Context Characterization for Synthesis of Process Architectures

By

Christian LaFon

Master of Science – Electrical Engineering, *Telecommunications* (2001)
The Johns Hopkins University

Bachelor of Science – Electrical Engineering (1996)
Rutgers University

SUBMITTED TO THE SYSTEM DESIGN AND MANAGEMENT PROGRAM IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN ENGINEERING AND MANAGEMENT
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2008

Signature of Author.. 5/6/08
Christian LaFon
System Design and Management Program
June 6, 2008

Certified by.....................................
Dr. Ricardo Valerdi
Thesis Supervisor
Lean Advancement Initiative
Engineering Systems Division

Certified by.................................
Patrick Hale
Director
System Design & Management Program

Context Characterization for Synthesis of Process Architectures

By

Christian LaFon

SUBMITTED TO THE SYSTEM DESIGN AND MANAGEMENT PROGRAM IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN ENGINEERING AND MANAGEMENT

June 2008

ABSTRACT
Analysis steps are proposed as an aid for establishing Lean Product Development (LPD) activities in an organization. The proposal is offered as an aid to engineering managers and process designers for coping with the unique challenges of implementing processes from their inception – for example, at a new enterprise. As such, the thesis focuses on the *creation* of LPD, as opposed to traditional Lean *improvement* activities which benefit from the perspective of hindsight of a legacy process. Without established product development processes to improve upon, the implementation of product development activities at a new venture relies on the use of foresight to instance a LPD environment in new organizations. Therefore, the paper stresses stakeholder value delivery within the specific context that an enterprise operates and competes. A generic framework for context characterization is proposed and discussed. The framework is then evaluated for its usefulness in process design activities. The analysis steps are based on literature review and case study interviews.

The proposed analysis steps include:

- a comprehensive definition of the business *context* in which the enterprise operates and competes,
- a statement of *goals* and *objectives* for the product development organization based on this context, and,
- a determination of appropriate *behaviors* to meet these goals.

Traditional Lean research has typically been approached from a large-scale, complex systems, for-profit perspective. Unique insights are gained from the perspective of small, privately funded, new ventures. The benefits include foresight-only value objectives for product development (process creation) and uniqueness of context (i.e. resource limited, mindshare-driven). The analysis method was validated by examining process design case studies within three contexts: large-scale aerospace, industrial process monitoring, and high-technology start-up.

Thesis Advisor: Ricardo Valerdi

Acknowledgements

My thesis journey could not have been completed without a support network of friends and family for which I am sincerely grateful.

To my wife, Patti, for her tireless support and understanding. Words cannot express my optimism for our future and for our young family.

To my parents, who provided love and opportunity such that few obstacles have stood in the way of dreams. To my sister, Michele, for the loyalty and consideration she has expressed to me my entire life.

To Dave, for inspiration and friendship (and for providing shelter during my stay at MIT).

To Spiros, for polite and successful collaboration.

Special thanks to my thesis advisor, Ricardo Valerdi, for his guidance, patience, and perseverance over a long process.

Table of Contents

Table of Figures

Table of Tables

1. Introduction

This thesis investigates the role of the process architect and the challenges of process design in a product development organization. The strategic role of product development is emphasized, with a focus on stakeholder value delivery within the *context* of the development effort: business environment, enterprise capabilities and resources, team composition, and product architecture. The context of the development effort within its marketplace informs goal setting for the product development organization, management, and the team. The paper explores the utility of Lean Product Development as an aid for process creation, decision making, and continuous improvement.

1.1. Motivation

Superior processes lead to competitive advantages.

Business leaders have an interest in process design because they recognize that *the ways* in which organizations go about delivering products and services are potentially as important as *the characteristics of* the products and services delivered. That certain companies can consistently reach their market either with higher quality, in less time, or while incurring fewer costs than competitors is the prime motivation for this work.

The notion that competitive advantages are achieved through process innovation has been demonstrated in Lean manufacturing organizations and others, in a variety of industries. Toyota and Dell, for example, have created process advantages over their competitors through maximization of production efficiencies. Toyota's revolutionary production processes have led to remarkably high quality standards in relation to their competitors (Liker, 2004). In the case of Dell, production and supply chain strategies have helped to maximize operating margin in an industry where profits are notoriously difficult to achieve (Kraemer & Dedrick, 2002). Both cases are excellent examples of companies that have used *process innovation* to differentiate themselves in partially, or even fully commoditized markets where *product differentiation* is otherwise difficult to achieve.

These examples highlight the potential hidden value of enterprise processes. In the view of Barney (1996), "a firm's resources and capabilities are valuable if; they reduce a firm's costs or increase its revenues, or if, they enable the firm to respond to environmental threats or opportunities." Further, resources result in sustained competitive advantages if they are *valuable*, *rare*, and *inimitable*, and if the *organization* (Barney, 1996, p. 162) of the firm is such that these resources can be exploited. In both the cases of Toyota and Dell, innovative production processes have emerged as differentiating, competitive resources. Other factors being equal, companies like Toyota and Dell retain competitive advantages over time because of operational efficiencies executed along the supply chain and on the production floor.

It is motivating to imagine how certain advantages can be obtained through choices made when designing product development processes as well. The way in which a product development team is organized, their experience and motivation, and the infrastructure and tools at their disposal – are all critical factors which can affect a product's quality, cost, and time-to-market. The behaviors and activities of engineering leaders and development teams, within the environment provided by the enterprise, may result in competitive advantages, or in rare cases, disadvantages.

How appropriate are the processes employed to the task at hand?
Critical characteristics of product architecture contribute to define the appropriateness of culture and behavior of an organization. For example, Toyota's reputation and standard for reliability in its automobiles sets an enormous precedent for their manufacturing approach – the level of precision required of tools and machinery, and the extent of coverage of test cases. Critical product traits such as reliability are of primary consideration for determination of appropriate development behaviors and activities. In addition to product traits, other critical factors must be considered when deciding what methodologies are appropriate for a product development organization – the company's resources, market position, and customer relationships are examples.

These factors and others contribute to define the context in which the company operates, and must be comprehensively analyzed before decisions are made about the appropriate mode of operation for the product development team. The focus of this work is to uncover the predominant contextual factors to consider when determining appropriate behaviors for achieving strategic goals.

Process design is a difficult, but necessary task for technology companies.
Whether to improve processes and behaviors or to innovate them, the challenge of process design is a real and necessary task for technology companies. Process design decisions are necessary as a process must frequently be *incrementally improved*: adapted to changes in market conditions, changes in the products manufactured, or changes in customer demands. Occasionally, as in the case of Toyota and Dell, processes are *innovated*, or overhauled to radically new models. In certain circumstances processes are even architected "from scratch" – perhaps best exemplified at a new enterprise where the behaviors entire organizations are almost entirely undefined. In all cases, capabilities are required for the design of process architecture – technology companies must make determinations as to the environment provided, the organization of technical individuals, and how problem solving is implemented. Process design, and in particular the unique challenges of process creation, forms the main motivation for this thesis.

> *In the universe of process styles and approaches, how does one select the most appropriate behaviors when given the opportunity to improve, innovate, or define processes "from scratch"?*

Starting off on the right foot.
Processes, behaviors, and their participants – perhaps in place for decades – establish a culture that defines an engineering organization. As processes and behaviors define culture, process change equates to culture change. Implementing culture change can be a notoriously difficult proposition in many companies, as evidenced in Lean transformation literature (Bozdogan et al., 2000).

For new ventures, starting off on the right foot with regards to culture is critically important, and serves as a final motivation for this thesis. It is clearly best to establish the right approaches from the onset, as culture will be increasingly difficult to manipulate as time passes.

1.2. Problem Statement

Organizations must continually adapt their processes and behaviors to keep pace with the dynamics of changing market conditions. When the performance of a particular process is insufficient for the goals at hand, the process must be evaluated and transformed. Therefore, engineering leaders are in need of tools and methods to aid in process evaluation and design.

Provided that existing processes are relatively well matched to the dynamics of a particular market, these adaptations may require only incremental changes to existing processes, or process improvements. Occasionally, existing processes must be radically changed to match the unique demands of a particular market, or a product development organization may need new processes altogether. In these instances, process design decisions may require process innovations, or process creation.

In either case, technology companies must provide the appropriate environment, build a development team, and prescribe the right behaviors. In regards to environment, what infrastructure should be provided? In regards to individuals, what are their motivations, and experiences? In regards to the choreography of the development team, how are effective communication channels ensured?

Traditional Lean process initiatives are known to be useful for the former case described above – the case of incremental change, or process improvement. It can also be demonstrated that Lean is useful for the latter case, that of process creation. In support of this hypothesis, this thesis formulates an analysis methodology for determining product development goals and objectives within a specific context, and for applying Lean principles for process creation in fulfillment of those objectives in product development organizations.

1.3. Product Development Process Architecture and Corporate Strategy

The Lean philosophy promotes a strategic view of enterprise processes. From LAI's Transition-to-Lean Roadmap, Volume 2:

> "The idea to consider the Lean paradigm need not originate in the executive offices of the Enterprise. It can originate at any level in the organization. Ultimately, though, its consideration must be pursued as a major strategic issue for the Enterprise. In fact, to be successful, any effort to transform an Enterprise must be totally compatible with, and ideally an

outgrowth of, the Enterprise's strategic planning process." (Bozdogan et al., 2000)

The strategic role of the product development organization in an enterprise is to provide the right technology, at an appropriate price (Womack & Jones, 1996), and with the right timing to enable the enterprise to operate and compete effectively in a particular marketplace. In this manner, the product development organization is considered a tool that is employed by the enterprise to fulfill its strategic needs with regards to products and technology. Process should be designed around strategy – strategy drives process, and process drives organization (Kiraka & Manning, 2005). Furthermore, as the strategic needs of the organization change, so does the strategic role of the product development organization. In order to serve the strategic needs of the enterprise, the process should deliver value to all of the stakeholders of the product development process.

1.4. Product Development Process Architect

In many of today's technology companies, product development processes are typically implemented corporate-wide with a single formula for all current active projects in the enterprise. Product development processes are created by members of engineering leadership (i.e. engineering management, project engineering, project management) or by consultants brought on only temporarily to evaluate, critique, and restructure processes. Process redesign, if and when it occurs, is typically a landmark event with enormous cultural implications for the organization.

To emphasize and support the strategic importance of product development to technology companies, this thesis explores the notion of a Product Development Process Architect as a permanent, full-time organizational role in the high-technology enterprise. The role of the process architect is to advise, design and advocate for product development environments that evolve to deliver value to all process stakeholders, consequently satisfying the dynamic, strategic needs of the enterprise.

The product development process architect must be a holistic, systems thinker with broad and deep understanding of a variety of influential factors. First, the process architect must have market knowledge and business training. The process architect must have a comprehensive knowledge of the enterprise: its organization, supporting processes, and resources. The process architect must have comprehensive knowledge of the development staff: their personalities, work preferences, and their experience. Finally, the process architect has knowledge and experience with products, training in systems and architectures, and must maintain a working relationship with the *product architects* themselves. The process architect must communicate with the enterprise, its leadership, and development staff. The architect must have credibility with, and the respect of the development team, and must have the ability to persuade and influence behaviors, and indeed effect cultural change where necessary.

The role of the process architect should be a relatively independent role - the architect should not be a participant in the process, nor a strict representative of the customer or the enterprise – such that the role should not be biased too heavily by an individual stakeholder or by the current organization of the enterprise, business unit, or functional department. Still, the process architect must be a permanent member of the enterprise, (in lieu of a consultant, for example) due to the required level of intimacy with the organization and the staff.

1.4.1. Proposed Steps for Architecting Processes

As stated above, the role of the process architect is to design, advise, and advocate for the components and characteristics of a product development environment which deliver the strategic needs of the enterprise. The following procedure is proposed as an analysis methodology for determination of appropriate process architectures.

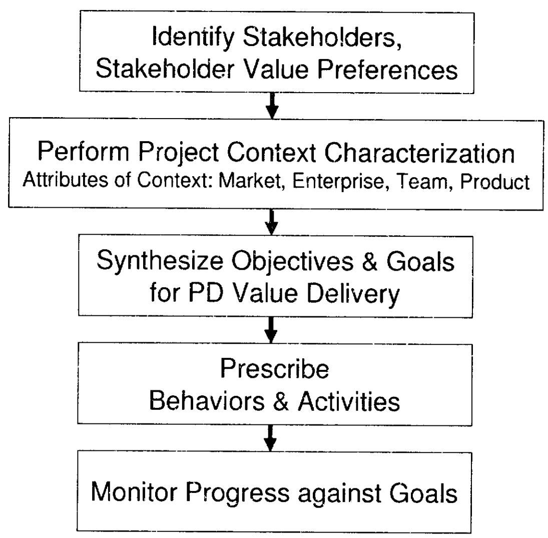


Figure 1.1 Architecting Processes for Value Delivery

This analysis procedure is explored throughout this thesis. Throughout the exploration, the role of the product development process architect is expanded upon in detail.

1.4.2. Stakeholder Perspectives

As stakeholders of the enterprise, the topic of product development process design and methodology is of interest to engineering managers, technical leaders, and participating engineers alike. Management strives to align processes with corporate goals. Technical leaders desire to organize teams tactically for delivering technology to the enterprise in fulfillment of business objectives. Process participants require the infrastructure and coordination necessary to meet quality, cost, and schedule constraints – all while maintaining a healthy work-life balance.

When considering the various stakeholders of the product development process, several interesting questions come to light. What do stakeholders from different industries value in common, what do they value differently? What characteristics of the product development process can be manipulated to maximize value for a particular stakeholder? What are the linkages between these value perspectives and modes, or styles, of behavior?

To answer these and similar questions, the process architect must begin by holistically considering the environment surrounding the development effort and its various stakeholders. Architecting a product development process with the objectives of enterprise stakeholders in mind requires a characterization of the *context* in which the enterprise, its processes, and its stakeholders exist.

1.4.3. Context

> "Process design is a management decision and must be made by considering several factors: size and complexity of project, whether or not we've done this before (precedentedness), ability of the team, schedule, etc." (Boehm and Turner, 2004)

The words of Boehm allude to some of the many components of context: product traits, traits of individual team members, competition, resources, access to customers, and experience. The implication of factoring the uniqueness of team and product attributes into the project context is that the process architect will likely prescribe *unique behaviors* and activities for *each of* the current active projects in the enterprise. In Chapter 3, a detailed analysis of these and other components of context are presented.

1.4.4. Product Development Goals and Objectives

Once captured and understood, how does the contextual environment translate into goals for the team? Broadly, the product development organization aims to deliver and ultimately maximize value to its stakeholders. Unfortunately, process goals as derived from stakeholder value delivery remain ambiguous without considering the contributions of the greater context of the development project. Process goals derived from the project context are explored in Chapter 4. The prime outcomes of this thesis are clear and concise goals for the development team and the process itself.

1.4.5. Value of Product Development

Product development processes are the behaviors, activities, and methodologies employed to achieve the goals and objectives of the product development organization. The value of product development (Chase, 2001) lies in its ability to meet or exceed these goals. While there are many types of behaviors employed by development teams today, the most appropriate behaviors for a particular team (within a particular context) are of particular interest. It is critical to architect and choreograph product development activities to *most efficiently achieve* strategic corporate goals. In Chapter 5, the use of Lean Product Development in translation of goals to behavior is explored.

In summary, the architecture of product development activities should only be created after a stakeholder value analysis is completed within an appropriate context. The contextual environment informs a suitable strategy for fulfillment of corporate goals, while the activities of the product development organization are the strategy put into action. Finally, the value of product development activities lies in the extent to which they deliver on the various strategic goals.

The following sections briefly review Lean process improvement initiatives and introduce the notion of Lean process creation, both of which complement the type of analysis described above.

1.5. Lean Improvement Initiatives

The Transition-to-Lean (TTL) Roadmap (Bozdogan et al., 2000) proposes an iterative transformation cycle where enterprise processes are transitioned towards Lean. Using the TTL Roadmap, an enterprise is encouraged to first determine strategic goals, and second to implement them via processes. The following figure presents the TTL Roadmap, which has been amended to highlight its relationship to enterprise processes and their transformation from "current state" to a Leaner, "future state".

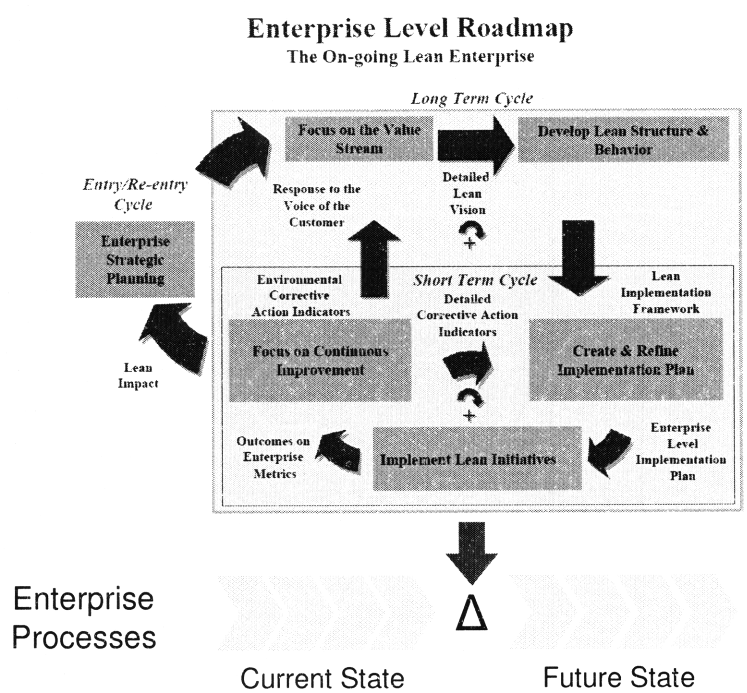


Figure 1.2 TTL Roadmap in Relation to Enterprise Processes (adapted from Bozdogan et al., 2000)

The iterations in the TTL Roadmap emphasize the spirit and philosophy of continuous improvement, a foundation of Lean initiatives. As revealed in the detail of the figure, each cycle of the transformation involves; an evaluation of the current strategic performance of the enterprise, a reevaluation of strategic goals, planning and preparation for process improvements, and finally implementation of process change.

The transformation cycle can be thought of as a combination of hindsight and foresight utilized by the enterprise to continuously improve processes. *Hindsight* is used to monitor the alignment of outcomes with goals, and *foresight* to make adjustments to behaviors to improve this alignment. Relatively abrupt adjustments are made in earlier stages, while perhaps only incremental adjustments are made in later stages when the Lean transformation matures and alignment improves.

1.6. Lean Process Creation

Continuous improvement initiatives, such as Transition-to-Lean, scrutinize enterprise processes and behaviors to identify and eliminate inefficiencies and wastes. As such, an improvement initiative *implies the existence of* the current state. As mentioned above, monitoring and observing the current state of processes requires and exploits the benefits of hindsight.

How can the TTL Roadmap be useful to process architects designing new processes? Process architects in new organizations, for example, do not have existing processes to incrementally improve. In a sense, creating, planning, and implementing strategic behaviors requires a reliance on foresight only.

The following figure presents the TTL Roadmap again, this time amended to highlight the unique challenges of process creation. In the figure, the benefits of hindsight are removed (grayed out). The process architect must rely solely on foresight – a comprehensive stakeholder value analysis, within a specific context – to aid in process design decisions. Additionally, emphasis is placed on creation of the *initial state*.

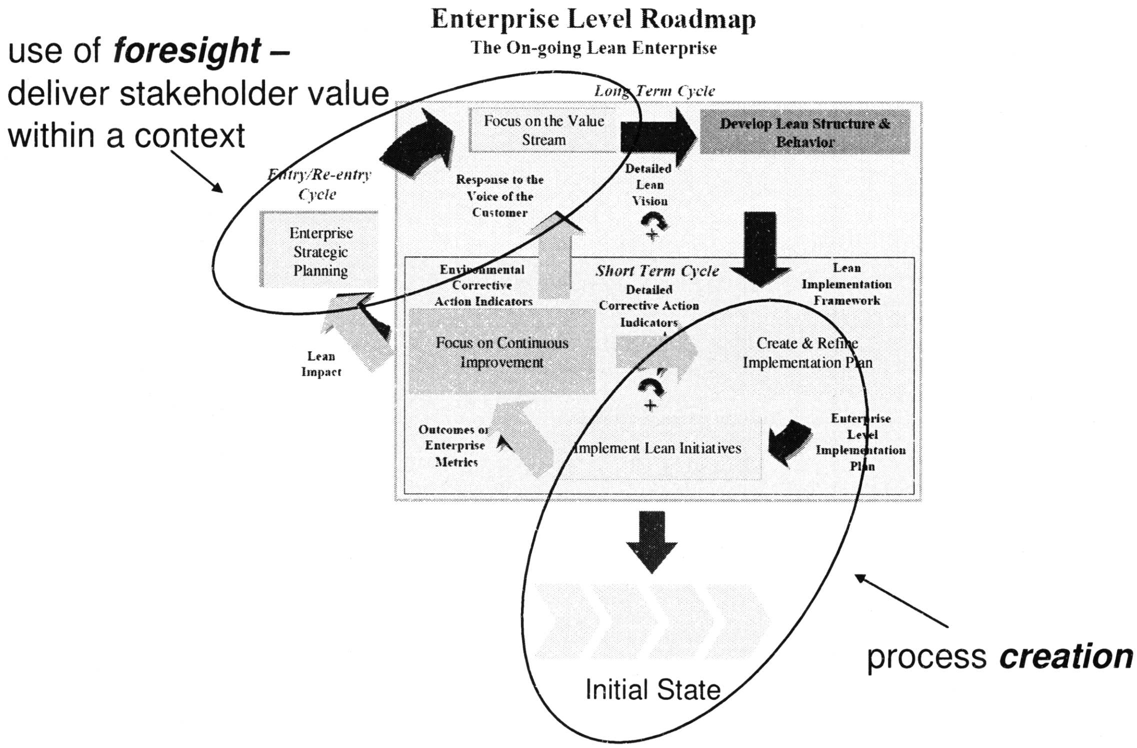


Figure 1.3 TTL Roadmap for Process Creation (adapted from Bozdogan et al., 2000)

After the benefits of hindsight are removed, process architects are left with strategic planning and goal setting for definition and implementation of an environment for Lean activities to flourish. Without a current state to improve upon, process designers must define processes and behaviors from scratch – that is, definition of the beginning, or initial state. Once the initial state is defined, the enterprise can enter the normal continuous improvement cycles of the TTL Roadmap as a transforming entity. The challenge for process architects is to define the initial state as appropriately as possible for the task at hand.

2. Background

2.1. Lean Enterprise Concept

Lean is a set of principles, practices and tools aimed at business process improvement. Inspired by revolutionary manufacturing initiatives first used by Henry Ford and adapted and perfected further by Toyota, Lean has evolved from a characterization of Toyota's revolutionary manufacturing practices into a sophisticated field of study impacting business processes across the enterprise. Lean Product Development is the subset of Lean research that applies the Lean philosophy into the product development function of the enterprise. To develop an understanding of Lean Product Development, it is critical to understand this evolution.

In the following sections a review of the evolution of Lean is presented; first, a brief history of Toyota and the genesis of Lean, next, the adaptation and translation of Lean principles to other business processes, and finally, the formulation of the notional Lean Enterprise. In this manner, Lean Product Development is captured as a process operating within the Lean Enterprise.

2.1.1. Lean Enterprise Foundations

Toyota's long history of production excellence has been a topic of investigation by the academic and industrial communities for many years. Volumes of literature are available discussing the practices and principles of Toyota, formulating theories of Toyota's success, and testing those theories at other firms. Additionally, many theses have traced the history of Toyota and summarized the collection of Lean manufacturing literature. Slack (1999), Chase (2001), Whitaker (2005) are comprehensive, and are recommended for review. There is no lack of recounting of neither the history of Lean, nor the production practices and successes of Toyota, and therefore no need to comprehensively retrace those steps here.

Briefly then, after WWII, events in Japan forced Toyota into a new manufacturing paradigm. With limited finances and resources, automobile manufacturing simply could not proceed following the mass production model. New techniques were developed to accommodate a build-to-order system more appropriate in post war Japan. These techniques developed and evolved and were successful, particularly at Toyota.

Over time it became apparent to the global automobile industry that the Japanese approach was working. Consequently, armies of industry experts and academics traveled to Japan to study Toyota's methods. Toyota's processes were found to be radically different than others'. These investigations have resulted in literally volumes of studies, books, and theses. Over time, the collection of this literature, along with its resultant principles and methods, has come to be accepted as Lean. Today, Lean is widely covered in academic literature, widely trusted and accepted in industry, and is widely practiced. Lean has been applied worldwide in manufacturing, and indeed has revolutionized the industry.

