do alone.

9.2 Utilize Outsider-Insiders

The DSM requires stakeholder buy in and cooperation. In both case studies, a few "champions" were able to facilitate the DSM build and build support. In general, the best champions were the Manufacturing Test Engineering Manager and the Product

Operations Director for the Internet Systems Business Unit. Having someone who controlled resources and understood how studying the process could lead to greater process improvements was helpful in gathering other resources. The full time members of the Mx team were also helpful to find the right resources and to begin getting traction. They were also valuable as resources that knew something about several things and could get the ball rolling.

This suggests that one of the great values of the full time Mx Team is in gathering support of key champions and convincing them of the value of process improvements. These champions do not lie solely in manufacturing though. In order to be successful the MX Team will need to build bridges to the Business Units inside Cisco. As seen in the second case study, the prototype phase of the NPI process is the most intensive part of the process. This phase includes a large number of stakeholders across several different organizations. Robust process improvements would include all stakeholders in the process.

9.3 Utilize virtual teams to the fullest

Moenaert *et al.* notes that the creation of a "core team," which other teams find as a connection point, enhances a project's information exchange.[17] While Cisco does have these "core teams," the virtual nature of the teams can diminish the positive effects. Co-location would help the teams with "face time" and build the social networks that can help drive information flow in the product development process.

However, co-location may not be the right solution. With large functional teams, it can be difficult to figure out how to co-locate since many team members work on more than one product. A different option is to better utilize the virtual teams that are in place. For example, while the Cisco NPI teams use systems such as email, standard formats for reviews and gates, web databases and shared drives to communicate virtually, there is still work that can be done to standardize the processes across the Business Units. One option is to consider is a wiki system or other web-based interactive technology.

However, technology should not be considered to be the solution, but rather a part of a continually improving process to get the best performance out of the team.

9.4 Spend more time upfront – the 80/20 Rule

As show in the Figure below,[18] most firms spend most of their management time in the later stages of the process when the ability to influence the outcome is the smallest.

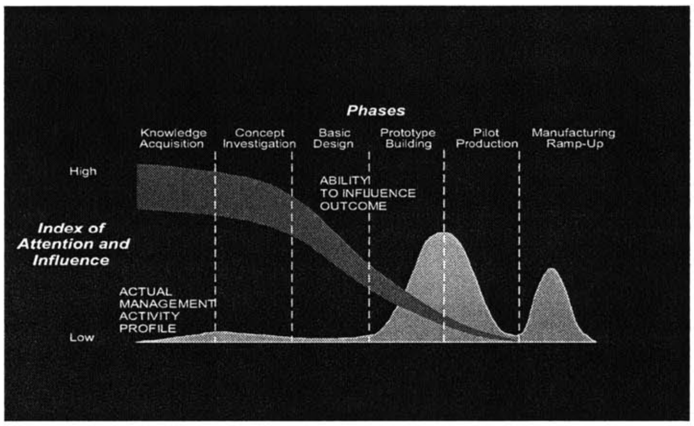


Figure 23: Managerial attention and influence versus process phase

According to Fine (1998), "As a rule of thumb, many managers assume that as much as 80% of lifecycle system design and manufacturing costs are fixed by decisions made during the product development process. This decision making occurs often within the first 20% of the design and manufacturing life cycle of the product."[19]

Spending more time upfront requires management support and stakeholder buy in, which is counter to the idea of fire fighting.

9.4.1 Stop Giving Incentives for Fire Fighting

The second case study showed that early engagement can help to reduce cycle time. Specifically, that more time spent upfront on the early stages of design work can reduce rework later on. Repenning (2001)[20] argues that firefighting can be a self-reinforcing phenomenon. Once it starts on one project, it can spread to other projects within the product portfolio. According to Repenning, fire fighting execution mode is characterized by little attention to the early phases of the development cycle and a consequent focus on rework can become the *de facto* development process. Inadequate attention to the early portions of the development process creates a self-reinforcing cycle that traps the system in a permanent state of unbalanced and undesirable resource allocation. Cisco must see to it that its incentives are not aligned to reinforce firefighting.

This is easier said than done. A "diving catch" is often more tangible and visible to management than putting in the work upfront to avoid that "diving catch." There is also the temptation to continue to focus on today's issues at the expense of the entire system. In addition, because incentives should be consistent across groups, gaining buy-in across the company will be difficult.

Not only does Cisco need to carefully discuss what it want to value as a company, it need to determine an equitable way to reward employees to drive the right behaviors. In a perfect world, the wrong behaviors would be caught and stopped immediately. However, in complex processes with tight deadlines and market pressure, this is tough to do.

9.4.2 Simulate First

From the DSM model, we found that a reduction in rework probability of BOM changes could significantly reduce the cycle time of the entire process. BOM changes occur for a number of reasons including parts shortages, design changes, and the way components interact with other components.

Utilizing tools and technology up front to simulate how a product and its components will behave before it even begins prototype manufacturing is key in cutting down time downstream. Simulation can identify weak spots in the design that can be easily modified in a computer-aided design program, rather than at the end of the product development cycle when a change translates into expensive modification and re-engineering. The investment in such tools and technology will prevent costly delays later.[21]

In this chapter, several ideas were put forth as ways that Cisco could improve its NPI process. First, Cisco needs to provide an infrastructure for change. This means that management must be on board with process improvements but also the necessary ingredients must also be available. This includes both the data infrastructure and the larger organizational infrastructure. Next, Cisco should utilize Outsider-Insiders. These are people who work inside the process but are still champions for change. The Mx Team should continue to actively search these people out and utilize their expertise. Third, Cisco should examine its virtual teams and understand how to mitigate the communication concerns. Co-location and use of additional technology are options but neither is necessarily correct. Lastly, as the 80/20 rules goes, in product development, 80% of time and resources is spent on items agreed upon within the first 20% of the product development lifecycle. This is powerful incentive to plan ahead and engage early in the process.

10.0 Conclusion

In this thesis, I have done two case studies using the Design Structure Matrix (DSM). In both, the DSM was an excellent tool in highlighting the barriers to a successful launch.

In the first case study, the manufacturing test development process was studied. The DSM was a useful tool to start the dialogue between the Manufacturing Test Engineering and Software Diagnostics groups who, while working closely on a daily basis, do not have many opportunities to step back and analyze their process. In developing a DSM of the process, they were better able to see the dependencies between the groups and begin to improve. In that case, they found that doing more work defining the product and customer requirements as well as the engineering design requirements were important in reducing delays due to unplanned rework.

In the second case study, the entire NPI process was studied. The DSM showed the Mx team how crucial it was to partner with engineering and design in order to improve the cycle time of the NPI process. It also showed that the cycle time and variance in the process is very sensitive to changes in the Bill of Materials (BOM). The model also suggested that there was a tipping point where reducing dependency between BOM changes and other tasks could greatly improve the cycle time and variance of the process.

In addition, I have shown that for Cisco, there are serious organizational barriers to a successful process improvement effort. Because Cisco is currently in excellent market position and there is no burning platform for change, there are powerful incentives to continue the status quo. Finding a pull for change is also hampered by the lack of a robust data infrastructure. However, in order to remain on top, Cisco must not rest on its laurels. It must continue to innovate processes in addition to products.

Interestingly, the Manufacturing Excellence Initiative in NPI is a misnomer. New Product Introduction is not just the job of manufacturing but is highly integrated between such groups as marketing, design, and engineering. If the Mx Initiative in NPI is to fully meet its potential, all of these groups must fully realize this.