As the manufacturing environment became adequately characterized, further studies revealed that Toyota's processes extended beyond their own manufacturing floor, to influence their suppliers. Toyota had learned to approach their supplier network in new and interesting ways as well, extending their revolutionary ideas horizontally along the process chain feeding the automobile manufacturing process. This extension was proposed along the entire supply-chain, from raw materials to finished goods. These new insights, along with the successes of Lean improvements in production, inspired many to suggest the applicability of Lean principles as directed towards other business processes as well – specifically, business processes outside of the production process chain altogether. *Lean Thinking* (Womack & Jones, 1996), for example, has inspired many to consider Lean as appropriate for order fulfillment and product development, in a variety of industries.

In many cases, this meant reaching across traditional corporate barriers, both internal and external to the company. To apply the lessons of Lean, a new cooperative relationship among firms was necessary. *Lean Thinking* first proposed this inter-corporate cooperation as a concept known as the Lean Enterprise. As time has passed, additional studies and principles have emerged to characterize Lean behavior in firms. These proposals have been accepted into the Lean literature, and have contributed to the Lean Enterprise concept as well.

Lean studies have matured to redefine the traditional view of the "enterprise". Once clearly delineated by a *corporation's physical boundaries*, the new notion of the enterprise – the Lean Enterprise – is now defined solely by those processes and individuals participating *along a product's lifecycle*. These processes and individuals use Lean practices to deliver value to the various stakeholders of the enterprise. Today, the notional Lean Enterprise truly embodies a holistic, value-oriented, extended enterprise, applying Lean techniques both vertically and horizontally across multiple business processes and corporate boundaries.

2.1.2. Lean Enterprise Composition

Defining the Lean Enterprise requires a translation of the successes of Lean to processes external to the manufacturing environment. An appropriate method of translation is first to identify the guiding principles at work in Lean production processes. Once abstracted, these principles are then reconditioned to be applied within other process domains. The translation suggested here has been tackled by a number of researchers with notable results, some of which are referenced below.

In addition to the "principle-practice" translation described above, the Lean Enterprise is further defined by a number of notable concepts adopted and developed by Lean research. Some of which have been inspired by Toyota's *other behaviors* – outside of manufacturing. For example, investigations into Toyota's own product development organization have been made (Morgan, 2006), leading to additional insights. These and other insights have been adopted by Lean studies to enrich the definition of the Lean Enterprise.

2.1.2.1. Toyota Production → Lean Principles

The most well known statement of the principles at work in the Toyota production system are the five guiding principles identified by Womack and Jones in *Lean Thinking* (1996):

1) Identify Value,
2) map the Value Stream and identify Waste,
3) eliminate Waste and promote Flow,
4) implement Pull along the Value Stream, and,
5) pursue Perfection.

While working to define Lean Product Development and the Lean Enterprise, many have anchored arguments and thinking within the Five Principles (Slack, 1999; Chase, 2001). Also commonly cited are Ohno's Wastes (Walton, 1999; Wake, 2003). Ohno's Wastes, the Five Principles, and other themes are presented in detail in succeeding sections.

In the translation of Lean initiatives to the enterprise, themes such as Ohno's Wastes, the Five Principles identified by Womack and Jones, and other Lean concepts are used *as principles*. They are abstract concepts resulting from years of observation and study, by many researchers, of Toyota's revolutionary production techniques. Originating as *practices* on the manufacturing floor, these principles have been abstracted from practices in order to characterize an approach so successful, and so vastly different than any other of its time. The step required for translation must now be to analyze and interpret them for use in non-manufacturing environments.

2.1.2.2. Lean Principles → Lean Processes

As mentioned above, since *Lean Thinking* was written many have speculated that Lean tools and principles can be applied to business processes outside of manufacturing. Again, the rewards of the Lean philosophy are real, and their contribution to the success of companies like Toyota are widely accepted and acknowledged. Therefore, the temptation to apply the Lean philosophy corporate wide is understood. While it is easy to suggest that the principles should be used, an adaptation is necessary to effectively apply Lean principles for use outside of manufacturing. The appropriate adaptation is the subject of many works.

Manufacturing has unique characteristics and traits that are fundamentally different than other business processes. These differences prevent direct application of Lean practices, forcing one along the path of translation described here. Much of the work referenced in this thesis have grappled with this translation, conducting a rigorous investigation into how these principles should be transformed back into practices, appropriately, within the context of the target process.

2.1.2.3. Enterprise Toyota

The concept of the Lean Enterprise is complicated somewhat by additional research, in which Toyota's other enterprise behaviors are being characterized. Notably, Morgan (2006) examined Toyota's *actual* product development practices.

2.1.3. Lean Enterprise Summary

Together, these studies are forming the picture of the Lean Enterprise as it stands today. In general, the Lean Enterprise is being defined by a variety of applications of the Lean philosophy in various aspects of the enterprise. These applications include Lean manufacturing, a translation of Lean principles to other business processes, and an occasional look at Toyota's behavior outside of manufacturing.

A Lean Product Development organization operates within the Lean Enterprise. Much of the work done to define Lean Product Development has centered on adapting Lean principles for use in a product development context. In the sections that follow, research aimed specifically at defining Lean Product Development is reviewed.

2.2. Product Development

2.2.1. Product Development

An investigation into Lean Product Development requires a discussion of the notional organizational function of product development, that is: its definition, its boundaries, and finally the role that product development plays in, and its responsibility to, the greater organization.

2.2.2. Definition

An organization, for various strategic reasons, makes determinations about which marketplaces in which to participate and compete. Whether for profit or for other strategic goals, the organization must produce products and solutions for introduction into that marketplace. *Product development* is the organizational process that designs and creates products and solutions, for fulfillment of the organization's strategic needs for a particular marketplace. Broadly, product development is a socio-technical system composed of the individuals, methods, and resources employed to approach problem solving. Designers and engineers, their communications and interactions, the tools and techniques at their disposal, and the complexity of the problem at hand all contribute to characterize this system.

Ulrich & Eppinger (1995) define product development as "the set of activities beginning with the perception of a market opportunity and ending in the production, sale, and delivery of a product." Naturally, these *activities* include processes operating external to classical design and development engineering (for example, marketing and production). While it is true that many organizational processes are ideally involved in the product development process, Lean Product Development literature generally focuses on the activities of engineers participating in technical problem solving in a product development context. These participants include architects, designers, testers, team leaders, project managers, etc. Ultimately, it is from their perspective that Lean Product Development will be defined. Therefore, further clarification of the boundaries of product development is useful.

2.2.3. Boundaries

Broadly speaking, there are at least four main phases, or stages, in "the product lifecycle" (Crawley, 2005): the Concept phase, the Design phase, the Integration & Test phase, and the Operating phase. The Concept phase generally includes marketing activities, customer needs gathering, competitive analysis, and feasibility studies. The Design phase includes system architecture and detailed design activities. The Integration and Test phase includes the construction and evaluation of prototypes, proof of concept, and refinement of design details. The Operating phase includes production, fielding, and sustaining of the product.

To distinguish the product development activities of designers and engineers from that of marketers and assemblers, the Design and Integration & Test phases of the product lifecycle roughly bound the primary phases of interest for Lean Product Development research. While various other viewpoints exist regarding the exact boundaries of the product development effort (Ulrich & Eppinger's definition, for example) it is not necessary to resolve this contention here. The intention is to establish product development boundaries for discussion purposes, most particularly since we are attempting to define new Lean processes *outside of* manufacturing and production. Therefore, for discussion purposes, the boundaries of product development are taken as a subset within the overall product lifecycle defined above, beginning at the Design phase (generally downstream of concept development activities) and ending at the Integration & Test phase (generally upstream of production).

2.2.4. Roles and Responsibilities

Functioning within the boundaries established above, the role of product development in the larger enterprise is to materialize various product concepts and to produce plans for production. The materialization of a product concept requires an exploration into the greater set of possible problem solutions, a down-selection of said solutions, building, testing, and comparison of a chosen subset of solutions, and finally convergence on an appropriate alternative. Ideally, the processes employed result in plans for production that are feasible, reliable, and otherwise generally attractive to potential customers.

In this manner, product development provides the corporation with tools – in the form of physical products – to show, sell or otherwise operate and compete in a marketplace. The methods and procedures used by the product development team to produce these tools can vary in their effectiveness, timeliness, and costliness. It is the responsibility of the product development organization to deliver results as efficiently as possible.

2.2.5. Resources, Targets and Goals

The product development team is constrained by the quantity and quality of resources at their disposal. The two main resource categories are *people* (headcount, expertise, experience, and motivation) and *their environment* (facilities, infrastructure, tools, etc.). A budget is created to staff and tool a project as determined by management. As a project runs its course, resources are typically evaluated, negotiated and adjusted as necessary.

Utilizing the resources granted above, the product development team strives to meet quality, cost, and schedule targets for a given project, all of which impact the company's success. The product development team uses metrics to monitor behavior and progress along all three categories, continuously making trade-offs to keep the scope of each category achievable and appropriate for project goals.

Achieving the appropriate balance of quality, cost, and schedule in product development can prove to be somewhat recursive. Decisions regarding this balance can also affect those supporting, participating in, and depending on the process. For example, designers attempt to meet the customer's requirements for features and reliability while their design decisions commit production costs to be incurred during manufacturing. These costs affect the company's ability to profit from the sale of the product, and therefore affect the bottom line. Additionally, procedures used by designers to organize and communicate with one another to solve problems consume time, and therefore affect the ability of the product to hit the market's window of opportunity. On the other hand, aggressive schedules take tolls on employees and introduce quality uncertainty.

The Iron Triangle

The trade-offs, as defined above, tie into the classic "iron triangle" of project management. It is widely cited that these targets are frequently in contention and that "resolution of this tension has a direct bearing on customer value" (Slack, 1999). Clearly though, a balance is finally achieved as project runs its course. The question is, how should a management team tip the scales, or, (in hindsight) if the scales had been tipped differently, would the project have been more or less successful?

Consideration of the tension of the iron triangle provides an opportunity to introduce two additional perspectives on product development – that of *context* and that of *stakeholders*. Context, used here, refers to the surrounding business environment in which the company operates and competes, and in which the product will be marketed. The term stakeholder refers to any entity that relies on the product development organization to fulfill their needs and expectations. The customer, for example, is traditionally the most well-known stakeholder of the product development process, and is typically regarded as its primary beneficiary. Other critical stakeholders include the management team, who benefit from business opportunities created by the products developed, and investors, who benefit financially from a product's ultimate success.

The concepts of business context and stakeholder viewpoints are critical when attempting to consider why organizations competing in different marketplaces will vary in their decision making with regards to the iron triangle. The variations depend on the relative weighing of importance of quality, cost, and schedule targets – as determined by various stakeholders (desires and preferences) within the context (resources and constraints) of different settings. It is useful to consider that the appropriate balance of these targets is a strategic need for the enterprise, a need that is determined by business context and stakeholder viewpoints.

Context and Stakeholders Define Balance

> "Value in product development has historically been regarded as the appropriate balance of these metrics [quality, cost, schedule]." (Chase, 2001)

In summary, the goal of the product development process is to maintain the appropriate balance of the targets of quality, cost, and schedule, as determined by context and stakeholder viewpoints. The factors influencing perceptions of value are unique to how each stakeholder participates in the enterprise, and depend largely on the context, or business environment in which the company operates. The product development process serves the needs of various stakeholders with varying needs, such as directly impacting profit, research and development of intellectual property, developing prototypes, reaching exceptional quality and reliability targets, aggressive (or specific) time-to-market constraints, or perhaps competing in a market with particularly low margins.

2.2.6. Product Development Summary

Product development involves people: developers, designers, and engineers, with specialized skills and experience, who work independently and as part of a team. Individuals and teams participate in an environment: management and infrastructure, resources and constraints, employing processes and procedures in their approach to problem solving. Teams are organized to solve technical problems, of varying complexity, while balancing quality, cost, schedule and risk. The product development organization attempts to create value for all stakeholders within the context of business and enterprise factors.

2.3. Lean Manufacturing → Lean Product Development Translation

The premise that Lean principles and practices are applicable to processes beyond the manufacturing floor gained acceptance over the course of the evolution of Lean research. Lean Product Development is a practice based on this premise. The first step in applying Lean to processes outside of manufacturing, or target processes, requires an abstraction of the principles at work behind Lean process improvements. The Five Principles, as captured in *Lean Thinking* (1996) are generally accepted as the most appropriate statement of these principles. Next, the principles are reevaluated within the context of the new, target process. Throughout this reevaluation, cases of the principles at work in manufacturing are continually referenced for examples of the principle at work and the resultant improvements made, and for a general perspective on the characteristics of the process that allow the principle to play out so favorably. In this manner, it is beneficial to become acquainted with some of the mechanics of the manufacturing process, and particularly, to compare these mechanics with those of the target process.

In succeeding sections a review of Lean Product Development literature is presented. The review will sample a variety of research with regards to the reevaluation of the Five Principles while introducing a variety of topics, including product development stakeholders, product versus process value, and product development wastes. The review begins with a mechanics comparison of manufacturing and product development, our target process.

2.3.1. Manufacturing versus Product Development

In order to begin to apply Lean principles to other business processes, it is useful to first compare and contrast the mechanics of the target process (i.e. product development) to the manufacturing domain from which Lean principles were first derived. Both processes involve technical people working to solve technical problems, and aim to serve various common stakeholders – most notably the customer. Both processes scale in complexity with that of the product, and, for complex products both are particularly vulnerable to waste. These similarities provide a basis for motivating the Lean thinker to analyze product development further to test the applicability of Lean tools and techniques. Ultimately though, it is acknowledged that the two processes are sufficiently distinct that their differences need to be characterized, sorted, and analyzed.

Many authors (Slack, 1999; Bauch, 2004; Morgan, 2006) cited a variety of fundamental differences between the two processes, and view these differences as obstacles in applying Lean Manufacturing principles towards product development. For a comprehensive look at these differences, Bauch's discussion (2004, p. 33) is recommended as a particularly thorough review. For the purposes of this discussion, a few notable distinctions are highlighted below.

Nature of Tasks

> "Manufacturing processes are generally sequential in nature, they are non-iterative and they are repetitive. Contrast this with the product development process which is highly networked involving both sequential and parallel processes, it can be highly iterative with feedback loops to earlier process steps, and in general, is not highly repetitive." (Slack, 1999)

Due to the repetitive nature of the production process, improvement efforts are simple to conceive, are generally straightforward to implement, and come with tangible results.

Manufacturing assembly steps occur in plain view with obvious participants. Product development, on the other hand, is often iterative, with grey boundaries among individuals, and with less tangible results.

Analyzing the implications of these differences brings to mind the relative simplicity of human interaction required for production processes. It is interesting to note that the predictability that a repetitive process affords not only impacts technical aspects of the process, but social aspects as well. In contrast, consider the potential magnitude of the social complexity of product development activities. Multidisciplinary problem solving is largely a social activity. Product development inherently involves more complex social interaction than manufacturing, and likely more communication needed (Morgan, 2006).

Task Timing

The repetitive nature of manufacturing tasks leads to stability with regards to task completion times. In contrast, product development tasks are often new and unique problems which naturally vary in complexity – and therefore vary in timing as well. The variations of task timing in product development versus the stability and predictability of task timing in manufacturing is a characteristic difference between the two processes.

The Toyota Production System has exploited the time predictability of manufacturing tasks through a concept referred to as *takt time*. Takt time refers to Toyota's system of task start-time/stop-time synchronization, where a sequential production process is iteratively decomposed into tasks of relatively equal duration, and execution of tasks are synchronized across the entire assembly line. In this manner, the time predictability of individual tasks is extended to provide time predictability along the entire assembly line. Toyota has proven the benefits of takt timing to include lowered inventory buildup between tasks, improved process transparency, and a reduction of time to discover weaknesses and faults in the process. Today, implementation of *takt timing* is considered fundamental to Lean manufacturing initiatives, and is largely acknowledged for its positive impact on the production floor.

Many have tried to imagine a product development process where tasks are synchronized, desiring benefits such as improved schedule and staffing predictability, and more efficient management of information inventories (following sections address how poor management of information can lead to waste). Unfortunately, many have doubted that product development tasks can be synchronized, due to their time variability and unpredictable nature (Morgan, 2006). (Recall that *takt* time implementation is possible in a manufacturing environment mostly due to its repetitive nature.) While acknowledging the possible benefits, others have dismissed synchronization of product development tasks as an ideal impractical for an environment as uncertain as the product development environment (Slack, 1999).

Information vs. Physical Product

An important distinction between manufacturing and product development revolves around *what* is actually created in the process. A major hurdle in Lean studies has been to grapple with the fact that it is *information* (build-to specifications, for example) that is created in product development - as opposed to physical product. Physical production in a manufacturing environment is clearly visible, countable, and verifiable. In contrast, much work in product development is intangible (Morgan, 1996). The creation of information is inherently more difficult to track and control than physical product.

Much work has been done to resolve the implications of the "information product". Bauch (2004) for example, included a detailed study of information. Information studies have led researchers to suggest improvements in the quality and flow of information in product development. Major accomplishments by the Lean community include initiatives improving communication among team members and within and among information technology (IT) systems. Human communication initiatives include displays, "war rooms", collocation, and Integrated-Product-Teams (IPTs). IT system improvement initiatives include database and data type interoperability, and improvements in tools (i.e. computer-aided-engineering tools interoperability).

It is interesting to note that some product development activities more directly impact "physical product" than otherwise acknowledged. Software developers, for example, do indeed create deliverable product. Software, while certainly less tangible than physical product, is more akin to physical product than to a build-to specification. Therefore, when contrasting product development activities to manufacturing it should be recognized that there are types of product development that are perhaps more akin to manufacturing than not.

Position of the Process within the Value Stream

Referring back to the definition of the product lifecycle, note that manufacturing is located further downstream, or closer to the final customer, than product development. There are several important implications of this relative positioning which are highlighted below.

Manufacturing enjoys proximity to the customer, improving the likelihood of market interaction and feedback. Additionally, the mission of serving the customer's needs is more clearly defined. For example, there is little ambiguity in the definition of quality in manufacturing (i.e. meeting specs and tolerances).

Feedback in product development is more difficult to cultivate, and quality therefore is harder to define, and more likely to be off target. Product development activities are burdened with uncertainty and variability of inputs to a greater extent than manufacturing. Therefore, in product development it is necessary to continually reevaluate these inputs over time.

Unfortunately, lack of access or proximity to the customer does not relieve product development of the burden and importance of constant customer involvement. Product development teams must be sure of their interpretations of customer needs – whether through pre-approved and cooperative specifications or cultivated through focus groups.

The context in which the company operates defines the mechanism by which customer feedback is obtained.

Stakeholder Identity

Manufacturing has the luxury to focus solely on the end-user customer of the product, while product development must consider a variety of enterprise stakeholders. Manufacturing's direct customer is the end-user, and all efforts may be focused on serving that customer directly. Product development, on the other hand, has a more complex "customer" environment typically serving additional enterprise needs (Chase, 2001). Product development, therefore, must keep a broad range of stakeholder's needs in mind. Of course, manufacturing serves a variety of enterprise needs as well, but it is generally acknowledged (and Lean strongly advocates) that through a narrow end-user focus all of the other enterprise stakeholders will also benefit. A detailed discussion of product development stakeholders is given the main body of this thesis.

The differences between the manufacturing and product development domains present notable challenges for extending Lean initiatives into the product development domain. Despite these differences, the potential of Lean Product Development is widely accepted, and in fact is being practiced in bits and pieces today in industry. The following sections touch on how these and other challenges have been addressed by the Lean community while formulating a definition of Lean Product Development.

As stated above, many authors working to define Lean Product Development have anchored their thinking with the Five Principles identified in *Lean Thinking* (1996):

1) Identify Value,
2) map the Value Stream and identify Waste,
3) eliminate Waste and promote Flow,
4) implement Pull along the Value Stream, and,
5) pursue Perfection.

The sections below review a variety of authors' works within these principles.

2.3.2. Identifying Product Development Value

Value identification is the traditional starting point for Lean analysis. Efforts have been made in Lean Product Development research to comprehensively address the question of value (notably Slack, 1999, and Chase, 2001). Key points from this research are highlighted below.

Value Identification

The question of defining value has been studied and debated over time and has proven to be a daunting task (Chase, 2001). Generally, an acceptable qualitative expression of value is a ratio expressed as benefit-to-cost. It is common in Lean Product Development to define this ratio from the perspective of the various stakeholders of the product development process, and to evaluate their needs and desires within the general framework of quality, cost, schedule, and risk. As an example, in the ratio, benefit is typically decomposed into various product attributes (quality) which are available at certain times (schedule), at a particular cost, and so on.

2.3.2.1. Product Development Stakeholders

Traditional Lean manufacturing literature states that the focus of all value analysis should be through the eyes of the end-user, or customer (Womack & Jones, 1996), and that through serving the customer directly all other stakeholders benefit. Slack (1999) and Chase (2001) argued for a reevaluation of a customer-only value focus in Lean Product Development. As product development processes reduce uncertainties with regards to quality-, cost-, and time-to-market, they arguably serve a multi-stakeholder base (enterprise stakeholders as well as the customer). Value delivery analyses must therefore be extended to include multiple stakeholder viewpoints.

Product development stakeholders include customer-recipient, management, shareholders, and employees. Customer-recipients enjoy the benefits of the process output (the product), ultimately at some cost. Customer-recipients include the end-user or operator, of the product but also the customer-buyer as well. The management team is a stakeholder in the process because product development strives to fulfill their strategic needs in regards to the quality, timing, and costs associated with a product. The strategy employed by management again depends on the context in which the product is being designed, built, and delivered. When quality is a concern, the product development process must be robust. When timing is a concern, the process must be agile, and so on.

Shareholders are those individuals (stockholders, private investors) who have a financial stake in the enterprise. The long-term interest of shareholders is typically characterized as return on investment. Shareholders are stakeholders of the product development process because product development establishes a foundation for operating and production costs, a potential for revenue, and therefore potential profitability for the organization. Employees are stakeholders in a business process such as product development in a variety of ways. First, employees may benefit, similarly to shareholders, from the quality of the process and therefore its contribution to the success of the organization. More importantly, as participants in the processes used by an organization, employees experience a quality of work-life largely defined by characteristics of the process itself. Additional stakeholders are possible, including the environment, government, municipalities, and society in general.

In summary, a value analysis must be made from the viewpoint of all of the stakeholders of an enterprise process. In the case of manufacturing, value is characterized with regards to a single stakeholder viewpoint, that of the end-user. In the case of product development there are multiple stakeholder viewpoints, all of which must be considered.

2.3.2.2. Quality, Cost, Schedule & Risk

As suggested above, stakeholder value in Lean Product Development is generally characterized qualitatively within the framework of quality, cost, schedule, and risk.

Quality

End-users typically view quality as the ability of a product's features to suit their needs. The shareholder, on the other hand, may view quality as an attribute of the process itself (Chase, 2001). For example, process quality may refer to a process's consistency with regards to schedule, or its ability to create, capture, and share knowledge. Employees may value a process to the extent to which it is rewarding to be a participant within.

Cost

Costs are measured by the customer from the perspective of acquisition costs, measured either absolutely or relatively (to competing products). Costs with regards to shareholder perception are more complex. Typically, these costs are decomposed into production costs (costs *committed* during product development), and perhaps less significantly, into the cost of the process itself (costs *incurred* during product development), both of which ultimately effect profits. It is common in the literature to cite comparisons between the two, where costs *committed* typically dominate (Slack, 1999). Finally, employees may measure costs by considering compensation (Slack, 1999) and their work-life balance.

Schedule

Schedule, or timing, of a product's development is a critical attribute affecting value. End-user perception of timing is typically presented as availability, or "lead-time". Shareholder perception of timing is typically presented as time-to-market, where a window of opportunity exists in a marketplace. Employee perception of timing may take into account their work load, and with regards to reasonable schedules, overtime, and burn-out.

Risk

A significant contribution of Lean Product Development research is the perspective that product development activities reduce risk (and therefore add value to the organization) (Browning, 1999). Risk reduction is highly valued by all stakeholders, but again, unique preferences are likely. The customer may have very aggressive time constraints, and therefore schedule risk may be the most important factor. The product development team may evaluate the schedule, and propose to meet customer demands with more exposure to increased costs.

Product Development Context

The discussion above emphasizes the importance of context with regards to value. The characteristics of the business context within which the company operates and competes determine what tradeoffs are appropriate, with regards to quality, cost, schedule, and risk. One company may value time-to-market because of competitive pressures, while another may value quality-to-market because of consumer safety and company reputation. A closer look at business factors and other components of context is made at the end of this chapter, and is the primary subject of Chapter 3.

2.3.2.3. Product vs. Process Value

The general discussion of quality, cost, schedule, and particularly the perspective that product development activities reduce risk, highlights a critical distinction made by Chase (2001). In Chase's exploration of *Value creation in the Product Development Process*, a framework is presented where process value is characterized distinctly from product value.

In Chase's view, product development "embodies enormous uncertainty" (2001, p. 19) with regards to quality, cost, and schedule, while the activities occurring in the process itself decrease this uncertainty. Certainly, the organization and structure of these activities, and the methods and tools used to execute them, impact the efficiency by which the process reduces uncertainty. A goal of Lean Product Development should be to maximize this efficiency, and therefore maximize process value. As Chase points out, capable processes, while deemed valuable for risk reduction, often do not lead to successful products. Therefore, process value should be considered distinctly from the product value.

Chase's position on product versus process value underscores various complexities with regards to Lean Product Development, including; the importance of expanding the stakeholder analysis beyond the customer, and an emphasis on the process as well as the product.

In summary, a simple customer definition of value, while suitable for manufacturing, is insufficient for Lean product development. Once an understanding of value within the context of the company and the product as determined by the specific stakeholder objectives has been gained, an exploration of the applicability of Lean concepts to the development of that specific product can be made.

2.3.2.4. Qualifying Value

The factors influencing stakeholders' value perceptions lead to a variety of representations of value. Each stakeholder has its own unique representation (i.e., its own "value equation") where value is generally expressed in terms of quality, cost, schedule and risk. Various representations of stakeholder value have been proposed, each offering useful insights. In general, qualitative representations are deemed more appropriate. Some examples of these representations follow.

Customer Value

Slack (1999) offered both representative equations of value and value models. In his analysis of customer value, Slack offered the following equation:

$$\text{Customer Value} = \{\sum(N \text{x} A) \text{ x } f(t)\} / C$$

where,

N = a product feature to satisfy a need (as determined by the customer),

A = the ability or effectiveness of a product feature to satisfy the need (determined by how well product development processes are executed),
f(t) = the availability of the product or feature (lead-time), and,
C = cost of the product or feature to customer (a function of product attributes as well as process efficiency).

Slack's summation, $\sum(NxA)$, indicates multiple product features, and the ability of each to satisfy a customer need. Later, Slack added a link to risk, suggesting that ability, A, can be decomposed further into (1-R). Here, R represents the probability, or risk, that the proposed feature does not satisfy the customer's need. The quality, cost, schedule, and risk framework is evident in Slacks representation of customer value. The customer is concerned with a product's features and its availability, at a certain cost.

Shareholder Value

It is common in the literature to express shareholder value in purely financial terms with emphasis on profitability. Slack (1999, p. 19) for example, uses the economic concept of Economic Value Added, where "a company only creates value for its shareholders when its operating income exceeds the costs of capital employed." Chase (2001) also references traditional corporate viewpoints on value, expressed in terms of net present value, internal rate of return, and break-even point. Chase points out that from an accounting point of view product development appears as a liability, until revenues are added to balance the books.

Typically then, from a financial perspective, shareholder value is generally represented in the following manner: costs are incurred and (perhaps more predominantly) committed during the product development process, while product development activities determine the quality and availability (schedule) of the product. Overall, the product development process reduces uncertainty with regard to all three. As quality and availability determine the potential of the product to generate revenue, ultimately, shareholder value is determined by the extent to which a product's revenues outweigh the costs described above.

While an emphasis on profitability is appropriate from an accounting perspective, it can prove to be an awkward metric when considering product development value, particularly in the short term (Chase, 2001) where revenues are merely estimated. Profitability is not only an awkward metric, but it is incomplete to neglect other, less tangible outcomes of the product development process. The liability perspective, for example, is a viewpoint that fails to capture the value and significance of risk reduction or other process outputs, such as knowledge capture.

Employee Value

Slack (1999) expresses the employee's value perspective as a function of job quality versus compensation, as follows:

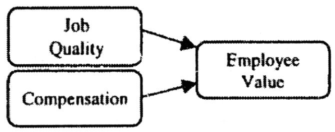


Figure 2.1 Slack's *Employee Value*

The characterization of job quality and compensation is the subject of a large body of research including works in motivation and employee satisfaction studies.
In regards to motivation, Benabou and Tirole (2003) introduce Intrinsic vs. Extrinsic motivators. *Intrinsic* motivation is a natural tendency, or a predisposition, towards motivation, while *extrinsic* motivation is that which is influenced by a reward, such as cash compensation. Studies (such as Desi, 1975) suggest that while the short-term influence of an extrinsic motivator may be great, the long lasting effects leave less to be desired. In a general investigation of motivational factors and influences, Herzberg (1968) revealed factors leading to job satisfaction (motivators) and those *preventing* job dissatisfaction (hygiene). Examples of hygiene include compensation and workplace conditions, while examples of motivators include recognition and responsibility.

In an investigation of employee empowerment, Bowen and Lawler (1995) characterize employee empowerment as control over how to perform a task, "awareness of the business and strategic context in which the job is performed", and accountability for the work that is done.

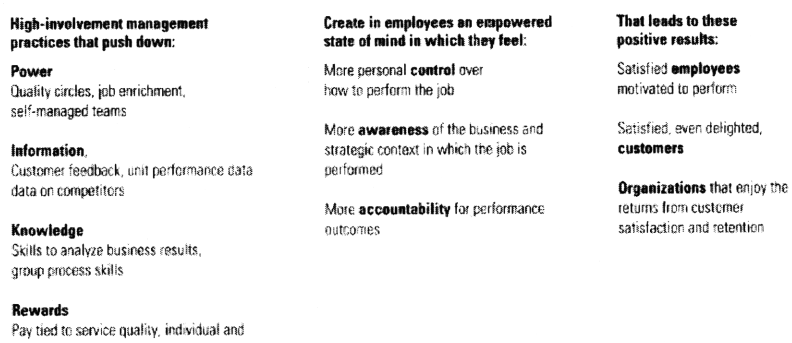


Figure 2.2 Bowen and Lawler's *Employee Empowerment Approach to Service*

The suggestion of Bowen and Lawler, Herzberg, and others is that the employee value perspective is a complex formula accounting for a variety of factors.

Composite Representations
As Slack (1999) evolved his value proposition further to include employee and shareholder perspectives, and concludes with a qualitative model for customer value, which includes employee and shareholder value perspectives implicitly. Oehmen (2005) also offered a characterization of value. Chase (2001) worked to resolve the contention between a strictly financial representation of shareholder value (*product-centric*) and a more holistic view of enterprise value. Chase (2001, p. 43) asserts that executives acknowledge the indirect contributions of product development to profitability, and asserts that a *process-centric* view of value is necessary to capture these contributions. Chase's conclusion is the proposal of a value framework which highlights his distinctions.

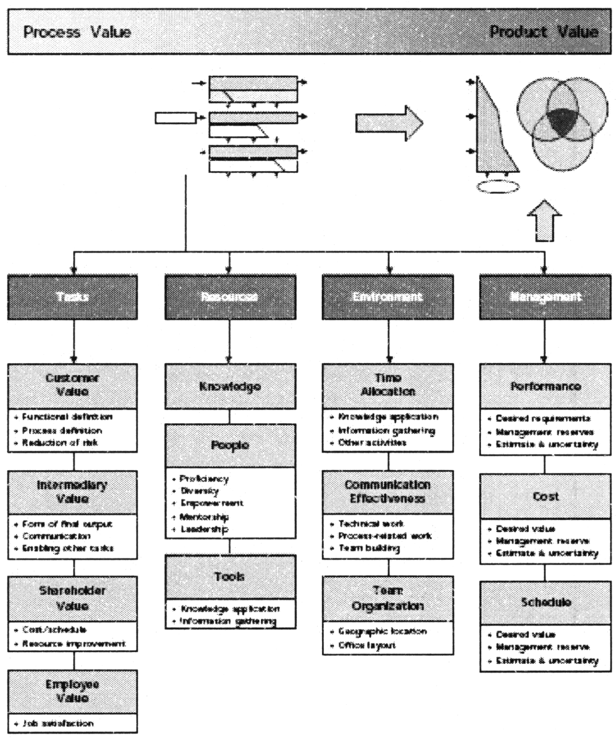


Figure 2.3 Chase's *Value Delivery Framework*

2.3.3. Product Development Value Stream

According to the Five Principles, the second step in a Lean implementation and analysis is to identify the value stream. Industrial processes, or value streams, are typically captured via process flow diagrams, of which there are many styles. In the case of Lean manufacturing the widely accepted technique is Value Stream Maps, or VSM. In VSM, boxes are used to represent tasks, or process steps, and are captured in sequence. Information flows are also captured. After mapping is completed, process steps in the map are scrutinized for their contribution to customer value. Process steps are typically labeled or scored as *value add*, *necessary non-value add*, or *non-value add*. This initial characterization of a process via process mapping is referred to as the current state in

Lean. A visual mapping of the value stream is critical for the purpose of waste identification in the current state.

2.3.3.1. PDVSM – Product Development Value Stream Mapping

Much work has been done in Lean Product Development research to adapt and improve VSM for product development (McManus, 2004; Morgan, 2006). Detailed works have grappled with the appropriate use of symbols and notations for capturing product development activities (notably Kato, 2005). The collection of this work has led to a fairly mature mapping technique, now referred to as Product Development Value Stream Mapping, or PDVSM. Additionally, the appropriateness of PDVSM for product development has been benchmarked and demonstrated to be useful by comparison with more traditional project management tools, such as EVM (Whitaker, 2005).

Value Stream Mapping is useful because it helps one to visualize and characterize wastes. Waste in product development, particularly in an aerospace setting, is widely acknowledged and accepted. Several studies are commonly cited in the literature to emphasize this point (Whitaker, 2005, p. 25-26). Indeed the numbers are staggering, with some studies showing >60% of engineering time spent idle on large projects.

2.3.3.2. Product Development Waste Categories - Bauch's Wastes

Manufacturing wastes were first identified at Toyota by Ohno (Womack & Jones, 1996). Ohno's classic seven waste categories are well known as follows:

1) Overproduction
2) Transportation
3) Waiting
4) Over-processing
5) Inventory
6) Unnecessary Movement
7) Defective Product

In an effort to characterize wastes in product development, it has been very common in the literature (i.e. Millard, 2001; Morgan, 2002) to draw analogies and comparisons to Ohno's original seven waste categories. In a comprehensive review of these works, Bauch (2004) distills and projects the analogs into the product development domain as a "re-organized and extended waste system". Bauch's analysis itemizes common occurrences of product development wastes falling within the seven categories above, and identifies three additional product development waste categories (Re-invention, Lack of System Discipline, and Limited IT Resources), yielding ten categories in all. Several key repeating themes in the study of product development wastes are highlighted below.

Lack of Synchronization

On the manufacturing floor, Ohno observed Overproduction waste - producing more than necessary, or faster than necessary. Overproduction in manufacturing results from lack of awareness of workload, or awareness of status of upstream and downstream tasks. Overproduction is the root cause of the buildup of Inventory, another of Ohno's waste categories.

In his analysis of Overproduction, Bauch (2004) asserts that Overproduction is the result of lack of synchronization between upstream and downstream processes. Bauch properly identifies pull processing as manufacturing's remedy for Overproduction, and notes that the contribution of pull processing is that it provides a synchronization mechanism between adjacent process steps. In a pull system, an upstream task never begins work on a task until it is requested, *or pulled*, by the adjacent task downstream. The implication is that a synchronization mechanism might be useful to prevent wastes in Lean Product Development.

Idling Process (Information and/or People)/ Information Availability

After a process is synchronized, wastes due to Waiting are exposed. Analysis of wastes due to Waiting involves the concept of *too fast vs. too slow*. In Overproduction it was suggested that process steps should be aligned in start times, and therefore a cadence established in the sequence. Once process step start times are aligned, it becomes evident that some steps are occurring *too fast*, where individuals will most obviously be standing still, waiting for the completion of other steps which are occurring *too slowly*. The result is waste due to waiting, either product buildup, or, people waiting for product. In product development there exists a similar potential for wastes due to Waiting, either information waiting for people, or, people waiting for information (Bauch, 2004).

Minimizing wastes due to Waiting requires an optimization step where process step boundaries are "relocated" to manipulate the volume of work completed during execution of the process step. In this manner the duration time of the process step itself is impacted, with the goal being to bring the duration time in line with other process steps in the sequence. This effort is referred to as *load-balancing*. While operating within an already established cadence in a sequential process, load-balancing can help to eliminate wastes due to Waiting. While load-balancing and synchronization are simple to imagine in an environment as repetitious as mass-production, they are harder to envision in the world of product development.

Communication Inefficiencies/Information Quality

On the manufacturing floor, Ohno observed Transportation Waste - when workspaces are not collocated, when sequential process steps occur across separate facilities, companies, or cities, materials must be transported. During transportation, materials are simply moved from point A to point B (Bauch, 2004), an effort not contributing towards completion of the assembly of the product, and therefore value is not added. The goal of Lean Manufacturing initiatives is to minimize the amount of material in transport, the time material is in transport, and to minimize the costs associated with transporting.

The first perspective on Transportation waste, and its analog in product development, is tied to the conclusion of the Lean community that *information* is created in product development. Therefore, inefficient or unreliable information exchange, or data-trafficking, is a potential cause of waste. Data trafficking is the transmission, conversion and compatibility effort related to the exchange of data and files between individuals and work groups. Furthermore, the sheer volume (Bauch, 2004) of data exchange taking place in the development of complex systems can be a haven for waste, even if handled efficiently, considering the nuances that arise between cross-functional, facility, company, and international borders. Ineffective communication – the ineffective transportation of information among individuals - inhibits team cohesiveness and ultimately their ability to produce efficiently.

Interruption and Disruption of Tasks
An alternate perspective (Bauch, 2004; Hallowell, 2005) on Transportation waste and its meaning in product development, introduces problems related to task interruptions. Data exchange among individuals is described by Bauch as a *hand-off.* The flow of productivity is interrupted during a hand-off, when one product developer passes information (the "product task") to another, who has to ramp up to the new task. A change of task incurs costs (i.e. ramp-up, learning curve) and is analogous to Transportation waste at process step boundaries. Hallowell (2005) points to task hand-offs and pleads, "minimize wasted 'knowledge recovery time' switching in and out of different tasks."

Addtionally, Bauch and Hallowell propose the issue of an individual product developer burdened with the assignment of multiple tasks, where that individual is continuously *task-switching*. During task-switching, an individual is effectively transporting information, continually "moving" from point A to point B, with mental set-up and ramp-up costs recurring. The implication is that Lean Product Development practitioners should minimize interruptions, restrict multi-tasking, and generally allow developers to freely continue with their work.

Time-sensitivity of Information
Physical inventories are widely accepted as a source of waste in operations. The study of *information inventories* in product development exposes that, similar to physical inventory, a critical characteristic of information is its sensitivity to time. Kent Beck (Wake, 2003) has asserted that inventories in software are "un-deployed decisions". In product development, information, ideas, requirements, or partially developed software code all have a vulnerability to Inventory Wastes as they begin to accumulate. Wake (2003) commented on the risks with regards to requirements not yet designed in, downstream to finished code not tested, all of which are sitting idle and have the ability to become stale and outdated. While wastes due to Inventory often deal with information staleness, wastes can also occur when information is utilized *prior to* maturity (Hallowell, 2005). The time-sensitivity of product development tasks can be managed through prioritization and delayed decisions.

Bureaucracy of Process

Product development suffers unnecessarily through excessive approvals and redundant sign-offs (Poppendieck & Poppendieck, 2003), all of which add little value, but instead waste time and add costs.

2.3.3.3. Waste Identification via PDVSM

Earlier, it was proposed (McManus, 2004) that product development could be successfully mapped using mostly traditional VSM mapping symbols and notations, with slight enhancements for representing product development activities. Since then, the work of Bauch (2004), and others cited above, has resulted in a more sophisticated understanding of the nuances of product development waste classes and waste drivers. Consequently, it has been argued by Kato (2005) that major improvements would be necessary to enhance PDVSM to represent these nuances on process maps.

Kato (2005) developed symbols and notation for capturing these new aspects of waste. While much of Kato's work centered on the *individual*, other mapping improvements are suggested by Kato and Whitaker (2005) to better capture *project-wide* efforts. Notably, Kato emphasized the use of "swim-lane" structures in mapping to show the parallel efforts of a project team. Whitaker's work maps extremely complex projects with many participants.

2.3.4. Product Development Flow

After the process map has been completed, process steps labeled (as value add, necessary non-value add, and non-value add), and wastes identified, the map can be improved to its future state. The process map future state is achieved by improving the map to eliminate all process steps that do not contribute to customer value. This effort is made to eliminate waste, and to promote the flow of value along the process stream.

2.3.4.1. Waste Elimination

Bauch (2004) created a path for waste elimination, and noted that many of the waste drivers are interrelated, and interdependent. To determine the extent of interrelatedness among the various waste drivers, Bauch uses a cause-effect matrix to determine those waste drivers that are either; passive, critical, active, or independent. Bauch concludes with clusters of waste drivers (2004, p. 80).

2.3.4.2. Lean Product Development Flow (LPDF)

Oppenheim (2004) proposed a complete schematic for a generic Lean Product Developement Flow (LPDF). The schematic incorporates the notions of takt timing and team concurrency, and is presented in the figure below.

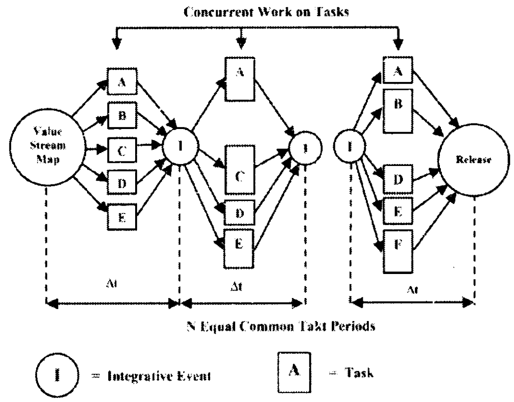


Figure 2.4 Oppenheim's "*Lean Product Development Flow*"

2.3.5. Product Development Pull

The principle of "Pull", as implemented on the manufacturing floor, is a system whereby cards are used to help downstream tasks control the flow of material from upstream tasks - before a downstream task is ready to accept what is produced. The cards, known as *kanban*, are passed in the upstream direction from process step to process step, and are considered a prerequisite to begin work on a task. The nature of this process allows a downstream task to control the flow of material or product from the adjacent process step upstream, virtually eliminating the buildup of Inventory between process steps. Toyota's system, known now as "pull", was indeed a revolution in auto manufacturing and has seen enormous adoption in other industries and processes as well. Pull is now accepted as one of the five Lean principles that define Lean's foundation.

Pull might be less applicable to product development than in manufacturing due to the fact that manufacturing steps are more predictable in nature, while product development tasks are certainly less deterministic (Slack, 1999; Bauch, 2004; Whitaker, 2005). Product development activities are non-repetitive, and require more complex interaction among team members. Much of the work is done in parallel with downstream tasks. Many of the tasks change form between iterations. Still, the notion of *information inventories*, and their associated wastes (Bauch, 2004), are concerning to product development managers. Many involved in product development would like to implement pull techniques to minimize information inventory. Still, the principles and mechanisms of kanban and takt timing have been noted as synchronization of process steps (Bauch, 2004), which as Oppenheim (2004) presented, may have applicability in product development processes.

2.4. Lean Product Development Summary

The research highlighted above demonstrates the diversity of research approaches taken and the richness of conclusions made while defining the complex field of Lean Product

Development. Review of this research has helped this author to form the realization that Lean Product Development is more than just a simple process. Instead, lean product development is a combination of individual, managerial, infrastructure, *and* process attributes collaborating to create a complete product development environment, striving for value delivery to all stakeholders. Below, the reader is offered a final perspective on Lean Product Development research. Finally, the section is concluded with a simple model for Lean Product Development.

2.4.1. Common Research Approaches

While the research approaches outlined in the sections above generally follow the traditional Lean Principles approach – value identification, value stream mapping, waste elimination, implementing pull, and continuous improvement – the vast majority of product development research has centered on the first three principles (specifically value, value stream, and waste). In a previous section it was mentioned that investigations into two critical Lean aspects, value and waste, yielded similar conclusions with regards to Lean Product Development. This section explores these points in finer detail, and uses these findings to frame general conclusions about Lean Product Development.

Linkages between Value and Waste

Value and waste are linked by the definition of value, generally accepted as a ratio of benefits to costs. Many authors have used this relationship to argue that elimination of wastes (improving costs) in product development increases value – "waste-based" approaches to increasing value. Indeed, the many successes of Lean are rooted in waste elimination on the manufacturing floor. Later, Browning (2000) and others (Chase, 2001) argued on the side of value (or benefits, in the numerator). Ironically, many of the same conclusions can be drawn from a comprehensive study of either waste or value.

2.4.1.1. Waste-based Approach

As mentioned above, many authors have explored product development wastes at length. A detailed look into Bauch's[1] (2004, p. 78) waste drivers in his "Checklist for Waste Elimination" reveals that those factors most influencing waste in product development fall into four silos, or categories: participants, infrastructure, management, and process[2]. The following table, adapted from Bauch's conclusions, is a summary of these four silos.

[1] Bauch's work is representative of a most comprehensive study on product development waste.

[2] While Bauch did not summarize his findings in this manner, the taxonomy is chosen for comparison with similar conclusions from others, including Chase's (2001) findings in his exploration of product development value.

Participants
employee skill sets – training and hiring practices to put the most competent team in place for a given task
employee attitude – true collaborative team players
employee discipline to work content
physical location of individuals – collocation necessary

Infrastructure
appropriateness of IT systems – data accessibility, compatibility, sufficient capacity
capability of CAE tools
capacity of tools
capacity of individuals

Management
disseminating/displaying information
creating awareness of individuals' roles and responsibilities
creating awareness of others' roles and responsibilities
creating awareness of schedule
effective project management behavior - critical path management
appropriate prioritization and sequence of assigned tasks
discipline regarding team members work load
discipline regarding team members work contents – minimize switching and hand-offs

Process
promotes communication of progress
promotes synchronization of tasks across development team
promotes effective and continuous group sharing of project details
stresses efficiency over bureaucracy

Table 2.1 Adapted from Bauch's "*Checklist for Waste Elimination*"

2.4.1.2. Value-based Approach

Similar to waste, many authors have looked at product development value at length. In Chase's "Framework for Delivering Value in Product Development" (2001, p. 56), Chase proposes that value in Lean product development processes is achieved through the selection of *tasks*, provisioning of appropriate *resources*, within a development *environment* that promotes communication among all team members, and finally, a competent *management* approach.

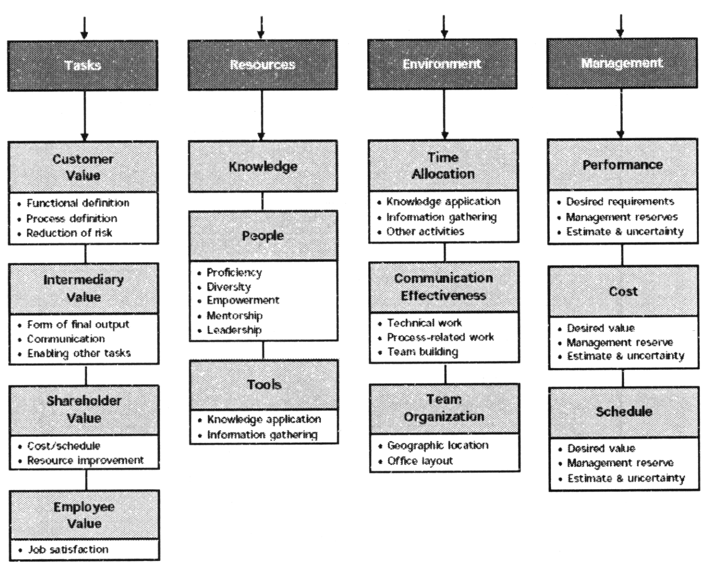


Figure 2.5 Chase's "*Framework for Delivering Value in Product Development*"

2.4.2. Components of Lean Product Development

Inspection of the results of the waste and value studies presented above reveals the many similarities of the conclusions made between the two points of view. The extent to which a product development organization is Lean is a function of several variables, including: qualities of individuals, the management techniques employed, availability and capability of appropriate environment, management techniques used, and the organization and traits of the processes or methods individuals use to achieve an end.