Appendix A: Manufacturing Test Development Partitioned DSM

Task Name	evel		1	4	5	6	7	8	2	3	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
understand product requirements (PRD	1	1																																			
understand HW specs/schematic (how	2	4	1		1																																
understand ASICS -	2	5	1	1		1	1	1																													
understand previous	2	6	1	1	1		1	1																													
understand central test	2	7	1	1	1	1		1																													
understand EDVT test	2	8	1	1	1	1	1																														
understand schedule	2	2																																			
understand priority	2	3																																			
understand current	3	9						1	1	1																											
understand the diag	4	10	1	1	1	1	1	1																													
determine technical mfg test requirements	4	11	1	1	1	1	1	1			1																										
write test plan	5	12							1	1			1				1																				
communication needs	5	13							1	1		1	1	1		1																					
test plan review and	5	14							1	1	1	1	1	1	1																						
budgeting for new	5	15							1	1			1			1																					
determine proto	5	16							1	1			1																								
receive boards	6	17																1											1								
write diags	6	18							1	1		1		1	1				1																		
test functionality of	6	19																		1			1														
understand diag	6	20							1			1		1	1					1																	
write scripts	6	21												1			1		1	1		1					1										
find and report issues	6	22										1		1	1					1	1		1														
diag eng works on bugs	6	23																			1	1	1	1													
diag eng write and test	6	24																							1												
understand new diag	6	25													1											1											
TE tweaks based on	6	26																					1	1		1	1										
design and layout	6	27	1	1	1			1							1													1									
scripts ready to go	7	28																										1								1	1
port script to CM	7	29																												1							
send test package to	7	30																													1						
CM build	7	31																														1					
CM tests funcationality	7	32																															1		1		
CM uses scripts	7	33																																1		1	1
CM finds and reports	7	34																																	1		
tweak scripts for new	7	35																																		1	

Block1

Block2

Block3

Appendix B: NPI Process Partitioned DSM

		product requirements	schedule	proto allocation	risk rating	BOM	parts qualification	mfg plan	DFM feedback pre proto	DFM feedback mitigation pr	valor run and results	schematic	diag readiness	system specification	mechanical tooling strategy	fabout custom mechanicals	second sourcing	PCB layout	EDVT - proto	MDVT - proto	dev test - proto	approved vendor list	RDT	CM builds proto	bring up proto	CAD work	fabout package CAD PCA n	DFM post proto feedback	design change	set up spares depot	functional EFI and AFD	HALT test	SW verification testing	HW readiness	calculate MTBF	DFM post proto mitigation	SW readiness	functional test script readine	functional test strategy	new test equipment determi	CM build pilot	new test equipment develop	PDT writes rules for orderab	compliance and safety test	turn on orderability in Oracle	packaging test	EDVT - pilot	MDVT - pilot	Dev test - pilot	FCS	TTQV
		1	2	3	4	5	6	7	8	9	10	11	12	13	15	16	19	20	21	22	23	24	27	29	30	39	41	44	50	42	25	26	35	37	40	45	36	14	17	18	33	34	38	28	31	32	47	48	49	43	46
product requiremer	1	1																											1																						
schedule	2	1	2																										1																						
proto allocation	3		1	3																																															
risk rating	4				4	1																																													
BOM	5	1				5	1							1			1		1	1	1	1						1																							
parts qualification	6					1	6										1																																		
mfg plan	7			1		1		7																																											
DFM feedback pre	8					1			8		1			1	1									1																											
DFM feedback miti	9								1	9				1	1									1																											
valor run and resul	10										10	1														1																									
schematic	11	1										11						1								1																									
diag readiness	12					1							12	1											1	1																									
system specificatio	13	1												13																																					
mechanical tooling	15													1	15																																				
fabout custom mec	16														1	16																																			
second sourcing	19				1	1											19					1																													
PCB layout	20					1								1				20																																	
EDVT - proto	21			1		1							1	1	1				21		1				1																										
MDVT - proto	22			1		1								1	1					22	1				1																										
dev test - proto	23			1		1								1							23				1																										
approved vendor li	24				1	1																24																													
RDT	27					1							1	1									27						1																						
CM builds proto	29							1		1	1				1	2								29			1		1																						
bring up proto	30																							1	30		1																								
CAD work	39													1				1								39																									
fabout package CA	41																	1									41																								
DFM post proto fee	44																							1				44																							
design change	50																		1	1	1		1					1	50																						
set up spares depo	42																													42																					
functional EFI and	25					1							1												1						25																				
HALT test	26													1																		26																			
SW verification tes	35																								1								35																		
HW readiness	37																	1		1														37																	
calculate MTBF	40					1																													40																
DFM post proto mit	45																											1								45															
SW readiness	36																																1				36														
functional test scrip	14												1	1																								14	1												
functional test strat	17	1	1																1																				17				1								
new test equipmen	18																																						1	18											
CM build pilot	33							1	1	1			1			1					1				1		1				1			1	1		1	1			33	1									
new test equipmen	34																																							1		34									
PDT writes rules fo	38	1																																							1		38								
compliance and sa	28																																								1			28							
turn on orderability	31																																										1		31						
packaging test	32	1																		1																					1					32					
EDVT - pilot	47																		1																						1						47				
MDVT - pilot	48																			1																					1							48			
Dev test - pilot	49																				1																				1								49		
FCS	43																						2							1											1			1		1	1	1	1	43	
TTQV	46																						1									1																		1	46