The following model has been derived as a summary of Lean Product Development and emphasizes the realization that Lean Product Development is more than just a simple process. Alternative frameworks and decompositions *are entirely possible.*

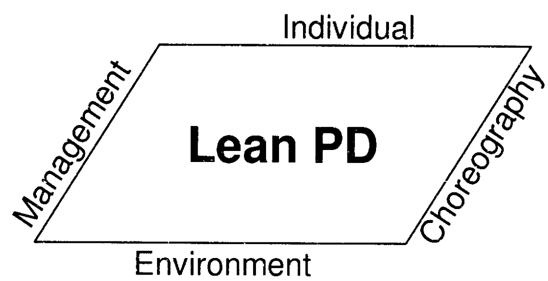


Figure 2.6 Components of Lean Product Development

2.4.2.1. Individuals

Lean Product Development *individuals* are detail-oriented and thorough in their own work. They are imaginative and creative, independent and holistic thinkers. Individuals are curious and aware of stakeholders, value delivery, project progress and success. Lean Product Development individuals have the appropriate experience for the task at hand, or the capacity and willingness to overcome learning curves quickly. Individuals are open-minded and excited about process perfection.

Lean Product Development individuals are consummate team players. Individuals are highly communicative, comfortable working with others, and are role-players (listeners and leaders) with flexible preferences for task assignments, occasionally acting as either mentors or apprentices when necessary.

2.4.2.2. Management

Lean product development managers (including functional management) are project-oriented, system and stakeholder aware, and are holistic thinkers. Managers are relentlessly aware of schedule, cost, and project status, and are wholly responsible for deliberate assignment of tasks and prioritizing of function and feature development. Management values communication with the development team, and is therefore highly communicative of project wide status. Management creates and establishes awareness for individuals of their roles, responsibilities, and their contributions to the team and to the project's success.

Lean Product Development *management* places trust in employees' capabilities, is highly supportive of, and empowers individuals to develop and perfect their own work structure. Management structures an individual's work environment through task assignment, while encouraging individuals to take ownership of tasks and responsibilities, and to pursue perfection for each. Managers carefully craft rewards and recognition in support of Lean behavior.

2.4.2.3. Environment

Lean Product Development *environments* are ideal with regards to resources, including infrastructure, information systems and supportive processes. Infrastructure, such as facilities and equipment, are ideally suitable for tasks at hand, including size, capacity, and vintage. Office space, workspace, and laboratory space are flexible and easily configured. Environments are ideal with regards to tool and system compatibility. Information, including internal and external sources, project information, component information, process information, requirements and specifications, and users' guides, is easily shared, highly accessible, to date, and non-redundant. Enterprise supportive processes are highly synchronized with and connected to the development team.

2.4.2.4. Choreography

Lean Product Development *choreography*[3] describes team organization and process flow. Lean Product Development choreography is team centric and value oriented. Individuals and teams are load-balanced and operate synchronously, both within development groups and groups across the product architecture, upstream and downstream of development. Lean Product Development choreography is communication oriented, cross-functional, and enterprise interconnected, absent of value-less enterprise boundaries.

[3] The label "choreography" has been chosen to avoid confusion with the word "process", but also is an appropriate term for characterizing the organization and movement, or modus operandi, of individuals and teams for technical problem solving.

3. Product Development Context

Traditional Lean approaches to enterprise process improvements begin with stakeholder value analyses. Stakeholder identity and their various positions on value are a critical first step in the Lean tradition and transformation.

It is of great interest to draw connections from stakeholder value perspectives to complementary and appropriate modes of behavior for a development team. Of particular interest to this thesis are

1) the variability and influence of stakeholder value perspectives across enterprises and industries,
2) the heterogeneity of development approaches (Barney, 1996) used by organizations for value delivery in product development, and,
3) the outcomes of various unique development approaches – particularly, how well outcomes are aligned to the strategic objectives for value delivery of the organization.

Regarding stakeholder value, numerous factors contribute to value identity and delivery: the business environment in which a company operates and competes, the makeup of the company (including its team, resources, and culture), and finally key attributes of the product itself (such as safety and reliability). These factors collectively define the product development context in which the development team will organize to deliver stakeholder value.

Stakeholder Review

Abstractly speaking, product development stakeholders are grouped as agents, beneficiaries, and victims (Routio, 1999; Crawley, 2005). *Agents* are those who are responsible for the creation of the product – both directly as participants in the product development process, but also indirectly providing relevant supportive processes. Process participants, or direct agents, include product developers of various functional disciplines including marketing, engineering, system test, manufacturing, and operations. These participants are responsible for conceiving, designing, integrating, and delivering products to customers. Indirect agents include the enterprise and its various processes, its investors and shareholders, management, and suppliers. Indirect agents are responsible for incubating, funding and nurturing the product development process. Product development process *beneficiaries* are those who enjoy or profit from the process (Routio, 1999) and its outcomes (patents, knowledge or the product itself). Beneficiaries directly include customers, of course, but also indirectly other industries that may exist as a result of product success (for example, third party service and support) and potentially the communities surrounding them. Finally, *victims* of the product development process are those who experience loss as a result of the process outcome, either intended or unintended, at present or in the future (Routio, 1999).

Stakeholder Value Approach

Abridged attempts to capture stakeholder value perspectives (i.e. simple statements regarding profitability, compensation, or even, "deliver value to customers") produce generic, homogeneous goals for value maximization, can cloud short-term versus long-term value objectives, and can lead to false assumptions about stakeholder desires and intentions. These shortcomings do not inform the process designer well for a chosen approach.

Long-term stakeholder value perspectives, such as the shareholder's desire for profitability, are typically characterized from a product's full life-cycle perspective, either in terms of market share, profitability or return-on-investment. While full life-cycle perspectives are appropriate at the shareholder level, they are difficult to measure from the perspective of the development project, and consequently difficult to translate into goals for the development team. How an enterprise delivers on market share objectives or return-on-investment is a complex sequence of successfully meeting objectives on costs, market expectations and windows of opportunities, product performance criteria, sales goals, and so on. Meeting these various objectives are the shared responsibility of all enterprise processes, some acting in coordinated activities with the development team, but others acting perfectly independently. Within the development effort, profitability would suggest an eye on costs as the project proceeds, but also the quality of the product needs to be monitored (so that it would be desirable in the marketplace), and of course the availability/timing of the product to market. Independent of the development effort, success in the marketplace depends largely on sales channels, access to the market, etc. In summary, identifying and delivering stakeholder value is a non-trivial exercise that accounts for influences both upstream and downstream of the product development process. It is unreasonable to project broad strategic goals solely on the hands of the development effort, and therefore unreasonable to evaluate their performance with broad measures.

To successfully translate broad stakeholder value perspectives into "process value" (Chase, 2001), we must explore further the specific contribution of the process towards high level goals. This thesis advocates a thoughtful and deliberate exploration of the product development *context.*

> **context**[4]
> *-noun*
> 2. the interrelated conditions in which something exists or occurs: environment, setting.

A product development context evaluation reinforces stakeholder value identification with a holistic consideration of market conditions, enterprise, team and product attributes. With these considerations in mind, the process designer is led to restate stakeholder identities and comprehensively consider their needs and wants, but also to consider the experience and competence of the team, the resources available to them, dominant architectural elements of the product they are creating, and various market forces relevant in their industry. The context evaluation characterizes the setting in which value delivery will take place, and in which the development project is placed in motion. Ultimately, the choice of an appropriate process for delivering, and finally maximizing, stakeholder

[4] context. *Merriam-Webster's Collegiate Dictionary*

value cannot be practically answered without an investigation into context. Therefore, in order to complete a stakeholder value analysis, we first look closely at the context of the product development process.

Outline of Chapter 3

Chapter 3 attempts to address the context issue by formulating and testing a useful model for the characterization of the product development context.

In section 3.1 the context of the development process is constructed holistically, from a variety of perspectives, both internal and external to the enterprise.

In section 3.2 the dominant contextual components most relevant to the development process are presented in a context model for characterizing product development context.

In section 3.3 the context model is tested with a context analysis of two distinct product development organizations. Stakeholder identity and value preferences are revisited again at the end of the section, within the context of our two examples.

3.1. Four Perspectives for Context Characterization

> "...the process of innovation represents the confluence of technological capabilities and market-needs within the framework of the innovating firm." (Rothwell & Zegveld, 1985)

Rothwell (1994) presented a graphical framework of the product development process, shown in the figure below. The framework offers the process architect a clear representation of the various contextual influences surrounding the development process: market influences, technology influences, as well as internal factors such as enterprise capabilities. These influences are explored in following sections.

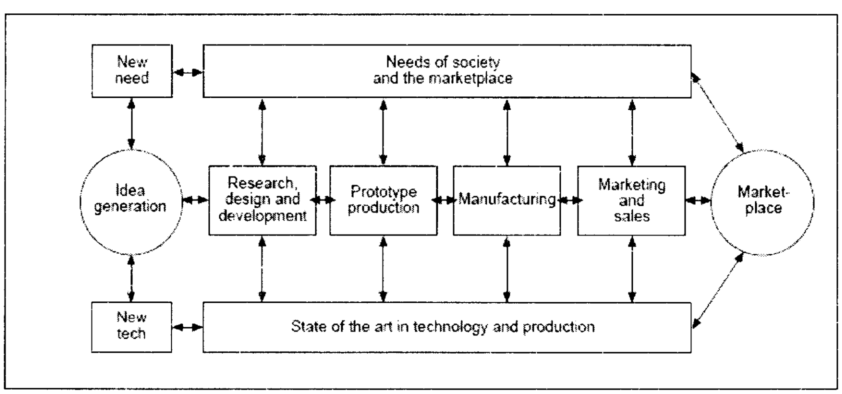


Figure 3.1 Rothwell's "*Third Generation Process*"

The sub-sections below review four relevant background perspectives for use in context exploration. Each offers unique insights. The perspectives – product-market, industrial organization economics, organizational theory, and systems engineering – offer the process designer a holistic framework in which to consider the development process and its strategic function in the enterprise.

3.1.1. Product-Market

The product-market perspective looks at the intersection of key product and market attributes and offers the process designer a broad notion of industry and customer base, and the nature of each. Hamel (2002) describes product/market scope as capturing the "essence of *where* the firm competes" (p. 73) in terms of its customers, geographies, and product segments. The product-market perspective is a point of departure for capturing the product development context.

Ulrich and Eppinger (2004), in their seminal work *Product Design and Development*, proposed product and market characteristics and suggested that, "the development process will differ in accordance with a firm's unique context". Eight product-market categories were highlighted in a table as follows: market-pull, technology-push, platform products, process-intensive, customized, high-risk, quick-build, and complex systems. Although Ulrich and Eppinger did not present their table as a formal model, such decomposition can aid one in attempting to understand the contribution of the product and its market when characterizing context.

3.1.2. Industrial Organization (IO) Economics Models

It is common in business literature evaluate forces internal and external to a firm to capture a firm's strengths, weaknesses, core capabilities and competences. These analyses are commonly used to develop strategies and to determine the likelihood of sustained competitive advantages. Various models are popular and commonly employed. While such models are typically used for evaluating a firm in its entirety, below they are revisited with the development process in mind, as opposed to their traditional corporate point of view. The perspective of the firm allows the process designer to begin to detail enterprise interaction with the market, and looks inside the enterprise to consider its capabilities and strategies for competing within the market segment.

Two models, Hamel's *Business Concept Innovation* (2002) and Barney's *Evaluating Firm's Strengths and Weaknesses* (1996), are proposed as useful for the formulation of product development context. In the first model, Hamel teaches one to "decompose a business concept" (p. 70) into four *components*: core strategy, resources, customer interface, and value network. Between the components are *linkages*: customer benefits, configuration, and company boundaries. Finally, four factors that determine *profit potential*: efficiency, uniqueness, fit, and profit boosters. The second model, *Evaluating Firm's Strengths and Weaknesses*, is used to expand on Hamel's *resources* component.

3.1.2.1. Strategy

Hamel's view of core strategy is that it offers general objectives and direction, a mission statement perhaps, or "big, hairy, audacious goals" (2002, p. 72). Corporate level strategy defines and sets broad expectations for the scope of products and markets, determines high level enterprise structure and goals, sets shareholder expectations (Crawley, 2005), and outlines the way in which a firm will attempt to differentiate itself in the marketplace. Enterprise strategies flow down from the corporate level to the business unit (narrowing in scope and in time as appropriate), on to departments, functions, and finally to individual processes.

3.1.2.2. Enterprise Resources

Commonly, a firm's resources are called into question when evaluating its strengths, weaknesses, and its ability to sustain competitive advantages within a given marketplace. Barney (1996) summarizes the resource and capability literature and presents four categories of resources: *financial*, *physical*, *human*, and *organizational capital.*

Financial Capital

Financial capital refers not only secured funding but also to the enterprise's access to funding, and various approaches to financing processes and projects. Enterprise processes are either internally or externally financed (monies borrowed, or paid for with revenues), publicly or privately, with deep coffers or lean budgets.

Physical Capital

Physical capital refers to the quantity and quality of physical resources owned by, or available to the firm. Physical resources include real estate (i.e. land, property, facilities) and infrastructure (i.e. laboratory and test equipment, CAE tools, IT infrastructure).

Human Capital

Human capital resources refer not only to the quantity of individuals (headcount) but also the qualities of the individuals employed. Human capital addresses the experience of individuals, and answers the question of whether or not the enterprise is appropriately staffed for pursuing the various strategic goals of the organization.

Organizational Capital

Organizational Capital refers to attributes of "collections of individuals" (Barney, 1996). Organizational capital includes the capability and maturity of all business processes (i.e. legal, sales, development, manufacturing) and also the connectedness of processes, and the efficiency (Hamel, 2002) of each in support of others.

3.1.2.3. Customer Interface

The enterprise's customer interface refers to its market access, or customer reach. The interface includes pre- and post-sales customer interaction: business development, marketing and advertising, as well as training, support and service. Of particular interest to the process designer are primary marketing mechanisms and their effectiveness, as the knowledge gained becomes critical to development efforts. The quality of the customer interface ranges from intimate to "arm's length" (or worse), depending on a wide variety of factors: supplier versus buyer power, past experiences, and strength of current channels and relationships.

3.1.2.4. Value Network

An enterprise's value network refers to its external network and various partners that "complement and/or support a firm's own resources" (Hamel, 2002). The value network extends both upstream and downstream of the firm along the value chain (i.e. suppliers, manufacturers) and envelopes all enterprise processes. All firms rely on a value network to some degree, varying from relatively self-sufficient to highly dependent on partners and consultants, but also outsourcing (even off-shoring).

3.1.3. Organizational Theory

Organization refers to the background organization of employees within the enterprise, specifically their reporting structure and peer groups. Organization is not a representation of specific approaches or processes used in an enterprise, but rather a structure and hierarchy of human and team resources. The organization of employees is relevant to product development context because it is a strong suggestion of cultural norms among individuals and teams (i.e. strength of management, team structure, modes of communication, and knowledge sharing and management).

The perspective of organization enhances the process designer's consideration of groups of individuals, particularly within the culture of their own enterprise and within the context of the products it produces. Knowledge of organization becomes critical as it either complements or contends with the formation of development teams, and the appropriateness of the structure of those teams for design and development of various product architectures.

Classically, organizations vary in the degree to which they are functional- versus project-oriented, with advantages and disadvantages for each (Allen, 2001). Functional organizations are appropriate for advancement of knowledge within specialized disciplines, while project organizations excel for coordination of inter-disciplinary teams. The trade-off is often referred to as a matrix-organization, typically with an emphasis on either generic form (Allen, 2001). According to Allen, critical factors to consider when forming organizations are product traits (such as complexity and technology cycles), project duration, and market dynamics.

Product Attributes versus Organization

Critical product traits are considered in the choice of team organization. The organization of each are highly connected, and "in many ways determine product success" (Crawley, 2005). Teams can either be matched to the organization of the product, or the product architected to suit the organization (Crawley, 2005, Ulrich & Eppinger, 2004).

Product traits, such as complexity, impact the appropriateness of an organizational structure. Complexity in product architecture determines the interdependence of specializations (Allen, 2001), that is, the luxury of functional specializations to work independently, or the need for functional specializations to coordinate efforts. Products of higher order complexity generally require more coordination among functional disciplines, thus, an emphasis on cross-functional teams.

Also considered is the rate of change, or clock-speed, of technologies used as it impacts the need for functional specialization (Allen, 2001). The shorter the clock-speed of technology (higher rate of change) the larger the precedent for the enterprise need for specialists, thus an emphasis on functional departments.

Project Duration versus Organization
The project duration should be considered as a factor for organization. Long-term separation of individuals from their specialized disciplines may lead to specialized knowledge management issues. Therefore, long term projects might better be functionally organized to keep individuals abreast of functional developments – otherwise developers are too long separated from their peers (Allen, 2001). Shorter projects are more suitable for project teams (long-term separation of individuals from their functional disciplines is a lesser concern).

3.1.4. Systems Architecture

Systems architecture is a complex discipline that addresses planned technologies and the *architecture* of the product system – specifically, its internal and externally delivered function, interfaces and form, processes employed to deliver value to users, and essential complexity (robustness) (Crawley, 2005). Attributes of product system architecture are perhaps the most critical in regards to product success, and the efforts put forth by the *product architect* to create good architecture should not be underestimated or trivialized in any way.

The responsibility of the *process architect* is to provide the development team with a suitable environment in which to deliver the engineering details of the architecture (i.e. feasibility analyses, detailed design, problem solving, prototype and test, and implementation details). As mentioned above, it has been argued (Ulrich & Eppinger, 2004; Allen, 2001) that various attributes of the product system architecture and the development organization should be aligned. In support of this notion, the *process* architect considers relevant *product* architectural traits, goals, and strategies such that the development process is created to support, nurture, and compliment them. Consequently, the product architecture must be *describable* by the process designer.

Two viewpoints on product architecture are proposed as suitable for characterizing product development context. Functional and non-functional (Lehto and Marttiin, 2005) viewpoints are selected as appropriate.

3.1.4.1. Functional Viewpoint

The functional viewpoint of product system architecture is that of its delivered benefits: what the product system accomplishes as deemed valuable to a user, and the composition of the architecture (objects and processes) employed to deliver said value.

The process architect must first achieve an understanding of what functional attributes contribute as quality attributes to the beneficiary – typically through a categorizing and mapping of needs and requirements to product functions. The mapping informs the process architect for strategies regarding prioritizing tasks, generating feedback, and detailed design. Second, the process architect must decompose product architecture into functions (processes, operations) and form (objects, modules, sub-systems). The process architect understands the manner in which function and form interoperate, react to external stimuli, and produce benefits. The functional viewpoint of product system architecture is typically non-trivial to describe and classify.

To fully realize the functional viewpoint of the product system architecture, the architecture is preferably communicated *by* the product architect, *to* the process designer, through the use of a modeling methodology. Modeling methodologies (Rechtin & Maier, 2000) are commonly employed by system architects, and range from very formal and extensive (DoDAF[5]), to software-oriented (UML[6]), to abstract (IEEE1471[7]). Physical, mechanical and information systems have commonly been modeled using object-process methodology (OPM[8]) or other capable tools, while software systems might be modeled using others (i.e. UML). Preferably, the product architect will communicate architecture at a high-level, but using object-process tools so as not too communicate the functional viewpoint too abstractly. The process architect must be fluent with modeling methodologies such that the architecture can be effectively communicated, visualized, and understood.

3.1.4.2. Non-functional Viewpoint

The non-functional viewpoint of product system architecture includes all architectural factors, issues, and considerations (the "ilities") not overtly related to customer value, but otherwise necessary for product deployment and success. Lehto and Marttiin (2005) reviewed five well-known architecture evaluation methods and synthesized an Architecture Evaluation Framework (AEF) which emphasizes non-functional considerations. The AEF is presented in the figure below.

---

[5] Department of Defense Architecture Framework (DoDAF). A requirement for DoD programs, intended to describe highly complex systems. SEE: DoDAF 1.5 Volumes 1-3, available at: http://www.defenselink.mil/cio-nii/docs.

[6] Unified Modeling Language (UML). A general purpose modeling methodology commonly used in software engineering. SEE: Fowler, M., (2003), *UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3rd Edition,* Addison-Wesley.

[7] IEEE-Std-1471-2000, Recommended Practice for Architectural Description of Software-Intensive Systems. SEE: http://standards.ieee.org/catalog/software4.html#1471-2000.

[8] Object Process Methodology (OPM). A framework for the study and development of systems. SEE: http://www.objectprocess.org.

1. System/Product Characteristics/Capabilities
   - 1.1 Reliability
   - 1.2 Availability
   - 1.3 Fault Tolerance
   - 1.4 Performance
   - 1.5 Capacity/Coverage
   - 1.6 Functionality
2. Product Creation Excellency
   - 2.1 Testability
   - 2.2 Flexibility
   - 2.3 Commonality
   - 2.4 Simplicity
3. Logisticability
   - 3.1 Ease of Installation
   - 3.2 Serviceability and Maintainability
   - 3.3 Learnability
4. Affordability
   - 4.1 Costs
   - 4.2 Timeliness
   - 4.3 Environmental Issues
5. Industry Acceptance
   - 5.1 Standardization
   - 5.2 Business Horizontalization
   - 5.3 Control Over the Product

Figure 3.2 Architecture Evaluation Framework[9]

Non-functional factors are proposed as contributing to product development context in that they coax the process designer to consider the extensive range of technical issues the team will address throughout the lifecycle of the development process.

3.1.5. Section Conclusion

The perspectives presented in the sections above were chosen as relevant perspectives for exploration of the environment surrounding the product development effort. It became evident while researching these perspectives that a variety of forces were contributing to define this environment. The forces, broadly categorized into four main groups, or categories, could be used explicitly in the form of a model for the purpose of characterizing the context of a development effort.

First, the product-market perspective inspired a *Market* category, which captures the various forces related to marketplaces and customer identity, including the nature of different marketplaces based on product technologies and systems prevalent therein. Second, the industrial organization economics perspective motivated an *Enterprise*

[9] adapted from Lehto & Marttiin (2005)

category, which captures enterprise strategy and planning, but also includes a comprehensive view into enterprise resources, which will support the development team in its efforts. The view that human resources are critical resources also inspired a *Team* category, which captures the experience, attitudes, and motivations of individual team members. The resource perspective also enriched the Market category, with access to customers and customer relationships evident as a contributor of context.

Next, the organizational theory perspective contributed organizational structure, leadership, and the inter-relationships of enterprise supportive processes to the Enterprise category. This perspective also compounded the importance of individuals and Teams, particularly in relation to the *Products* created (thus, the final category). Finally, the systems engineering perspective fully informed the Product category with considerations including architectures, complexity, robustness and reliability. In the following section, these four categories are formally presented as a useful model for characterizing the context of a product development effort.

3.2. Components of Product Development Context

A context model has been created and is proposed as a tool for use in context characterization. A hierarchical representation of the model is presented in the figure below. The complete model is presented in table format and is included in Appendix A. Four main categories of contributing factors to the product development context are outlined in the model: ***Market***, ***Enterprise***, ***Team***, and ***Product*** attributes. In all cases, context is considered from the point of view of the product development organization and the processes used to bring a product to market.

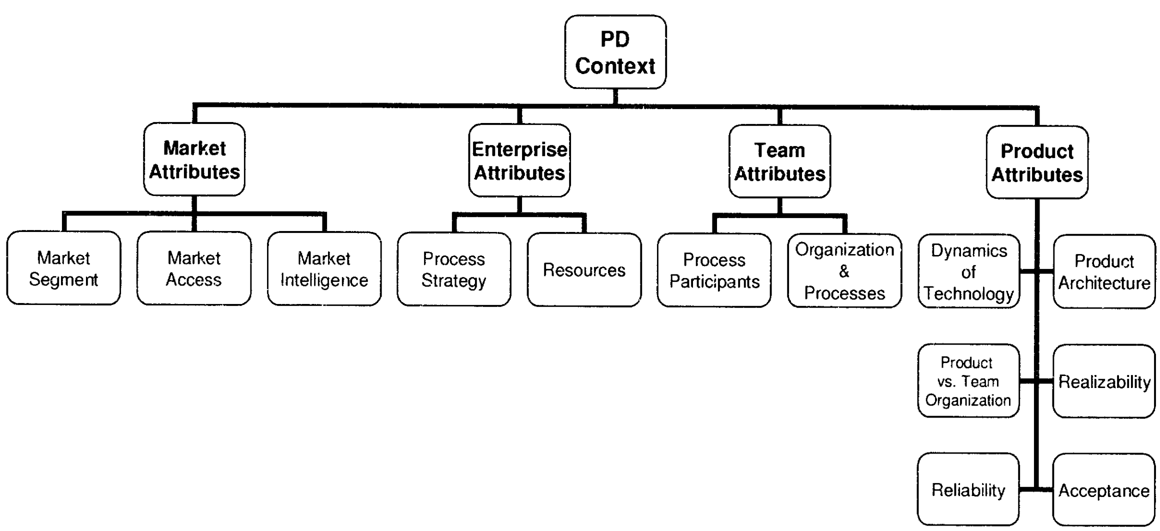


Figure 3.3 Model for Context Characterization

The following sub-sections explain the cells in the context model in detail. In the final section of the chapter, the model will be used to analyze the context of two distinct product development efforts.

3.2.1. Market Attributes

Market Attributes include considerations regarding the existing or potential customer base: the nature of product development in a particular market (formality, bureaucracy), points of contact and interaction with customers, and knowledge of customer value.

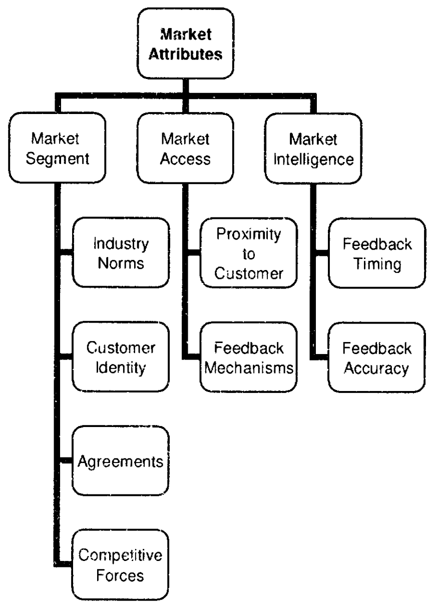


Figure 3.4 Market Attributes of Context

3.2.1.1. Market Segment

Market segment refers to the identity of potential or existing customers and the nature of doing business with them defined by the industry at large. Considerations of a market segment include *industry norms*, *customer identity*, the types of *agreements* common between customers and suppliers, and finally various *competitive forces* in the market. These factors combine to define the cadence ("clock speed") of a marketplace, and therefore the cadence required of the development project.

Industry Norms

The industry of interest brings to bear certain common practices with regard to quality, reliability, and cost and availability, which typically influence the level of bureaucracy and formality commonly practiced. These norms set the stage for doing business in the industry, and influence standard modes of behavior and interaction for consumers and suppliers. Industries of interest include high quality, or price-sensitive industries: medical products, military/aerospace contracts, or high-volume consumer electronics.

Successful participation in such industries mandates that the development team and the greater enterprise follow not only formal regulations, but also standard practices and cultural norms.

Customer Identity

The enterprise segments a potential market, within which the product development effort is set forth targeting one or more potential customers. The process architect must be aware of customer identity, their needs, and the positioning of the product in relation to satisfying those needs.

The process architect explores customer identity to build a broad picture of the target market. Customers are either end-users ("B2C"), integrators/OEMs ("B2B"), or even customer-developers (i.e. open-source, open-interface). Products are marketed as strategic, operational, functional or perhaps as emotional solutions. Examples include improved productivity, improved working conditions, reduced expenses, or enabled services. Within the customer segment, what individuals are typically involved in purchasing decisions? How does the product satisfy the variety of needs of the various decision makers? Analyzing customer identity also reveals the size of the target market, ranging from one to many customers. The size of the market is a suggestion of the expected demand, whether the development team expects low-volume (or even one-off, custom made products), medium, or high-volume production. It is also possible that a product development team works in advance of a known market, for example in technology-push products, or simply in traditional research and development projects.

Agreements

Of vast importance to product development context is the nature of *agreements* made between supplier and consumer. Various types of agreements are common, ranging from high-formality contracts to handshakes. Agreements contribute to define the cadence of the market in that they suggest a degree of access to the customer, can impact the intensity of competition (i.e. competitor lock-out), and generally determine a schedule of milestones to market (Crawley, 2005).

In many industry contexts, contract/bidding processes are commonly employed where vendors submit proposals in response to requests-for-proposals (RFPs). Proposals are reviewed as a down-selection of vendors is made, followed by additional rounds of bidding, and finally award of contract. Proposals can include existing products (off-the-shelf components), product concepts (proposed solutions), or both. In this manner, contracts can be awarded at various stages of development: in early stages, after product concept, system detail and specification; during intermediate stages, after initial prototyping and development; or at later stages, products are simply manufactured in fulfillment of an order.

Without formal agreements, products are produced ahead of sales, without commitment from target customers. Vendors may decide to develop products in anticipation of market needs, based on trends, conversations, and handshakes. In such contexts, vendors may experience less access to the market, with more risk due to competitive forces.

Development teams may decide to proceed with more focus on costs, or speed to market, to hedge against perceived competitive threats.

Competitive Forces

Intensity of competitive forces contributes to the cadence of the marketplace – at times, the pressure to get to market as quickly and efficiently as possible as the project proceeds. Intensity of competition ranges from highly competitive to "competitor lockout", and is influenced by the nature of agreements, number of competitors, supplier power, and market size. Will existing products be targeted and undercut from competitors? Will current development efforts be used to strategically undercut competitors? What are the realities of the competitive landscape - one competitor, many competitors, any competitors?

3.2.1.2. Market Access

Market access refers to the nature of the team's interface with its customers and/or beneficiaries: specifically, the team's *proximity* to customers, and the avenues and *feedback mechanisms* by which market information is relayed to the development team.

Proximity to the Customer

A development team's proximity to its customers impacts the probability of customer interaction with the process (team, iterations) and the probability of interaction with the product (prototypes). Direct access to the customer occurs when designers and developers interact with the customer first-hand, through meetings or demonstrations. Many enterprises have a long-standing history with customers and can exploit such relationships for direct and constant feedback and needs gathering. Direct access is common at one or many stages of development: concept, early design and system specification, intermediate design and development stages (iterations, perhaps), late development stages, and production stages. Types of direct access include conference calls, testing or interaction with prototypes, product demonstrations, lab trials, qualifications, and post-sales support. Indirect access to a market is also common, through marketing or sales contacts, but requires that needs are interpreted. A system specification can be considered a form of indirect access, or interpretation, particularly when such specifications can misrepresent customer desires. Alternatively, the development team may not be able to predict when market interaction will occur. In such contexts, customer access arises at random intervals when opportunities present themselves through marketing efforts or through existing sales channels. Finally, while undesirable, customers may be inaccessible for long periods, during which time the team must proceed with very limited contact. This situation is common in new ventures whose relationships have yet to be established. Disconnects between developers and users are an important component of context due to the high potential for mismatch of the product's features and performance with customer desires.

Feedback Mechanisms
Resulting from proximity, various feedback mechanisms are possible. Feedback mechanisms refer to the individuals and processes used to capture and record information from the market, and affect the quality of the customer's voice, the interpretation of such information, and likelihood of value delivery. Feedback can be recorded first-hand by the team, or interpreted or translated through indirect contact. The mechanisms used to capture and interpret feedback drive information quality, discussed below.

3.2.1.3. Market Intelligence

From the point of view of the development project, market intelligence refers to the quality of market information available to the team and consequently the stability of requirements over the course of development. Stability of requirements depends on the timing and accuracy of customer desires as perceived by the development team. Timing and accuracy of information is related to the market access discussion above, but additionally considers transitioning markets and the sureness (or fickleness) of customers themselves. The combination of timing and accuracy defines the level of information quality (Bauch, 2004) where earlier and more accurate are linked to fewer product changes (Rothwell, 1994).

Timing of Access and/or Feedback
The nature of the customer interface affects the timing of requirements and feedback from the market, another critical aspect of product development context. The process architect considers how often, and at what stage of the development process (concept, design, implementation, or operating phases) requirements and information are collected from the market. For example, in a contractual situation, perhaps customer interaction is or can be regularly scheduled. Timing of market intelligence affects the stability of requirements, which is addressed in the next section.

In a "market-pull" ideal, requirements are stable throughout the development project. In this context, the market is well understood and there is an opportunity for the team to quickly pursue technologies to complement market needs. Conversely, the later preferences are known to the team, the greater the risk of moving forward due to the likelihood and impact of changes on the project. Unfortunately, early information is often difficult, even impossible, to ascertain in many product development contexts.

In addition to the absolute timing of requirements, equally important is the relative rate of change of the market as compared to the timing of the development project itself (Allen, 2001). Transitioning markets typically require that development ensues with risk that portions of the product architecture will be replaced during the course of product development.

Accuracy of Access and/or Feedback
The process architect considers the accuracy of customer needs and feedback, as poor information quality is a key source of waste in product development activities (Bauch, 2004). Highly accurate market data affords the development team confidence to proceed

with minimal assumptions, providing opportunities for process efficiency. Conversely, inaccurate customer preferences and feedback – related to "lost in translation" effects due to indirect access or poorly translated needs – create risks for the project, and the less likely market needs are satisfied.

Similarly, the process architect considers the fickleness of the market. The less certain (more fickle) are customer's preferences, the greater the risk of moving forward due to the likelihood and impact of changes on the project. Unfortunately, accurate information is often difficult, even impossible, to ascertain in many product development contexts.

3.2.2. Enterprise Attributes

Enterprise Attributes are critical components of product development context as the product development process exists within the enterprise. Enterprise considerations include process strategy, organization, and the resources and infrastructure available (or unavailable) to the development team. As knowledge of the enterprise is vital to the product architect (Crawley, 2005), it is also vital to the process architect.

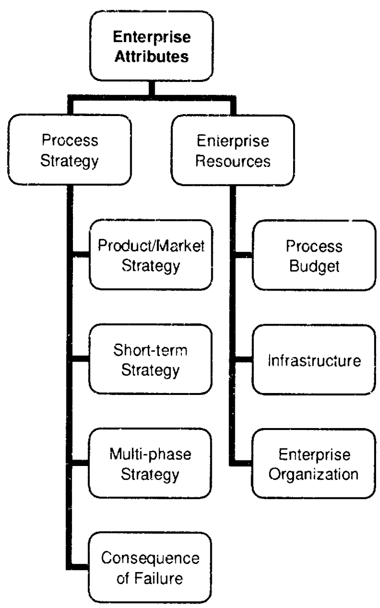


Figure 3.5 Enterprise Attributes of Context

3.2.2.1. Process Strategy

Corporate strategy flows down to business unit strategy, which in turn flows down to functional strategy. Ultimately, through a process of narrowing focus and scope, a strategy exists for the development process itself. Process strategy, very much like corporate strategy, is closely related to shareholder and corporate value perspectives.

Process strategy is a translation of stakeholder (indirect agents) expectations for the development project and the team's performance.

Product/Market Strategy

Certain product-market strategies easily translate into process strategies, and therefore are difficult to decouple from the development process context. For example in brand strategies, brands with differentiating characteristics like product or process excellence (i.e. high-reliability, longevity) burden the product development process with proportionally high degrees of testing and verification, and an intense level of coordination with production. Alternatively, in certain contexts the strategic purpose of a product development is merely as an enabler of other products and services. In this context, a product may be non-differentiating, non-profitable, or even sold at a loss with the expectation that other products would consequently sell in volume.

Short-term Process Strategy

In many contexts, enterprises have the luxury of easily determining process strategy due to industry norms where efforts are inherently shorter-term, or simply with manageable scope. A multitude of short term strategies are possible. Examples of short-term process strategies include risk reduction, mind-share, or cost reduction.

Multi-Phase Strategies

For complex or robust products with relatively long development cycles, projects are typically decomposed into smaller projects and/or into phases – each with its own strategy. For example, an initial feasibility phase might be followed by feature enhancement ("planned iterations"), and ultimately, cost reduction. Without decomposition (rare) the team is left with one-chance to get it right, or a "one-play" strategy (bad strategy). Ironically, it is likely that too narrow (short-term) of a focus will introduce risk (Allen, 2001). Generally, a mixture of short and long term strategies is appropriate. Are we trying to hit a window of opportunity, are there many more windows open in the future? Will we have an opportunity for cost reduction in future iterations?

Consequence of Failure

A final aspect of process strategy is the penalty of failure. If the process fails, does the company lose a customer, or, is another opportunity possible? Is it better to miss a deadline and get the job done right? What are the considerations for brand and reputation? When will the development project, or the company, run out of funding? The higher the costs of failure, the higher the pressure will be to succeed. High pressure is a critical contributor to context.

3.2.2.2. Enterprise Resources

From the perspective of development processes, enterprise resource analysis considers the various resources available to a technology company and its product development team. Resource analysis is of interest for development process context in that it provides a humble and honest assessment of the capabilities and competencies of an organization generally, but more importantly how well capabilities are aligned with specific process objectives.

In large companies with vast resources, perhaps many options/strategies are available to a development team. For smaller companies with lesser resources, resource analysis may only uncover limitations. Development teams must evaluate their current capabilities, exploiting or adapting them where necessary. Where resource gaps exist, future (planned/budgeted) resources are applied.

Development Process Budget

The process architect considers the monies available for project funding: the source of funding and amount, and how close to monitor costs as the project proceeds. The sources of funding for a project are either internal or external, contractually agreed to between buyer and supplier (at times with special stipulations), backed by a healthy stream of revenues, or funded within a single round of private financing.

Certain contexts are not heavily constrained by funding, spending is liberal and even encouraged (typically, in order to speed things up). In other contexts, projects proceed with extraordinarily few financial resources. The process architect must be aware of finances such that resources can be applied to fill in gaps where necessary, also the general ability to compensate team members and provide incentives.

Infrastructure

Infrastructure refers to the various infrastructural items available for the design and development process. Items to consider are space (office and laboratory) and equipment (design tools, test equipment, computing capacity, assembly equipment). Of special interest is the sophistication of information systems, which affects the team's access to information, information quality, and their ability to share information (Bauch, 2004). Such an inventory of physical capital reveals an enterprise's *infrastructure readiness* - can the enterprise provide the team the tools and resources required?

Enterprise Organization

The process architect considers how well the enterprise is organized for the development project – internally, extended, and externally. Enterprise organization includes management techniques (effectiveness and influence), project structures, knowledge management approach, and the quality of and coordination of other vital supportive business processes. The process designer considers the size and complexity of enterprise bureaucracy, and the formality of internal processes ("red-tape"). Of particular interest is the efficiency and coordination of the linkages among the development process and supportive processes (Barney, 1996), business-units, test centers, and evaluation centers. Supportive processes for a development project include marketing and product management, technical management (functional, project), systems integration and test, and operations (procurement, materials tracking and inventory, production).

3.2.3. Team Attributes

Team attributes include the qualities of individuals (process participants), their organization, and processes.

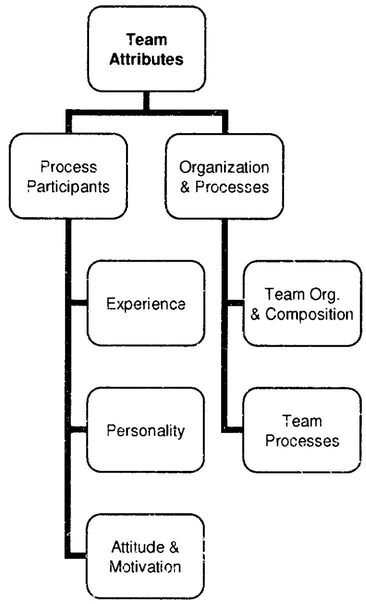


Figure 3.6 Team Attributes of Context

3.2.3.1. Process Participants

Process participants refer to the individuals employed by the enterprise who are cooperating and coordinating efforts in order to achieve the product development. Process participants are the direct agent stakeholders of the process (Routio, 1999) as defined above. Process participants include the conceivers, architects, developers, and engineers currently employed as part of the team, and those who are anticipated to join the team over the course of development. Qualities of individuals on a development project include their experience, personalities, motivations and attitudes (willingness to communicate and help others). The qualities of process participants are critical aspects of context and are of considerable interest to the process designer.

Experience

The process architect considers the team's experience as it contributes to quality-, cost- and time-to-market, and as a (presumed) measure of competence and productivity. Factors include functional expertise (specializations of skills) or generalized expertise, or, in certain contexts, an appropriate diversity of experiences (Rechtin & Maier, 2000). Where a learning curve is required, the process architect considers the energy that must be expended (time, dollars) and the approach necessary to educate and prepare the development team. Ultimately, the competence, experience, and expertise of the team as relates to the development task at hand must be considered as a fundamental component of the development context.

Personality

When considering individuals as a human capital resource (Barney, 1996), one should not ignore personality traits of employees. The preferences of individuals to work alone (as *specialists*) or as part of a team (as *platform experts*) are tightly coupled with complex, albeit classic, personality traits (introversion versus extroversion). (In fact, the Myers-Briggs Type Indicator has been used to identify preferred personality type for team membership, NT – systematic and strategic analysis (Rechtin & Maier, 2000, p.47).) Perhaps our participants are more productive on their own, contributing more to their functional discipline than to the greater system. Perhaps our participants are more suitable for implementation than for research. Perhaps our participants can play both roles – not only make contributions within their unique discipline but also act as an integrator, a cross-functional participant, or even a core team leader. In this manner, personality traits are considered when structuring teams, considered relative to key architectural attributes, such as partitioning, interfaces, and modularity.

Attitude & Motivation

Equally important to the personality of individuals is their attitude and motivation. Motivation deals with the *capacity* of individuals to retain a focus and dedicated to the task at hand. While highly dependent on various intrinsic and extrinsic factors (Herzberg, 1968), motivation of individuals ranges from lethargic to high-energy and is critical to process success. Attitude is a general characterization of approach and cooperativeness with management and the team, or a *willingness* to maintain interpersonal relationships with team members, and if necessary, to reach outside of preferred comfort-zones. Perhaps our participants are able to mentor others well, to share knowledge, and to teach. It will be evident later that information sharing, cooperation, and coordination are critical socio-technical characteristics of high-performance teams (Rechtin & Maier, 2000). Those in key integral "architectural" positions will ultimately play an equivalent integrative role from a social perspective (Allen, 2001).

3.2.3.2. Organization & Processes

Team organization and processes are considered due to their suggestion of culture, relationship to product architecture, and impact on knowledge sharing and management.

Team Organization & Composition

Team organization considers the alignment and division of teams, both *logically* and *geographically*. The logical organization refers to team structure, either highly functional or with strong project emphasis (or a variation of each). Geographical organization includes collocation versus highly dispersed teams, and perhaps even psychological separation of individuals (i.e. "the wall" that commonly divides individuals and functions in engineering organizations). Also a factor is the team's composition, including the coordination of teams from different business units, distinct enterprises, and reliance on contractors, sub-contractors, and outsourcing.

Of special consideration is the appropriateness of team organization and composition for the *current* strategic goals of the enterprise. Is the team structure relevant for today's goals, or was it created for past projects? Product development organizations must evolve with their current products, technologies and trends, or risk biasing development projects within outdated paradigms (Rechtin & Maier, 2000; Ulrich & Eppinger, 2004) or exaggerating the wrong competence (Barney, 1996).

As stated above, the organization of the development team must be considered with the organization of the product architecture in mind. Some critical attributes of the product architecture are explored in detail in the Product Attributes section below.

Team Processes

Team processes refer to the current processes employed by the development team in their approach to product development. Processes range from formal to informal, as a matter of culture or rigid documented practice. Processes are traditional or progressive, iterative, spiral, waterfall, or no processes at all. Processes and approaches can be inherited, adopted, or self-taught. In general, development processes must be adapted (as should all enterprise processes) over time as the challenges of markets and technologies evolve. Ultimately, the culture and competence of the processes used as relates to the development task at hand must be considered as fundamental components of the development context.

3.2.4. Product Attributes

Product attributes are a critical component of the product development project context, contributing to quality, cost and schedule norms within an industry and marketplace. Considerations include the dynamics of planned technologies, a product's key functional architectural traits (i.e. components, processes, complexity), and non-functional traits (i.e. manufacturability, reliability, and serviceability).

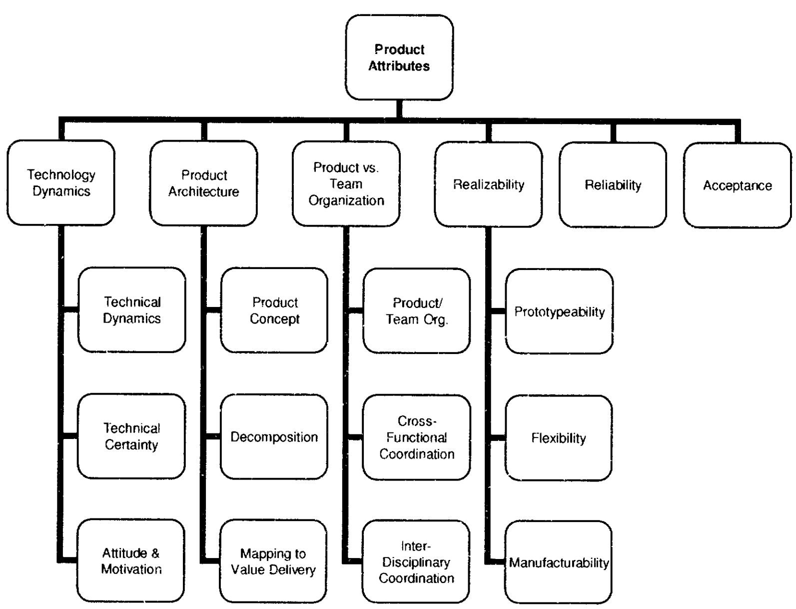


Figure 3.7 Product Attributes of Context

3.2.4.1. Dynamics of Planned Technologies

Technology dynamics affects products and organizations in rapidly evolving industries and compares the maturity and availability of technologies to the ability and experience of the enterprise to adopt them. Technology *stability* (questioned from the perspective of planned technologies) addresses whether technologies are quickly evolving or relatively static. Technology *readiness* (considered from the perspective of the enterprise) refers to the enterprise's faculty and adeptness to use and incorporate planned technologies.

Enterprises attempt to go to market with *new* products, or to *incrementally improve*, or simply even *sustain* existing products. For new product developments, teams will either enter into existing markets with established technologies, or will "push" technologies into the marketplace. In either case, the development team may have to undergo a period of learning while adopting and developing new technologies. For incremental changes to products, an enterprise may exploit past experiences through reuse of physical components or knowledge and information (code, schematics)? In the variety of these instances, technologies can be fairly mature to certain companies, and entirely new to others.

Technical Dynamics

Technical dynamics refers to the rate of change, or fluidity, of technology. Rate of change of technology is most critical as it is compared to the length of the development cycle (Allen, 2001), affecting the likelihood that the technology will evolve during the course of development. Similar to varying market requirements, dynamic technologies may require that developers reconsider and replace portions of the product architecture during the course of product development. This occurrence is possible in rapidly changing technological industries, or perhaps in industries with long development cycles, such as aerospace systems.

The extent of the impact of transitioning technology on the development project is a function of the product architecture, specifically its transformability/flexibility, which is discussed in subsequent sections, but also the agility of the process.

Technical Certainty

Technical certainty refers to the feasibility of technology – certain technologies are those that are known to a team and considered low-risk, while uncertain technologies are considered relatively higher-risk from a team's perspective. Certain, low-risk technologies include platform technologies, incremental improvements, or other "same-generation" products with which a team is generally familiar. Other certain technologies may include market-pull, where existing technologies are available and simply matched to customer needs.

Uncertain technologies are those that are relatively higher-risk for a particular development team, where feasibility has yet to be determined. Examples of uncertain technologies are technology-push, innovative, displacing, or exemplar (first-mover) products. Extremely uncertain technology, or high-risk technology, may add dimensions of testing, tracking, and management to the process – or, may add a whole new stage (R&D, risk reduction stage). Also, several options may be pursued throughout the process. In practice, development teams are typically exposed to new and old technologies.

3.2.4.2. Product Architecture

Basic product traits affect the structure of the processes, tools, and individuals – the approach – to create them. The evolution of the processes used in many industries has been steered by predominant architectural characteristics of the products themselves.

High-level Product Concept

The product concept answers questions such as "what is it?" and "who will use it?" When considering the product concept, the process architect uncovers the basics of product architecture: hardware versus software, embedded systems, web or network applications, physical product or a service.

The product concept also provides a suggestion of complexity and ambiguity, including modes of operation and variety of systems within which the architecture operates. The challenge to the process architect, is deciding whether or not these complexities or ambiguities are relevant for the team.