Endnotes

[1] Innovation at the Speed of Information, Eppinger, HBR, January 2001
[2] Q&A with Geoffrey Moore, Cisco Executive Thought Leadership Quarterly, June 16, 2003
[3] John Chambers, Cisco Executive Thought Leadership Quarterly, June 16, 2003
[4] Is the Make/Buy Decision a Core Competence?, Charles Fine, Daniel Whitney, Feb. 1996
[5] Clockspeed, Charles H. Fine, 1998, Perseus Books
[6] Clockspeed, Charles H. Fine, 1998, Perseus Books, p.71
[7] Dynamic Capabilities and Strategic Management., By: Teece, David J., Pisano, Gary, Shuen, Amy, Strategic Management Journal,1997
[8] Misalignment of Product Architecture and Organizational Structure in complex Product Development, Manuel Sosa, Steven Eppinger, Craig Rowles, Management Science, Vol 50, No. 12, Dec 2004, pp1674-1699
[9] Eppinger, Innovation at the Speed of Information
[10] Cory Welch, LFM Thesis, 2001
[11] Steven Wheelwright and Kim Clark, Revolutionizing Product Development, New York : Free Press ; Toronto : Maxwell Macmillan Canada ; New York : Maxwell Macmillan International, c1992
[12] True Change: How outsiders on the inside get things done in organizations, Janice Klein, Jossey-Bass, 2004
[13] True Change, : How outsiders on the inside get things done in organizations, Janice Klein, Jossey-Bass, 2004
[14] Factors that Influence Technical Communication in Distributed Product Development, Sosa, Eppinger, Pich, McKendrick, Stout, IEEE Transactions of Engineering Management, Vol 49,, No.1, Feb 2002, p45-58
[15] Information Leaders in Product Development Organizational Networks: Social Network Analysis of the Design Structure Matrix, Diego Andres Batallas and Ali A. Yassine, IEEE Transactions on Engineering Management, Vol. 53, No. 4, November 2006
[16] True Change: How outsiders on the inside get things done in organizations, Janice Klein, Jossey-Bass, 2004
[17] R. K. Moenaert, F. Caeldries, A. Lievens, and E. Wauters, "Communication flows in international product innovation teams," Journal of Product Innovation Management, vol. 17, pp. 360–377, 2000.
[18] Wheelwright and Clark, Redefining Product Development
[19] Clockspeed, p177
[20] Understanding Firefighting in Product Development, Nelson P. Repenning, Journal of Product Innovation Management, March 2001
[21]Manufacturing Engineering. Dearborn: Dec 2006. Vol. 137, Iss. 6; pg. 37