Decomposition of Function and Form

The process architect considers key architectural characteristics of the product – specifically, dominant functions and forms, their linkages and interaction. Functions include processes and operations, while form includes objects, components, modules, and subsystems. Decomposition of form reveals the product's size, and what materials, parts and assemblies are used. Is the architecture monolithic, modular, or hybrid? Is the product organized as a *platform*? Decomposition of function reveals how do the components interoperate and communicate. Form and function are connected at various interfaces. Are the components highly interconnected? Are interfaces standardized? Are they proprietary?

Architectural complexity is a broad topic that highlights many product characteristics contributing to product development context, either individually or collectively. Key characteristics of complex architecture are the numbers and size of components and interfaces, high-interconnectedness, modularity and flexibility, and the number of processes (dependent and independent) both internal and external of the product architecture. Architectural complexity is a critical component of product development context in that it is a dominant driver of cost and schedule risk precedents for a project, and is commonly a factor when forming teams.

Mapping of Architectural Attributes to Customer Value Delivery

Consideration of the product architecture addresses the notion of *value proposition* - "what are the high-level needs that are satisfied by the product?" The process architect must understand (for emphasis, prioritization) how key elements of function and form combine to deliver particular value to beneficiaries.

Of particular interest to the process architect is product *operability*, specifically a product's various user-interfaces. The operation of the product once deployed, either by human or by machine, is a key factor of product acceptance, and therefore market success. Human interfaces are critical for general performance preferences (usability, intuitiveness) but also for ergonomics (comfort, safety). Example architectures with key human-interface considerations are laptops, automobile cockpits, and web-design (GUIs). An extreme example of a user-interface is that provided to a co-developer in collaborative designs (Rechtin & Maier, 2000), where the user-interface is actually a development interface (common in open architectures and the internet).

In addition to the human aspects of operation, machine interfaces are also considered. Similar to standards compliance above, machine interface design is critical for interoperation with other devices, computers, infrastructure and appliances. Also considered are service interfaces (replacements, revisions, upgrades) which are a form of, and perhaps as critical as, traditional user-interfaces.

3.2.4.3. Product Architecture versus Team Organization

There are several impacts of the level of product complexity on the development process and how complexity impacts the appropriateness of team organization. Considerations include the logical team organization, inter-disciplinary coordination, and cross-functional coordination.

Product/Team Organization

> "In partitioning, choose the elements so that they are as independent as possible; that is, elements with low external complexity and high internal complexity." (System Architecting Heuristic, Rechtin & Maier, 2000)

The process architect considers the partitions of architecture and attempts to select complementary partitioning for the team. Considerations include logical partitioning of functions internal to modules and subassemblies, but also physical interfaces between them.

System architects strive for modular architectures which are desirable for flexibility. Such architectures are designed to accommodate evolution (such as technology evolution) and change, and commonly allow for independent, simultaneous efforts among individuals and teams. Of particular interest is the expectation that iterations are either likely or desirable during development. Are teams of individuals able to iterate independently of other teams? Are platform teams aligned with platform products?

Cross-Functional Coordination

The complexity of product architecture affects the level of coordination required between cross-functional groups to conceive, design, engineer, integrate, and test. Cross-functional coordination refers to the necessity for different development functions operating together as part of the development team. Cross-functional coordination occurs at the design and engineering phase of product development amongst functional design and development groups such as electrical, mechanical, and software engineering and integration.

Inter-Disciplinary Coordination

Interdisciplinary coordination refers to the upstream and downstream coordination necessary – mfg & supportive processes coordinating with the development team. Equally as important are functions upstream (marketing, systems planning) and downstream (manufacture), but also parallel support processes (materials planning and procurement, operations, and facilities).

The influence of product complexity on the importance of inter-disciplinary coordination is exemplified by special considerations that affect the manufacturability of a product. Complex products are typically difficult to manufacture, and therefore pose considerable risks to the quality of the product after manufacturing, and the cost and time to complete the manufacturing phase. Considerations include intensiveness of process, production volume, specialization required, size-of (very large, or very small) (Crawley, 2005). In many industries, product architecture is constrained by standard processes and materials used in manufacturing (Ulrich & Eppinger, 2004). In other industries, manufacturing processes are limited by the characteristics of the product itself.

3.2.4.4. Realizability

In certain contexts, the size and complexity of a product constrains (costliness, risk) the team's ability to prototype and manufacture. Really small, really big, or really complex products are typically costly to realize. Consider the cost to "tape-out" an application-specific integrated circuit (ASIC), or the intensity of production process of major construction projects such as long-span bridges or skyscrapers. For these architectures, the risks (schedule and costs) associated with *realizing* the product architecture, or even the simple opportunity, dominate the context of the development approach.

Prototypeability

Implementation and assembly considerations are typically associated with manufacturing and production, but also impact the difficulty and likelihood of building prototypes for test and iteration. Various product attributes (i.e. size, complexity) affect the ability of the development team to prototype the product architecture quickly and successfully. Considerations include the time and cost to simulate and prototype, which in turn impacts the *time and cost* to iterate. Iterations and prototyping stages are desirable but not always practical. How likely are you to build multiple iterations after factoring in schedule and costs? Options are desirable, but how expensive is it to consider multiple options/versions simultaneously?

Flexibility of Architecture

> "Be Agile. Architectures should be modular, reusable, and decomposable to achieve agility. Architecture descriptions should consist of related pieces that can be recombined with a minimal amount of tailoring to enable use for multiple purposes. An agile architecture provides the means for functioning in a dynamic environment." (System Architecting Heuristic, DoDAF, 2004)

Modularity, flexibility, and agility are commonly used to describe architectures that accommodate *change* – requirement changes, or otherwise. As stated above, iterations are desirable but not always practical. Architectures vary in the degree in which even updates and revisions are feasible. How transformable is the product architecture? Alternatively, how does architecture constrain iterations? What is the impact of requirements changes, and at what stage of the process?

Manufacturability

In certain architectures and industries, production operations are extraordinarily specialized processes. In commercial airliner production, for example, the concept of design-for-manufacturability is so interwoven with design decisions that it practically drives the design and development process. Boeing revealed planned strategies[10] and goals for their development and production processes of commercial airliners: design for manufacturing (DFM) critical, streamline production methods, enhance quality, enhance efficiency, and finally, "an ambitious goal of reducing final assembly to three days".

Drivers of manufacturability are elements of complexity (i.e. size, difficulty), time to manufacture, capital intensity, and production volume. The higher the complexity of the architecture, the higher the quality, schedule, and cost risks associated with the manufacturing process. In high production volumes, the impact of complexity and poor design decisions begins to scale.

3.2.4.5. Reliability

The quality goals for the product play a key role in context, particularly extraordinarily high quality, or "ultraquality" (Rechtin & Maier, 2000), standards (i.e. brand quality, mission, and safety critical systems). Brand quality products are produced with high quality for differentiation (strategic marketing) of a particular brand (i.e. Mercedez, Lexus). Mission critical architectures include various aerospace, military, or naval architectures. Safety critical architectures include life-support, medical devices, and emergency protection systems. Other architectures with high quality standards are high

[10] Boeing's Annual Report to Shareholders, 10-K 2006

durability, long service lives, and high-availability systems (i.e. nuclear power plants, satellite systems and telecommunications).

Testing and verification of high quality systems is typically costly and extensive. Planners weigh options for the volume of testing, methods used, and coverage (the degree of conformance to specifications, and possibly the extent to which specifications will be exceeded). Has the product been architected to accommodate testing?

3.2.4.6. Acceptance

Regulations, governmental or industry enforced, are in place typically for safety (either human, machine, or environment), but also to stimulate industry, protection from competition, or for other incentives. Governmental agencies requiring compliance to regulation include those at the international, federal, state and municipal levels. Regulation at the industry level comes usually in the form of standardization, which can also be required by customers and consumers directly. Compliance to standards is usually desired for interoperation among common industrial devices, but can also be desired by (or forced on) others due to pervasive products already in operation. Both regulation and compliance can also be a function of internal company standards.

Related to testability, compliance to standards and regulations is typically certified internally, by a customer, or by a third party. The process will require cooperation and coordination between the producer and certifying agency, consuming both time and resources.

3.3. Variations and Similarities of Context

In this section the context model is applied to unique product development efforts at distinct organizations. The contexts of these distinct efforts are synthesized and contrasted.

3.3.1. Analysis Overview

The purpose of conducting the case study interviews was to test, critique, and improve the context model with authentic development projects. The initial case studies were conducted using hindsight projects, so that the model could be calibrated from its initial incarnation at the conclusion of literature review and research.

3.3.1.1. Selection of the Case Studies

The case studies were selected from industry such that development contexts would provide useful insights, particularly when compared and contrasted with one another. Candidates were chosen based on their experience developing processes and their perspective within their respective organizations. The contexts for the case studies were chosen for their value as contrasting industrial examples. Case Study A was representative of a privately funded, entrepreneurial startup atmosphere, in a commercial data communications context. The context of Case Study B was quite the opposite – a well-established, large market-cap, formal environment typical a modern-day aerospace systems integrator. Accessibility of the candidates became critical after it was decided that a discussion format was most appropriate for application of the context model (in lieu of a questionnaire). The interviews proved to be lengthy, approaching 5-6 hours each to complete, and many more hours to compile and analyze.

3.3.1.2. Interview Approach

It was deemed appropriate to conduct interviews in an open-discussion format. The cells presented in the summary table of the context model (Figure 3.3) are not meant to be a complete taxonomy of contextual factors. The cells are designed merely to stimulate thought and generate discussion pertinent to the unique circumstances of the process architect's development context. In this manner it seemed appropriate to conduct interviews in a discussion format (ideally, in-person).

The context model was used to explore the context of the candidate's development process by addressing the contents of the cells of the table, in succession wherever possible. Initially the contents of the current cell were briefly addressed, after which the table was set aside such that an unstructured discussion might follow (which was not biased by the contents of the cell). Once the attributes of the current perspective were exhausted, the cells of the summary table were revisited in succession until all of the perspectives in the summary table were addressed. The contributing factors to product development context are highly interwoven and interdependent. Consequently, no attempt was made to limit the conversation to a single cell in the table as the interview progressed through the various Attributes sections.

3.3.2. Case Study Interviews

3.3.2.1. Case Study A: Plexis GbE Transport Development

Case Study Interview A investigated the product development context of the Plexis GbE Transport product platform. The Plexis interview was conducted with the former director of engineering of Optinel Systems, Inc., over two 2½ hour sessions in April, 2008. The first session took place on April 3rd and covered stakeholders, Market Attributes and Enterprise Attributes of the Plexis context. The second session took place on April 8th and covered Team Attributes and Product Attributes. The details of the interview are presented in Appendix B. Updates made to the model as a result of uncovering details of the two contexts are presented at the end of this chapter.

Interview Notes

The interview was conducted in-person, in a discussion format. The various perspectives in the summary table of the context model (Appendix A) were presented in sequence, discussed briefly for the purpose of stimulating dialogue, while cautiously avoiding bias. During discussion, the table was temporarily set aside so that the discussion relative to the pertinent cell could be recorded. Once the attributes of the perspective were exhausted, the cells of the summary table were revisited in succession until all of the perspectives in the summary table were comprehensively addressed. After the interview, the data was compiled and examined for the purposes of critiquing and improving the table. Deficiencies in the table were addressed with updates, adjustments, or deletions.

As the interview commenced, it was rather awkward to simply "identify stakeholders". Stakeholder identity is not a trivial matter, particularly for an interview candidate who had not been considering stakeholders at length. In any event, assembling a list of stakeholders proved lengthy, and the utility of such an exercise at the onset of the interview seemed unclear to the interview subject. At the same time, the discussion at this point in the interview was in high gear as many of the contextual attributes in the table were being discussed – that is, the question of stakeholder identity was, in hindsight, a useful icebreaker for the interview.

As the conversation transitioned to context, it was again difficult to keep stakeholders in mind explicitly. The value of identifying stakeholders at this stage of the process lies mostly in the extent that they contribute to the context of the development effort itself. Later, value delivery within this context is explored, at which time stakeholders will be revisited for the purpose of identifying appropriate goals for the development effort. Still, as stakeholder identity contributes to context, it is therefore proposed as a useful first step in the analysis process.

No attempt was made to limit the conversation to a single cell in the table as the interview progressed through the various Attributes sections. Also, it was difficult to predict which direction the discussion might proceed, relative to the sequence of the cells. For example, following the summary table sequentially leads the discussion through team factors before discussing product factors. In the Plexis context, the product evolution was extensive such that the corresponding impact on the team was significant. The product context actually significantly affected the team organization, their experience, and relevance. It was difficult in conversation to avoid the product evolution before discussing the team, but also difficult for the reader to follow the same sequence in writing. It is not expected that the same difficulties will be observed in all contexts, and also unlikely that a single sequence would be sufficient.

Administered as a discussion forum, the interview comprehensively explored the uniqueness and the specifics of the Plexis context. While the discussion format was time consuming and exhausting, the interview achieved a high degree of coverage for the Plexis context. It seems unlikely that a questionnaire could be designed to stimulate the richness of discussion achieved in the Plexis interview.

3.3.2.2. Case Study B: ProjectX Satellite Telemetry Subsystem Development

Case Study Interview B investigated the development project context of ProjectX, an electromechanical controller and satellite telemetry subsystem. The ProjectX interview conducted in a discussion format on a conference call, over a single 4½ hour session with a program manager on April 27th, 2008. The details of the interview are presented in

Appendix B. Updates made to the model as a result of uncovering details of the two contexts are presented at the end of this chapter.

Interview Notes

The ProjectX interview commenced with an updated context model (Appendix A), so that project stakeholders and project description were addressed up front as a part of the formal model. The model was received well and the stakeholder discussion was somehow less awkward than it was during the Plexis interview (before stakeholders were added to the model). As before, the initial discussion quickly evolved into context as attributes of the market, enterprise, team and product were quickly confronted in relation to stakeholders.

Similar to the Plexis interview, various perspectives in the summary table were presented, discussed briefly for the purpose of stimulating dialogue, and temporarily set aside to allow the discussion to flow and to facilitate record keeping. As before, discussions moved quickly from topic to topic, confirming the notion that contextual factors are highly interrelated and are likely to be addressed in relation to each other.

3.4. Section Conclusions

3.4.1. Reflections on Product Development Process Context

Characterization of product development context is a mindful process that requires an appreciable investment in time and energy. Reflecting on the chapter, it is clear that the question of context is complex: markets evolve, customers are fickle, individuals and teams are diverse, technologies are dynamic – the permutations of which are numerous. The sheer diversity and complexity of contextual factors is such that homogeneity of context is highly unlikely spanning across products and industries, but also within products and industries, and in fact within a single organization! A large enterprise with multiple simultaneous projects for example, with even marginally diverse customers, will certainly have different teams, products, budgets, or simply even projects in different phases of development (unique short-term strategies).

Product Architect as a Stakeholder of the Development Process

The product system architect creates a broad definition of the product architecture and provides this to the development organization. The development team *delivers the architecture* to the architect, to the extent that it is predefined. Where design details are open, undecided, or ambiguous, the development team is responsible for making determinations regarding feasibility, practicality, and costs. The product architect is a development stakeholder in that the product architect relies on the process for delivery of broad concepts, but also for resolution of architectural details.

Project Context

In the manner described above, the exercise on product development context advocated in this chapter is really not a process viewpoint after all, but rather is a *project* viewpoint. Using the context evaluation, the process architect evaluates the context of a *specific* product development effort, the *project context* – market, enterprise, team, and product factors – not all of which are applicable to the entire product development organization.

The complexity of contextual factors validates the need for a holistic evaluation. Any single viewpoint of product development context might prove too narrow, leading to homogeneous perspectives of the development effort. In summary, a holistic view of context, or *project context*, results in a properly characterized development effort. Within this context, the process architect will ultimately attempt to prescribe a development approach (resources, team structure, management philosophy, and processes). Similar to context, heterogeneity of approaches is probable as well.

Looking ahead towards the remaining chapters of the paper, perhaps a single prescription for product development for an entire enterprise is inappropriate. In lieu of publishing a single, common, enterprise process for all projects, an organization might employ any of a variety of approaches, depending on the unique circumstances of a given development project. Perhaps an enterprise could formalize a host of possible processes, allowing a project manager or technical leader the discretion to select the most appropriate for his or her current project needs. This view compounds others' (Ulrich & Eppinger, 2004; Allen, 2001) that enterprises should strive for flexibility in their approaches to product development processes.

3.4.2. Updates to the Context Model after Validation

After reflecting on the research, the context model was updated to Project View instead of a Process View, as discussed above in section 3.4.1. Additionally, several updates were made to the context model after at the conclusion of the case study interviews. The updates are itemized below. The context model as presented in Appendix A is a final draft that incorporates all of the updates, improvements, and deletions.

3.4.2.1. Updates Resulting from Case Study A: Plexis GbE Transport Development

Stakeholder Identity and Project Description

At the time of the interview, the context model did not explicitly include stakeholder identity and project description sections. The interview was conducted such that these topics were addressed prior to introducing the context model, but not included as formal discussion points. It became quite obvious that these discussions were necessary leading into the context table. After the interview, the "context model" was improved to become *a model*, and was positioned downstream of leading blocks where stakeholder identity and project description would be addressed. Using the model, the process architect would now be reminded to precede the context evaluation with a deep-dive into stakeholder identity and project description.

Updates to the Summary Table

The enterprise, and indirectly the Plexis team, was subject to a high degree of uncertainty and variability caused by the inconsistent actions of the board of directors. The board's fickle behavior, particularly regarding budgeting and headcount, was both distracting and disruptive to the product development process. The notion that uncertainty occasionally originates *internally* was overlooked in initial drafts of the summary table. Consequently, the table was updated to include *enterprise dynamics* (Enterprise Attributes/Resources section).

At times it was difficult to limit discussion to Plexis without addressing the team's history with previous products, customers, and with each other. Historical contexts were particularly solicited when addressing the question of currency (enterprise and team organization), where discussions went beyond to organization to include team dynamics and experience with transitioning technologies employed on past development efforts. The simple context of "how people get along" was not addressed in the table initially. It was decided that *relevant historical* contexts were pertinent in several categories of the context model.

During the discussion on employee motivation at the company, it was interesting to note that while enduring a degree of constant uncertainty – including several instances of reorganization of the team, changes in technical direction where roles were questioned and altered, even outright elimination of jobs – the Plexis team marched along with high levels of energy. It seemed particular to the startup atmosphere that a high energy level was sustainable despite circumstances, and particularly unlikely that similar resiliency would be common in many other contexts. It was concluded that the *expectations of stability* (Team Attributes/Process Participants/Attitude & Motivation section) were unique to the context of the company. While uncertainty (particularly job-stability) inhibits motivation in all contexts, perhaps startup teams are more resilient to uncertainty than others based on expectations regarding work-life when individuals agree to join a team. Conversely, also considered is a general *resistance to change* in an enterprise.

In the context of startup it became clear that several issues related to the power of the company's suppliers were highly influential to the Plexis team. The power of suppliers is of course argued in industrial-organization economics models (i.e. Porter's *Five Forces*), but was overlooked in initial research. To accommodate both powerful and weak supplier contexts, the context model was updated with the sub-section *supply-chain* (Enterprise Attributes/Resources section).

During discussions regarding the cross-functionality of the Plexis development team, the Director recalled an individual who was contributing to development in several functional disciplines (hardware, software, and optics). The project of course benefited greatly from this individual's diverse qualifications, but a larger impact was made, that was greater than the sum of his contributions. The cross-functional contributions of this individual helped to blur functional lines in the organization, greatly contributing to the fluidity of the team. Consequently, it was decided that the context model should be expanded to include the concept of a *difference-maker* (a bit of a sports analogy – MVP, rule-breaker, clutch player – but also possible in the negative as well). The perspective of the difference-maker is intended to characterize individuals whose contributions (either highly positive or highly negative) lead to extraordinary team performance.

The Plexis team, like many others, faced tough decisions regarding selection of components, in fulfillment of architectural requirements, particularly costly components (pluggable optics modules, data processing ASICs). The Product Attributes section of the context model did not explicitly include *high-risk components* (quality, cost, availability).

As a result of the company's size and general lack of resources, but also its willingness to expose team members directly to the market, the Plexis team was highly involved in product deployment, service, and maintenance. Many product development teams are asked to support current and previous products currently in test, manufacturing, or in the field. The level of distraction due to direct developer involvement in product lifecycle processes *outside of development* (i.e. installing, servicing, sustaining), of past or current products, is a critical factor contributing to product development context. The context model was updated to suggest these supportive processes explicitly, such that the process architect will consider the extent to which the development team might be required to participate in lifecycle processes outside of development.

The importance of leadership as a resource was explored by Hamel (2002). Initial revisions of the context model experimented with the subject, but leadership was not explicitly addressed at the time of the interview. In the context of the Plexis development effort, the strength of leadership of the President/co-founder was not only a source of motivation, but actually a source of stability in an otherwise unstable environment. In the unique context of the company, this individual was highly visible to the development team, while not formally a technical project leader. In larger organizations, it is likely not the case that the company president is as easily accessible. Therefore, the Team Attributes section of the context model is extended to simply include *team leadership*, which is meant to capture leadership qualities of individuals in a variety of roles, as a critical factor in product development context.

3.4.2.2. Updates Resulting from Case Study B: ProjectX Satellite Telemetry Subsystem Development

Updates to the Context Model

Many facets of the ProjectX architecture were largely predetermined by systems specifications, particularly external interfaces, as is common in many product development scenarios. In addition to having knowledge of the product architecture generally, the process architect must understand what subset of the system architecture is predefined by the product architect, whether or not the product architecture is flexible or negotiable, and what remaining subset of the architecture is to be determined by the development team. To account for architecture that is required or mandated of the design team, the Product Attributes section was updated to include required architecture. What has, and what has not been predetermined by the product system architect?

The study of ProjectX was really a story of conditions before, and then after the replacement of the program manager at the subcontractor, and the subsequent restructuring of the project team. The impact of the arrival of the lead software engineer on the software team, and the strength of leadership of the program manager, reaffirmed the earlier notion that strengths of key individuals were critical and that certain individuals could make extraordinary contributions. In ProjectX though these two difference-makers brought wisdom and experienced to the team, with more consequence than motivation. The Team Attributes section was updated to include difference-maker as a factor of experience, instead of attitude & motivation.

As ProjectX was a relatively small job for the subcontractor, the project was poorly staffed at the onset and treated as a low priority internally. Not until the project was far enough behind schedule, such that the prime demanded a staffing change did the subcontractor turn its attention and resources towards ProjectX. To capture this important aspect of context, the Enterprise Attributes section of the model was updated to include relative strength of project as a component of project strategy.

In this context, developers are required to produce much more than just physical product. For example, the software development team was responsible for nine software documents (i.e. SDD, SRS, ICD, etc.) as formal deliverables to the prime. Such a large volume of documentation takes many, many hours of engineering man-hours to generate. The Team Attributes section of the context model was updated to include process formality, and required deliverables.

ProjectX was subject to an extraordinary level of scrutiny as applied by the management team (risk mitigation, cost, and schedule tracking). This management scrutiny drove prioritization of tasks, meeting attendance, and generally consumed time and energy. To capture this or other nuances of management behavior, management techniques was added to Team Attributes.

ProjectX was conceived as a strategic project for the prime, and consequently was underbid in an effort to win the business. Once the contract was secured, the prime had to exert a high level of schedule pressure on the development team. As the interview candidate revealed, "There is real tension in the industry between business acquisition and business execution. Proposals are written by people who disappear when it comes time to do the work. Also, if you bid at no risk, you'll never win a contract." This tension is true at every layer of the sub-contracting hierarchy. To better capture this context and others that inevitably result in extraordinary pressure on the development team, estimation of scope, cost, & schedule was added to the agreements section under Market Attributes.

4. Goals and Objectives for Stakeholder Value Delivery

Stakeholder value delivery is explored both through traditional means and within the context analysis as described in the previous chapter. The results of the analysis are expanded upon in the sections to follow.

4.1. Background

4.1.1. Product Development Objectives

Arguing in support of formalized development processes, Ulrich and Eppinger (1995) proposed "Goals of a Structured PDP" (Table 4.1).

Customer-focused products
Competitive product designs
Team coordination
Reduce time to introduction
Reduce cost of the design
Facilitate group consensus
Explicit decision process
Create archival record
Customizable methods

Table 4.1 Ulrich & Eppinger's *Goals of a Structured PDP*

Inspection of Ulrich and Eppinger's goals reveals a variety of objectives for value delivery. Stakeholders are explicitly represented: *market* focus, *enterprise* strategic objectives, *team* coordination and learning. The product is of primary concern: reduced cost- and time-to-market, customer and market driven focus. Process capabilities are suggested: flexibility (customizable methods) and efficiency. Finally, enterprise learning and knowledge management are goals of the process as well. These and other goals are explored further in this section

Product versus Process Objectives

Chase (2001) clearly distinguished between product and process value, suggesting that value is delivered separately through two distinct mediums. The purpose of this section is an exploration of stakeholder value delivery primarily through process, with the intention to synthesize process goals and objectives (as opposed to product goals) for value delivery.

4.1.2. Stakeholder Value Delivery

The product development organization aims to deliver and ultimately maximize value to its stakeholders. The following sections briefly revisit the literature to explore stakeholder value delivery without a deliberate exploration of product development context.

4.1.2.1. Value Delivery to Process Agents

Value Delivery to Direct Agents

Recalling the work of Benabou and Tirole (2003), and Herzberg (1968), value is delivered to employees by the greater enterprise as well as through participation in the development process itself. The enterprise contributes primarily Herzberg's *hygiene* factors: compensation and benefits, advancement, and growth. Alternatively, *motivators*, also a critical component of employee value, stem from the day to day work-life experience (what we really do). Therefore, process designers have a valid opportunity to impact (positively, or otherwise) employees' job-quality through process attributes that affect short- and long-term employee motivation and satisfaction. The process should provide extrinsic motivators, such as rewards and recognition, but also stimulate intrinsic motivation by challenging individuals to contribute to problem solving in a nurturing environment. Ideally, participation in the development process results in a stimulating and rewarding experience for employees.

Value Delivery to Indirect Agents

As stated in previous sections, it is common in the literature to express shareholder value to indirect agents (investors, shareholders) in purely financial terms with an emphasis on profitability. Recall that Chase (2001) argued that purely financial metrics such as profitability are awkward as short-term success indicators, but also fail to measure process outcomes serving other needs. Subsequently, Chase (2001) identified *process value*. Clearly, the product development process serves other needs of the organization, for example, to develop and capture knowledge, or to capture "mindshare" (if not market share) by way of a customer demonstration or a tradeshow presentation. A wide variety of strategic needs are possible. In summary it is argued, here as well as elsewhere in this thesis, that enterprise and shareholder value delivery through product development is highly dependent on context, and should be decomposed through such.

4.1.2.2. Value Delivery to Process Beneficiaries

Reconsider Slack's (1999) simple but representative expression for Customer Value:

$$\text{Customer Value} = \{\sum(N \times A) \times f(t)\} / C$$

where,

N = a product feature to satisfy a need (as determined by the customer),

A = the ability or effectiveness of a product feature to satisfy the need (determined by how well product development processes are executed),

$f(t)$ = the availability of the product or feature (lead-time), and,
C = cost of the product or feature to customer (a function of product attributes as well as process efficiency).

Working from Slack's equation, the question of customer value delivery results in *process goals* to maximize each of NxA (in $\sum(NxA)$), manage $f(t)$, and to minimize C. From a process point of view, the ability to deliver highly desirable product features (NxA) is a function of delivering on the details of predetermined systems specifications (assuming they are in line with customer expectations) and reducing any ambiguity due to partially defined or undefined requirements. Equally important are adjustments and refinements to product features where specifications incorrectly represent needs, and to identify new features as necessary where needs are not addressed at all.

Product lead-time ($f(t)$) should be considered relative to customer's expectations, which in many contexts does not suggest minimization of lead-time at all costs. Rather, effectively managing product availability suggests an understanding of what the timing constraints are (windows of opportunity, defined by market), and planning to meet them accordingly. Finally, acquisition costs (C) are determined by a variety of factors, primarily a function of the costs of *designing* and *producing*, but equally as dependent on competitive and strategic business factors, and specific terms of agreements with customers.

4.1.2.3. Stakeholder Value Analysis and System Objectives

Rebentisch et al. (2005) prescribed the use of object-process methodology (OPM) "to derive exploration system objectives". Object process diagrams were used to capture stakeholders, needs and value delivery through possible system processes. An objective statement was derived for each diagram which "allows us to flow from objectives through concepts to specific implementations" (Rebentisch et al., 2005). A similar method for generating process objective statements is proposed at the end of this chapter.

4.2. Using Project Context to Determine Goals

Stakeholder value analyses as presented above are useful as a departure point for broadly defined objectives. To lead the process architect along a focused path towards value delivery, the following sections explore stakeholder value delivery within the various contextual scenarios captured in the previous chapter. The synthesis of product development goals using context analysis will help the process architect to more concisely target value delivery and maximization to various stakeholders.

4.2.1. Goals and Objectives from Market Attributes

The product development process must generally conform to industry norms, particularly in highly formalized and regulated industries. Participation in the government and DoD contracting environment, for example, tends to drive goals related to requirements tracking, acceptance, extensive testing, and integration. Agreements made with such customers will drive the schedule of milestones, mandating deliverables on specific dates. In some contexts, the processes may even be contractually *governed by* the customer. Value is delivered when the process is tailored after industry norms and various expectations of the customer.

Market access is a valuable commodity that should generally be exploited at every opportunity, independent of phase of development. Upstream, the creation of valuable products and technologies requires early and accurate information from customers through dialogue with development staff. During and downstream of development, customers must interact with prototypes and technologies for scrutiny and feedback. To preserve the customer's voice, the process should be designed to maximize the accuracy of information, whether from direct or indirect sources. The recording, interpreting, and general knowledge management of market data should be evaluated and reevaluated to the highest possible extent.

4.2.2. Goals and Objectives from Enterprise Attributes

Strategic contexts introduce a diversity of objectives and goals for value delivery. Value delivery requires design and delivery of architectures which meet or exceed expectations for brand quality and cost, and that allow the enterprise to market, advertise, and demonstrate products, and to develop business. Additionally, the process occasionally must deliver architectures in environments of extraordinary pressure (scope, schedule).

The process architect delivers value through knowledge of and access to resources within the organization. The architect must determine the availability and appropriateness of resources, and communicate resource gaps to the enterprise. The process must utilize resources efficiently, and ultimately the process and the team must adapt to the provided infrastructure.

4.2.3. Goals and Objectives from Team Attributes

The process architect considers the strengths and weaknesses of individual team members, with the objective that the productivity of individuals should be maximized. The architect considers the placement of individuals based on strengths and weaknesses related to "soft skills", experience (skill sets and specific domain expertise), personal preferences, and project needs (product architecture). The process is created with a focus on individual learning and awareness of technology and market factors relevant to the product system architecture. Finally, the process architect strives for general employee satisfaction by providing a stimulating and rewarding environment as a goal of the product development process.

The process architect strives for the appropriate logical organization of the team and must attempt to overcome any psychological or physical team boundaries, removing barriers to team cohesiveness and coordination. The process architect attempts to anticipate the many sources of organizational and personnel change (internal and external factors) and strives for organizational flexibility and ability to adapt.

4.2.4. Goals and Objectives from Product Attributes

The process architect considers planned technologies, determines where technical risk lies, and plans as necessary. Risk mitigation might be achieved through feasibility study and analysis, and team education. Value delivery requires anticipating and planning for potential product system architectural changes resulting from uncertain components and modules, component or module failure, or from general changes of focus which occur over the course of detailed design and problem solving. The architect strives to accommodate transitioning technologies, through planned and unplanned iterations.

The process should be appropriate for the product (i.e. hardware processes for hardware product, software processes for software products) without biasing the product architecture. The process architect must organize the team structure while considering the product architecture and its complexity, promote coordination and communication with inter-disciplines, cross-functions, and supportive processes, where necessary. The process must deliver architectures as predefined by systems architectural specifications and requirements (i.e. features, functions, ultraquality requirements), but also must resolve design details where not specified, including user interfaces (must complete the architecture).

4.3. Formulating Product Development Goals & Objectives

A variety of generic categories of process goals and objectives for stakeholder value delivery are presented in this section. The categories were selected based on the contextual analysis as described above, and stem from research and from the case study interviews as presented in the previous chapter. Process goals and objectives are either categorized as lifecycle goals (desirable for the course of the development project), or as objectives relevant to specific phases of a project. The project phases, *upstream* (early development), *midstream* (full development), and *downstream* (late project stages), have been only broadly defined for simplicity. This approach has been taken purposely to avoid any rigid borders or definition (or debate) of the phases of the product development lifecycle. The reader may feel that certain goals might be applicable to more than one stage, or a lifecycle goal only applicable to certain project phases. Again, the categories are generic, and are presented only as a suggestion of the vast array of possibilities for product development goals, and are not meant as a strict, complete taxonomy. Alternative representations are entirely possible. Finally, not all of the categories itemized below are appropriate for all development contexts.

4.3.1. Lifecycle Value Delivery

4.3.1.1. Enterprise Connectedness

Coordination with Vital Supportive Processes

The product development process is dependent on a number vital supportive enterprise processes, i.e. procurement, operations, and human resources. The development process should be highly connected with the enterprise, such that vital supportive processes are fluently and exhaustively operating in support of a project's ongoing and potentially dynamic needs.

A critical supportive enterprise function (not frequently mentioned with regards to product development) is human resources. A shared responsibility of the HR department and of functional and project management, the human resource function is involved in the location and selection of key individuals for team composition, but also for ongoing employee support and maintenance such as job satisfaction, motivation and career development. Over the course of a development effort, a variety of internal and external factors contribute to product development's dependency on human resources, many of which are suggested here in this thesis. Internal forces include changing enterprise strategies (reorganization, headcount reductions) or changes in product technologies and product direction. External forces include dynamic market preferences. Additionally, high pressure contexts such as extraordinary scope and schedule pressure are common. The dynamics of product development environments require adjustments to team composition over time, including additional or loss of team members, or changes in development leadership. Changing team structures and project direction affect individuals' work preferences and norms, and roles and responsibilities can become ambiguous. A goal of the product development process is to be intimately connected to the human resources function for appropriately fulfilling project needs with regards to new talent, and maintenance of the existing team with regards to roles & responsibilities, motivation and job satisfaction, and career development.

Cross-Functional Communication and Cooperation

The product development team should strive to operate without organizational boundaries, particularly between functional engineering departments: i.e. electrical, software, and mechanical engineering. This goal is appropriate for collocated as well as geographically dispersed teams. Value is delivered when a high level of communication and cooperation occurs among product developers of distinct functional disciplines. Where organizational boundaries exist, the development process strives to negotiate them successfully.

Inter-Disciplinary Communication and Cooperation

The product development team should operate without organizational boundaries, particularly between upstream and downstream lifecycle processes: i.e. marketing, development, integration and test, and production. This goal is appropriate for collocated as well as geographically dispersed teams. Value is delivered when a high level of communication and coordination occurs between product developers and process participants upstream and downstream of the development task. Where organizational boundaries exist, the development process strives to negotiate them successfully.

High Coordination with Manufacturing

Many product architecture contexts require intimate coordination between product development and manufacturing personnel. Anytime actual production processes are such that product features are constrained, or, product features are such that the production process is constrained (Ulrich & Eppinger, 2004), communication between producers and developers becomes critical to the success of the project. Product size and complexity (Allen, 2001), requirements for specialized production operations, and production volumes are all drivers of the level of coordination required. When appropriate, a goal of product development is to achieve an intimate level of coordination and cooperation with production personnel throughout the product development lifecycle.

4.3.1.2. Process Capability

High Process Efficiency

Recalling the discussions of the previous chapter, product development teams are subject to a variety of contexts with regards to resources: unlimited resources, finite but adequate resources, or less than adequate resources. The product development process should utilize enterprise resources (financial, physical and organizational resources) efficiently. With few exceptions, value is delivered to stakeholders when the product development process meets quality, cost, and schedule targets while consuming the fewest resources. Therefore, a goal of the product development effort is to operate with high efficiency and productivity.

Accommodate Pressure/Urgency

In product development organizations in competitive industries high pressure scenarios are common – many with the future of the business unit, or even the company itself at stake. The success of such product development efforts requires high process output and efficiency of operations, and indeed relies on the extra effort of individuals: technical leaders, developers, testers and builders. The development process should be architected in such a way as to accommodate and respond to escalated levels of pressure and urgency.

Value Management & Process Transparency

Tracking of development project progress and costs over time is highly valued by management and the greater enterprise (Chase, 2001). Project tracking is critical for planning activities related to market timing and approach, project continuation or cancellation, make versus buy decisions, calibration of estimation techniques (cost, schedule), and finally for general management insight into departments and teams so that resources can be evaluated and adjusted, if necessary. The process architect understands the concept of value tracking and management, and the value it provides to the enterprise.

The development process should be transparent with regard to *progress* of feature development, feature test, failure and resolution, system integration and test, environmental testing, and so on. In certain contexts, the ability to monitor certain aspects of progress is exceedingly desirable. For example, in cases of ultra-quality and ultra-reliability enormous effort is typically spent tracking quality progress and testing. The development process should also be transparent with regard to the *costs* of both process and product. In many contexts the desire to monitor costs is also particularly desirable. In new ventures and in contractual development scenarios alike, the ability to track the "burn-rate" of the development project is highly valued by the enterprise.

Risk Management & Mitigation
A goal of the development process is careful and deliberate identification and management of risk, risk mitigation, and if possible, risk elimination. The process architect strives to reduce risk (quality, cost, and schedule) wherever possible. It is possible in many contexts that *riskful* behavior is desirable or necessary (and possibly encouraged). In all circumstances, the process strives to provide the opportunity and resources necessary to effectively manage risk.

Enterprise Learning and Knowledge Management
A goal of the development process is to facilitate individual and enterprise learning through knowledge management. The process architect should provide mechanisms for careful and efficient collection, storage, analysis and dissemination of information.

Process Flexibility
The process architect strives for flexibility in process design. Value delivery occurs when processes quickly adapt to changes stemming from technical, market, enterprise, and team dynamics.

Process Scalability
The process architect strives for process creation such that processes scale as necessary to "handle the job". Commonly, product development projects and teams are subject to increasing scope as design and development activities proceed.

4.3.1.3. Acceptable Process

Acceptable Process – Market Perspective
In most industry contexts, buyers and suppliers adhere to customary or standardized practices and processes. Customer expectations of process range from highly structured (i.e. CDR, PDR for highly formal processes) to a minimal set of process deliverables. The process architect must create processes befitting an industry and that are acceptable to customers, particularly in formal contexts. A goal of the product development process is to negotiate milestones carefully and to operate within customary or standardized boundaries.

Acceptable Process – Product Perspective
A goal of the development process is that it should be designed in such a way as not to influence the product architecture itself. Too often product architectures are adversely

biased by the structure of development activities, either through outdated team structures, tools or processes. Individual and team experiences with legacy products, and with each other, combine to create organizational cultures that act as determinants of product and project direction. Value delivery occurs when the process architecture does not bias the architecture of the product.

Acceptable Process – Enterprise Perspective
Product Development serves the enterprise in that it delivers the product architecture in a timely and efficient, yet sustainable manner. Value is delivered to the enterprise when a high level of output is maintained by the development team, including contexts with shortened schedules and increased scope. High pressure contexts can require extended hours and overtime, extra travel, or can simply increase work related stress and anxiety. Preferably, high pressure contexts are short-term duration, but in many contexts commonly last for extended periods. The process should be architected and operated such that the development team maintains a sense and awareness of urgency, but with sustainable practices such that adverse affects on team members and product quality are mitigated appropriately.

Acceptable Process – Participant Perspective
Product development is the concerted effort of individuals whose motivation and satisfaction are paramount in regards to the quality and timeliness of the work produced, and the ultimate success of the development project. The value of the contributions of individuals is never underestimated by the process architect. A goal of the development process is to deliver value to participants by providing a rewarding and motivational experience resulting in long-term employee satisfaction.

4.3.2. Upstream Value Delivery

4.3.2.1. Upstream Market Goals

Reduce Market Uncertainty
Where markets are unknown or unspecified, the development process should strive to determine, or pull, market needs. Similar to goals related to feedback on product architectures, early and often access to the market is preferred.

Reduce Market Ambiguity
Where markets are ambiguous, the development process should strive to verify that original interpretations of customer needs are valid, remove any remaining ambiguity from architectural or systems specifications, and identify needs and deliver features where systems specifications fall short.

Maximize Information Quality
High quality market information as obtained by, or conveyed to, the development team is a goal of the product development process. The process should strive to improve the accuracy of the team's perceptions with regards to customer preferences and desires.

Value is delivered when the customer's voice is carefully and deliberately recorded and interpreted by the development team.

4.3.2.2. Upstream Team Goals

Participant Education – Market Awareness

A goal of product development is to educate its participants on the market, both of the specific desires (static or evolving) of known target customers, and current market and industry trends. Market education, unlike technical training, is not commonly practiced in product development organizations.

Participant Education – Technical Proficiency

Uncertain technologies, whether cutting-edge or existing technologies that are new to a development team, require learning curves that impact project performance to market. Value is delivered to stakeholders when the process strives to reduce learning curves and generally improve the technical proficiency of the development team. Technical proficiency includes product technology and standards, but also compliance and qualification practices related to customer and industry acceptance.

Individual/Product Structure

Recalling product architecture discussions of the previous chapter, critical characteristics of product architecture impact the appropriateness of certain individuals for specific roles on the development team. Individuals' traits related to experience, personality, attitude and motivation should be considered in regards to assignment of roles and responsibilities. The process architect should be aware of the unique strengths and weaknesses of individual team members. Value is delivered when developers and staff are placed in roles where individual performance is maximized.

Team/Product Structure

A goal of the development process is an appropriate emphasis on matrix organization (relative strengths of functional versus product structure) as required by the product architecture. Value is delivered when the team structure is aligned well with the product architecture, once understood.

Specialized Expertise

Team composition must include individuals with specialized skills, functional or otherwise, as necessitated by the product architecture. When learning curves cannot be overcome (particularly when time will not permit) the process architect must secure key individuals with appropriate specialized experience, and place individuals in key roles on the team.

Minimize Perceived Complexity

The process architect strives to achieve a degree of decomposition of complex projects. Projects are decomposed into manageable stages and phases, along several possible parameters: time, scope, resolution of detail, risk mitigation, and short-term strategy are examples. The project should be perceived as highly manageable from the perspective of an individual product developer's role and responsibilities.

4.3.2.3. Upstream Product Goals

Reduce Technical Uncertainty
A goal of the process is reduce uncertainty related to the feasibility of new or planned technologies. Feasibility, or risk reduction activities include research and development, prototyping for proof-of-concept and test, modeling and simulation. Extreme uncertainty, or high-risk technology, adds additional dimensions of testing, tracking, and management to the process. Value is added when technical uncertainty is resolved in early phases of development activities. Where feasibility can not be determined early, several options may be left open throughout the development process.

Minimize Product Costs
Many of the costs related to the product architecture are largely defined and determined by the product architect, and therefore are inherited by the team. Where the development process itself can influence product costs, through detailed selection of components or otherwise, a goal of the development process is to minimize the costs of product architecture. In certain contexts product costs are simply targeted, while in others cost reduction is in fact the strategic goal of the development effort.

Minimize Production Costs
The costs of production of product architectures should be minimized. As with product costs, many of the costs related to production are determined by the product architect, while others are determined by design details resolved during development activities. In certain contexts, costs determined by the specifics of design details are significant. In all contexts, the development process should strive to reduce the costs related to the production of the product architecture.

Prioritization of Product Architecture Development
Value delivery occurs when the process produces product features and functions in a thoughtful sequence, such that risk is reduced through feasibility or ambiguity resolution, or such that value proposition is fulfilled for customer satisfaction and product marketability.

4.3.3. Midstream Value Delivery

4.3.3.1. Midstream Market Goals

Accommodate Market Uncertainty
In certain contexts, customers and markets may remain inaccessible for extended periods. During periods when market ambiguity cannot be resolved, the development team must continue to add value to the project, while managing risk to the greatest possible extent.

Accommodate Changes in Customers' Desires
Process architect strives to accommodate market dynamics resulting in addition or removal of functions and features. Sources of change are fickle behavior from the market, misinterpretation of customer needs, or poorly translated requirements. Value is delivered when the process responds to changes in customer preferences and requirements.

4.3.3.2. Midstream Product Goals

Accommodate Changes in Technology
Process architect strives to accommodate technical dynamics. Changing requirements must be accommodated by the process and its team members. Sources of change are evolving technologies, determinations and discoveries made during feasibility analyses, discovery of missing or poorly designed systems specifications, or discovery of incorrectly specified components and technologies. Value is delivered when the process responds to changes in technologies and requirements.

Exploit Certainty
When requirements related to customer preferences and technologies are relatively certain the team has the opportunity to move efficiently to market. The process should exploit experience, pursue efficient facilitation of reuse, and minimize costly iterations.

Iterate Well
The process architect understands the value of product iterations for reduction of market and technical uncertainty, but also understands the cost and schedule impact that prototype spins incur. The process architect strives to prioritize features and organize teams such that iterations can occur when and where (what teams are iterating) necessary with the least impact on the project. Value is delivered when feasibility, feedback, and improvement objectives are achieved and maximized per iteration.

4.3.4. Downstream Value Delivery

4.3.4.1. Downstream Market Goals

Generate Feedback
A goal of the product development process is to stimulate the market for feedback and information regarding their tastes and preferences relevant to features of the product architecture. During process design, the process architect should incorporate mechanisms that elicit constant and regular feedback from the market. Generally, the earlier and more often market feedback is harvested, the better.

Deliver the Architecture to the Customer
A goal of the product development process is to ensure that the customer/beneficiary (i.e. the end-user, intermediary, or other buyer) is satisfied with the complete product architecture. The complete product architecture is made up of the high-level system architectural details (as conceived and planned by the system architect) as well as the detailed design (provided by the development team).

4.3.4.2. Downstream Enterprise Goals

Deliver the Architecture to the Enterprise
A goal of the product development process is to deliver the architecture to the enterprise (i.e. preparedness) for market access opportunities: scheduled or unscheduled, formal or informal, regular or irregular. In a "build to order" environment, the process strives to realize scheduled market opportunities as accurately as possible. In other contexts there may be multiple opportunities ("sliding windows") with a variety of customers over time, perhaps loosely scheduled or even unscheduled. In all cases, value is delivered when ideas, prototypes, and products are showcased to customers in the best possible functioning order. This goal is appropriate in "build to order" environments, but also in the context that demands a working, demonstrable, road-show version of a product ready to interact with potential customers on a moments notice. While listed here as a downstream goal, market preparedness holds true independent of the current lifecycle phase of the project.

4.3.4.3. Downstream Product Goals

Deliver the Architecture to the Architect
A goal of the product development process is to deliver the architecture to the product system architect. The product development process is responsible for fulfillment of general product requirements and features, meeting or exceeding specifications (i.e. quality, reliability, usability) as conceived and specified by the product system architect. To the extent that the product system architect relies on the development process to resolve design details, the development effort must fulfill its obligations to *complete and deliver* the product system architecture.

Direct Feedback on User-Interfaces
The importance of user-interfaces should not be overlooked by product and process architects alike. Often the fine details of ease-of-use, ergonomics and aesthetics cannot be determined upstream, but instead require active user-operator involvement and scrutiny. In order to create desirable and extraordinary user-interfaces, a goal of the product development process is to establish open dialogue, testing, and feedback of heavily trafficked user interfaces. Value delivery occurs when user-interfaces, both traditional UIs and service interfaces, are critiqued and scrutinized first-hand by users.

Deliver Ultraquality
In mission or safety critical contexts, the nature of product quality and reliability is such that it is a dominant influence driving behaviors of product development teams. A goal of the product development process is to deliver ultraquality architectures where required by system architects.

4.4. Goals and Objectives Case Study

The above analyses for objectives, context and stakeholder investigation were evaluated with regards to ongoing process design efforts in an industry case study.

4.4.1. Analysis Overview

The purpose of conducting the case study interview was twofold: first, to employ the validated context model in an ongoing process design effort; and second, to pursue the goal setting step of the analysis procedure with a process design expert for critique and feedback.

4.4.1.1. Selection of the Case Study

This case study was selected specifically for the unique opportunity to explore the analysis procedure proposed in this thesis with an individual whose current responsibilities involve process architecting activities. The case study provided an opportunity to influence the direction of decision making for initial architecting stages. While this research does not practically afford the opportunity to witness processes set "into motion", the initial thought processes involved in process architecting are deemed insightful.

4.4.1.2. Background

A preliminary interview was conducted by conference call with Aaron Spak, a senior manager at RLW Inc. The interview was completed on April 30th, 2008. The purpose of the interview was to make personal introductions, to learn about Aaron's company, position, and responsibilities as a process designer, and to gauge interest in possible participation in the research. The concepts of the analysis procedure were briefly introduced during the call.

RLW, Inc. of State College, PA, was founded in 2000 to develop embedded processor (hardware/software) electronics assemblies used in the implementation of industrial machinery health monitoring. The company was founded with a sound technical product concept and business/marketing plans, by two veterans of the equipment control and monitoring industry. The founders built a sound technical team from the "bottom-up" with relevant industrial controls and automation experience. Today, the company has 35 employees, of which 17-18 are engineering staff, the remainder management and support staff.

Development Process History

RLW formerly pursued an opportunity with a prime contractor to design the software architecture for an industrial monitoring system for a naval contract which had been secured by the prime's shipbuilding division. The prime showed strong interested in the company's technology, but CMMI certification was a requirement for the opportunity – a capability the company simply could not achieve in short order. The development team had few individuals with DoD contracting experience, particularly in key positions. Consequently, the company had few capabilities and little established culture with regards to formal processes, and certainly had not had the opportunity to pursue CMMI certification.

The prime offered to help with the CMMI certification. To achieve the certification, the prime sent the company the prime's process and procedure documents claiming, "if you use these, you'll get certified!"

> "The prime brought over a dump truck full of process – you could die from starvation, or die from indigestion."

Predictably, the prime's strategy for certification failed; the mountains of processes confused employees, and quite literally soured them to the idea of process. Later, for a variety of reasons including the certification, the opportunity with the prime expired, and today CMMI certification is no longer a priority or concern.

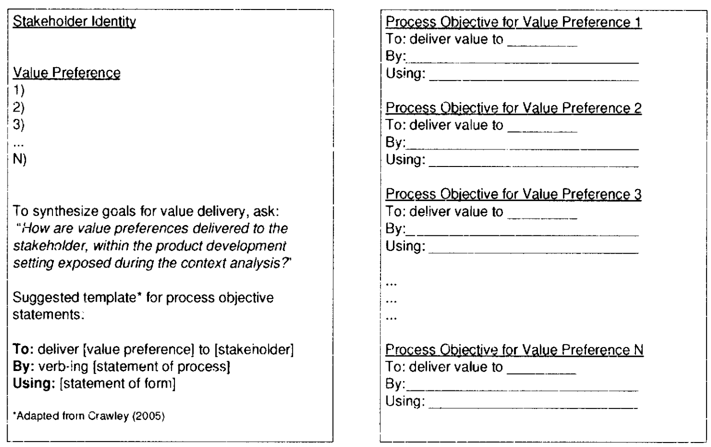
However, the experience with the prime contractor encouraged the company to take a detailed look at their processes. The company has an expanding product line, with a growing number of products and simultaneous development efforts. While standardized processes are no longer a contractual requirement, they are perhaps a simple "organic necessity". Aaron Spak, originally hired as a systems engineer (coincidentally, a former employee of the prime) and now a member of the management staff, has been tasked with establishing and formalizing product development and best practices processes for the young company.

The process designer's objectives are to drive quality, repeatability, and commonality across various development efforts. To achieve these objectives, the process designer began with simple checklists and procedures for generic product development tasks, for example a PCB (printed circuit board) design procedure was created to drive quality in hardware designs. Presently, the process designer is considering the product development process as a whole. For enlightenment, the designer has been researching a variety of process standards and approaches (i.e. DoD, CMMI, and PMI specifications). The process designer agreed to utilize the analysis procedure to further inform decision making for these design tasks.

4.4.1.3. Approach

Context characterization was performed independently by the process designer. A brief follow-up interview was conducted later to collect feedback on the designer's experience with the model. The results of this interview are presented below.

Upon completion of the context model and the summary table, the process architect was asked to continue the analysis procedure by synthesizing goals for value delivery to the pertinent stakeholders of his product development related processes. To aid the interview candidate in the goals synthesis step, a Process Objectives Template was proposed for each stakeholder (Figure 4.2).


Stakeholder Identity

Value Preference
1)
2)
3)
...
N)

To synthesize goals for value delivery, ask:
"How are value preferences delivered to the stakeholder, within the product development setting exposed during the context analysis?"

Suggested template* for process objective statements:

To: deliver [value preference] to [stakeholder]
By: verb-ing [statement of process]
Using: [statement of form]

*Adapted from Crawley (2005)

Process Objective for Value Preference 1
To: deliver value to ________
By: ________
Using: ________

Process Objective for Value Preference 2
To: deliver value to ________
By: ________
Using: ________

Process Objective for Value Preference 3
To: deliver value to ________
By: ________
Using: ________

...
...
...

Process Objective for Value Preference N
To: deliver value to ________
By: ________
Using: ________

Figure 4.1 Process Objectives Template

The template encourages the process designer to record the value preferences for each stakeholder of the product development process. To synthesize goals for value delivery, the process architect must explore value delivery within the product development setting exposed during the context analysis.

The following template[11], adapted from Crawley (2005), is suggested for process objective statements:

To: deliver [value preference] to [stakeholder]
By: verb-ing [statement of process]
Using: [statement of form]

[11] Adapted from Crawley (2005). The product architect is advised to create clear and concise goal statements for stakeholder value delivery through product architecture. Crawley's original template (To [statement of intent], By verbing [statement of the process], Using [statement of the form]) has been adapted for process objectives. The label "objectives" is applied here to distinguish the proposed template structure from its original form. The intention of product development is generically proposed as stakeholder value delivery. The "Using" clause, which is a suggestion process structure and team behaviors, should remain solution neutral, if possible. The details of implementation of the "Using" clause are explored in chapter 6.

The template is proposed in this thesis as a suitable alternative to the formal use of OPM to generate similar goals as suggested by Rebentisch et al. (2005).

An example was provided to aid the designer in use of the template. Finally, a follow-up interview was conducted to obtain feedback: specifically for critique of the objective synthesis method but also generally to judge the utility of the analysis procedure thus far. During the interview, objectives synthesis was explored for a task to create a customer service process. The results of the objectives synthesis step are provided below.

4.4.2. Case Study Results

4.4.2.1. Feedback on the Context Model

Upon completion of the context characterization, the process architect was questioned with regards to the experience with the context model. Specifically, the process designer was asked for critiques of the context model, the relevance of the attributes and its effectiveness for stimulating though.

The experience with the model was generally positive. The designer felt that the model was general, "non-leading", and that it "forces the user to think about what he or she is trying to accomplish." The designer commented that the model was attractive "because it's not cast in the language of an existing paradigm like CMMI."

The process designer felt that the contents of the model were appropriate. The model was not missing any dominant factors, nor did it cause any revelation of factors not yet considered. The model was simply "a good set of memory joggers". While not every factor in the model was relevant to his situation, the process designer agreed that in the variety of possible contexts, the factors in the model were appropriate. When asked if it would change his approach to process design, the designer replied that it would. The designer would now consider a more "structured way of approaching the process design."

4.4.2.2. Synthesis of Objectives

The Process Objective template was applied to a task to write a customer service process. For the process designer's task, two mission statements were provided by general management for the new process:

1) To provide an efficient customer service process to the enterprise by using a distributed support network.
2) Provide customer experience data to the product development process by managing support through common IT infrastructure.

The first mission statement was chosen for decomposition and analysis for goal setting. First, the mission statement was rewritten for clarity:

To satisfy customer expectations & enterprise needs (implied) by providing an efficient customer service process using a distributed support network.

Note that "a distributed support network" is not implementation specific, it has many possible implementations. Similar to the system architect who explores design space (recall that the objective statement was adapted from systems architecting methods), the process architect explores the universe of possibilities for appropriate development processes.

Next, stakeholders are identified from the mission statement. Multiple stakeholders are represented: including the customer, the enterprise, and the development team. Using the Process Objective Template, the enterprise is identified as a stakeholder along with an enumeration of value preferences.

<u>Stakeholder Identity</u>
Enterprise

<u>Value Preference</u>

1) Meet customers' expectations with regards to service and support.
2) Minimize the impact of (1) with regards to cost to the enterprise ('efficiency').
3) Minimize the impact of (1) with regards to man-hours (minimize the interruption to staff).

Finally, for each value preference, write a solution neutral objective statement. Using the Process Objective Template, the process designer decomposes the high-level statement above into more manageable statements. As the objective statements are synthesized, the process designer gains clarity on how the needs of stakeholders might be implemented through process.

To get started, the "customer's expectations" in the original mission statement needed to be enumerated to be more specific about what their value preferences essentially are. The "customers' expectations" (or equivalently, customers' value preferences) are interpreted to be timely service, service quality and pleasant service.

Finally, objective statements are synthesized:

To: meet customer's expectations with regards to service and support,
by: providing timely service,
using: highly accessible people.

To: meet customer's expectations with regards to service and support,
by: providing service quality,
using: highly knowledgeable people.

To: meet customer's expectations with regards to service and support,
by: providing pleasant service,
using: highly polite people.

In this manner, the decomposition progresses leading to more concise objectives for value delivery. The architect has determined that he needs: highly accessible, knowledgeable, and polite people. At this point, hiring and training a mobile support staff is an option, along with using internal personnel who can be contacted via email, as well as several other implementations. The procedure continues in a similar manner for the remaining stakeholders identified above.

4.4.2.3. Feedback on Objective Synthesis

While familiar with modeling languages (UML), the process designer had never experimented with similar objective statements in the past. The process designer agreed that this step of the procedure is time consuming and difficult. The statements are non-trivial to write, and clearly require practice. However, the process designer noted that "you are going make it much easier later" if one would take the time to write the objectives.

When questioned about the utility of generating clear, concise objective statements, the process designer offered unique insight. The designer's tasks are generally team-oriented activities. Until this point, the architect's role had been assumed to be a relatively independent role. The interview candidate praised the usefulness of the objective statements in that they provide a common language "especially if designing as part of a team. The building blocks of a successful team are common language. These statements build that type of vernacular."

When questioned on the utility of a solution neutral, intermediate step in the analysis procedure, the process designer agreed that it was "absolutely critical to have a conscious exercise to define outcomes neutrally."

In the process designer's estimation, the company would not be willing to provide resources for a full-time support staff - a constraint that became obvious after running through the context analysis. This realization emphasizes the usefulness of the context analysis, and how (and when) knowledge of context either supports or constrains the universe of process styles and approaches.

4.5. *Section Conclusions*

4.5.1. Reflections on Goals and Objectives

Retention of Stakeholder Knowledge

As mentioned in the introduction, a driving motivation for this research was a desire for a procedure for determining appropriate product development activities suitable for the unique needs of distinct product development organizations. From its inception, the analysis procedure was envisioned as a sequential activity with distinct boundaries between analysis steps: first, a stakeholder identity and value analysis through context evaluation; next, synthesis of value delivery objectives from the context evaluation completed in the first step; and finally, determination of product development behaviors and activities in fulfillment of value delivery objectives. A lesson of this chapter has been that the architect must not lose sight of stakeholders and their value preferences, even after completing the context analysis of the previous chapter. The exercise of goal setting requires that knowledge of stakeholder identity and value preferences be considered along with the contextual environment of the development effort.

Contention of Process Goals and Objectives

The process architect is aware that contention of goals is highly possible, particularly, short-term enterprise and product objectives versus long-term team goals. For example, it is difficult to balance objectives for employee satisfaction while increasing schedule pressure for an imminent product release. Similarly, it is quite difficult to maintain fluidity of team communication and cohesiveness while restructuring the team to complement a change in product strategy and direction. Objectives for agility and flexibility may ultimately mean rebuilding the team – a bit contentious with the goal of "fun process".

As the process architect synthesizes goals and objectives for the development process, it becomes clear that goals must be weighted and prioritized. Weighting and prioritization of process objectives helps the architect fine tune the process for the enterprise's dynamic short-term needs. Longer-term, objectives must continue to be considered holistically such that the process serves the various stakeholders over the life-cycle of the process, and beyond.

4.5.2. Looking Ahead

As a concluding remark for this section it was determined that process objectives could not be determined from context alone. Instead, knowledge of stakeholder identity and value perspectives must be retained from the onset, and carried forward through the analysis steps. Similarly, it is clear that product development behaviors and activities, the final step of the analysis procedure (and the subject of the next chapter), will not be determined by goals and objectives alone. The process architect will use the results of the context analysis once again, this time in concert with the synthesized goals and objectives, to prescribe the appropriate activities for the product development team. To demonstrate, consider that the realities of context translate into constraints on product development activities designed for value delivery (i.e. lack of resources, scope of effort versus time available, limited experience of team members). Factoring such constraints, while deriving methods and activities, guides the process architect towards behaviors that are more appropriate within these constraints (i.e. methods that compensate for lack of resources or deficiencies in experience). Looking forward to the final step of the analysis procedure, the process architect must again carry forward the results of analyses from earlier steps in the procedure (stakeholders and context, along with process objectives).

5. Behaviors and Activities for Stakeholder Value Delivery

The preceding sections of this thesis have guided the process architect through an investigation of the context of development processes – market conditions as well as product, enterprise, and team attributes – and have led the architect to enumerate goals and objectives for stakeholder value delivery.

Role of the Process Architect

To complete the analysis procedure proposed in the beginning of this thesis, the role of the process architect is extended in this chapter. The process architect is responsible for the prescription of behaviors and activities for the development team and its management, in order to realize goals and objectives for stakeholder value delivery. As stated in the concluding remarks of the previous chapter, in fulfillment of this responsibility the product architect must retain knowledge of stakeholder identity, value preferences, and context.

5.1. Background

5.1.1. Variations in Development Processes

Product development organizations have evolved their processes over time to cope with the needs of stakeholders within various contexts. To cope with an aerospace context, for example, processes are commonly cited as gated-driven, bureaucratic, and highly sequential processes rooted in a "waterfall" approach. Conversely, organizations competing in highly competitive commercial contexts practice frequent iterations, incremental deliveries, and constant reevaluation of customer needs. In these examples, the context of the development effort has shaped the structure of the process and influenced the behavior of the product development team.

Justifying variations in product development group behaviors, Crawley (2005) proposed "Key Differences in PDPs" (Table 5.1).

Number of phases (often a superficial difference)
Phase exit criteria (and degree of formality)
Requirement "enforcement"
Reviews
Prototyping
Testing and Validation
Timing for committing capital
Degree of "customer" selling and interference
Degree of explicit/implicit iteration (waterfall or not)
Timing of supplier involvement

Table 5.1 Crawley's "*Key Differences in PDPs*"

While it is evident that differences in organizational behavior exist, the concern explored in this thesis are methods by which activities and behaviors of product development organizations are determined, and the extent to which they are appropriate for satisfying stakeholder needs efficiently.

5.1.2. Using Lean to Determine Product Development Activities

Lean found its roots on the manufacturing floor, largely as an improvement mechanism for process efficiency and productivity. As Lean has been extended and applied to improve enterprise processes outside of the manufacturing domain, researchers have drawn on many of Lean's underlying principles for guidance. A substantial volume of research (some of which was addressed in the literature review of this thesis) has shown that the underlying principles of Lean (i.e. value stream and waste elimination) can be exploited for application in the product development domain.

In this chapter, linkages are established between various attributes of Lean Product Development and the stakeholder value objectives synthesized in the previous chapter. The establishment of such linkages forms a basis for the prescription of behaviors and activities for the development team. Since Lean may not formally propose to prescribe complete, end-to-end specifics for product development processes, the expectation is that Lean will address many, but not all of the needs of the process architect in fulfillment of process objectives. Still, the goal of Lean Product Development research is to establish linkages between Lean principles and product development in its entirety, to form a complete definition of Lean Product Development. In support of this goal, any new insights gained based on the investigation completed during this research are offered as contributions.

5.2. Fundamentals of Lean Product Development Behavior

The process architect understands that certain behaviors are applicable to most, if not all contexts. The following sections introduce *general guidelines* for product development behaviors which are based upon the underlying principles of Lean. These guidelines, several of which were introduced in the literature review, are the product of the research of many of the authors referenced in this thesis.

5.2.1. Lean Product Development Behaviors

Rebentisch (2007) presented "Some Lean PD Things to Do" (Figure 5.2).

Standardize work at individual and team levels • Standard tools, cycle times, performance expectations • Skills-based personnel progression system • Process owner responsibility for continuous improvement
Establish flow and pull processes • Focus on creating and measuring consistent hand-offs across processes • Create periodic integrating events/mechanisms/roles for project-level coordination • Enable cadence in process execution and integration cycles
Manage staffing for stability, capacity, and learning • Level work load, prevent overburden (static and transient) of resources • Keep pipeline of skilled staff, teachers, and leaders filled and flowing
Use product architecting process to increase PD learning cycles • Increase reuse of product artifacts, standardization, system integration understanding • Enable knowledge capture and process refinement • Use tradespace exploration as an opportunity to develop deeper understanding and knowledge about elements within the architecture (e.g., refine tradeoff curves)
Expand tiers of the value stream participating closely in PD process • Engage customers and suppliers in tradespace exploration and requirements specification

Table 5.2 Rebentisch's *Getting Practical: Some Lean PD Things to Do*

The table highlights many of the realizations of Lean Product Development community over the last several years. Below, several of these concepts are expanded upon in detail.

The following behaviors and activities are generally advocated for and promoted by Lean Product Development research.

5.2.1.1. Synchronization

Lean research has uncovered the enormous advantages of process-wide synchronization (Morgan, 2006). The advantages of synchronization include reduction in waiting, information inventory and wastes due to overproduction (Bauch, 2004).

On the manufacturing floor, Ohno observed wastes due to producing more than necessary, or faster than necessary. Overproduction in manufacturing results from lack of awareness of workload or status of upstream and downstream tasks, or alternatively, lack of a sense of actual customer demand. Overproduction is the root cause of backlog or buildup of materials and work-in-progress, either between process steps or as finished product not yet sold. Extensive wastes can be incurred from such backlog, or inventory (i.e. exposure to rework, as well as diverting time, energy, and material away from other "good" orders).

To combat overproduction wastes, Toyota implemented a system whereby cards were used to help prevent an upstream task from producing before a downstream task was ready to accept what was produced. The cards, known as *kanban*, were passed in the upstream direction from process step to process step, and were used as a prerequisite to begin work on a task. The nature of this process allows a downstream task to control the flow of material or product from the adjacent process step upstream, eliminating the buildup of Inventory between process steps. This activity results in synchronization of process steps, resulting in a cadence in the overall process. Toyota's practice, known now as "pull", was indeed a revolution in auto manufacturing and has seen enormous adoption in other industries and processes as well.

Toyota's remedy was revolutionary in its deviation from the (previously unchallenged) axiom that high utilization rates, per process step, per machine, should be the effort and end goal of process improvements. In contrast, pull forced all process steps in a sequence to work at the pace of the slowest step, eliminating the burden of inventory buildup. Ultimately, an elimination of inventory was accepted as a more cost effective goal than that of achieving high utilization rates. Bauch (2004) noted overproduction as a result of lack of communication and synchronization between upstream and downstream processes. Bauch identified pull processing as manufacturing's remedy for overproduction, and noted that the foundation of pull processing is that it provides the communication and synchronization mechanism required.

Synchronization in Product Development

Synchronization involves the alignment of individuals and tasks, regularity of team communications, revisions and releases, and management of scope of feature development. Synchronization of product development tasks is awkward due to the unpredictable nature of problem solving (Slack, 1999), which of course inhibits the ability to predict the duration of tasks. While product development tasks are certainly less predictable and deterministic than repetitious tasks in a manufacturing process, they are not necessarily without structure or bounds, and therefore are potentially "synchronize-able".

A synchronized process is analogous to a synchronous digital circuit. The first step towards synchronization is to understand the necessary atomic functions of the circuit so that a determination can be made for the speed of the system clock. Working towards synchronization in product development processes, development tasks must be achievable and manageable in scope and time (Bauch, 2004). Hallowell (2005) argued for the avoidance of "big batch production" by using planned iterations and through work-breakdown that includes smaller units that build incremental value. Since synchronization relies on the best predictive nature of the tasks being performed, small tasks and a well coordinated work breakdown are necessary for enabling synchronization in product development. Therefore, a critical step in achieving a process wide synchronization of tasks is to be reasonable about task scope and the capability is of those assigned to tasks.

The next step towards synchronization is to apply the clock. Oppenheim (2004) argued that takt-time is highly applicable to product development and proposed implementing short "takt periods" of equal duration. Oppenheim's suggestion is a choreography in which teams and individuals are permitted to independently pursue development activities in structured periods, but are regularly convened for the purpose of coordination, updating, and assignment of next tasks. As the application of takt and pull processing implements a cadence in manufacturing, establishing regularity of development activity and coordinated communication events creates a cadence in product development.

5.2.1.2. Communication

Lean research has emphasized the importance of communication in enterprise processes (Cusumano, 1998; Chase, 2001). Communication is an enabler for synchronization, employee satisfaction, reduction of ambiguity, and reduced waste in general.

Looking closely at what is actually communicated in pull processing, a downstream task provides notification to an upstream task (via *kanban*) that the downstream task has completed work, and that it is now *ready* to accept product (material, information, or other) from adjacent tasks upstream. Absence of *kanban* is also a medium for communication, notification that the downstream task has not completed work on its current task, and therefore is *not ready*. The essence of pull processing is that downstream tasks *communicate progress* of tasks towards upstream neighbors. In product development then, where development efforts are highly parallel (Slack, 1999), a *process-wide* awareness of task progress must be communicated in order to achieve synchronization, and therefore pull-processing.

Lean Product Development research emphasizes the advantages of cultivating a holistic awareness of schedule, roles, responsibilities, and dependencies, by the product development team. Team awareness of such project-wide data improves efficiencies by achieving knowledge of others' roles and responsibilities, and contributes to job satisfaction through knowledge of how an individuals' efforts contribute to success of the project overall.

Communication is vital in complex team and product scenarios (Allen, 2001) with shared work structures. Communication of project-wide data, and communication of individuals' roles and responsibilities in complex product development scenarios with shared work breakdown reduces ambiguity among participants regarding assignments and responsibilities, and task objectives and scope.

5.2.1.3. Task Management

Lean Product Development emphasizes the advantages of standardized work (Womack & Jones, 1996; Rebentisch, 2007). Task management helps to define standardized work practices and includes the definition of, prioritization and assignment of tasks in a well-managed and well-timed sequence. Task management leads to reductions in wastes due to rework, inventory, and waiting.

Continuity of Work-Flow

Task-management is critical for maintaining a continuity of work-flow. Bauch (2004) and Hallowell (2005) equated task-handoffs and multitasking in product development to transportation wastes on the manufacturing floor. The flow of productivity is interrupted when development tasks are passed among developers, or when task-switching curse in product development. Such discontinuities in workflow incur costs: ramp-up, learning curves, and "knowledge recovery time" (Hallowell, 2005). At times, the individuals involved in a hand-off may be members of separate groups or teams, either within a functional discipline or outside, perhaps even separated by facility, company, or IT system boundaries. In general, the hand-off creates ambiguity about roles and responsibilities in product development processes, and it is evident that discontinuities must be closely managed and reduced (if possible) to ensure success.

Time-Sensitive Inventory

Inventory (material, product or ideas) introduced into a process designed to address customer needs is critically time-sensitive (McManus, 2004). On the manufacturing floor, inventory becomes a waste risk when parts and materials build up at intermediate stages of production, either upstream (for example, raw materials) or downstream (unfinished, or untested product), because the material buildup is vulnerable either to rework from defects discovered later in the process, rework or discard due to changes in customer preferences, or becomes a liability due to lack of demand. Analysis of wastes due to inventory highlights the notion that narrow windows of opportunity exist for satisfying dynamic customer needs. Therefore, processes designed to effectively address customer needs must carefully manage the time-sensitivity of requirements and feature development.

The common known views of the wastes associated with inventory typically deal with *too late*, that is, that the longer materials, product, or ideas sit and idle, the more vulnerable they are to becoming irrelevant. In this sense, Inventory has a potential to become stale or even to rot while customer preferences change - before it is consumed by the process. In product development it is also possible that inventory can be consumed by a downstream process step *too soon*, before it is "finished" by the previous step, or before it is mature enough to be used (Hallowell, 2005). For example, in product development processes one can easily imagine the case that a developer begins work on a task under the assumption that a system specification is mature, when indeed it is not. A developer can spend many days coding up a feature only to learn later on that the specification was incomplete. In this case the "inventory" was a buildup of system requirements, derived in a previous step from an inventory of customer preferences. Our developer made a decision to begin work too soon, before the inventory was mature enough to use.

In summary, Lean Product Development emphasizes the general management of the *time sensitivity* of inventories. Process architects are mindful of the high potential for wastes due to inventories. In product development, information, ideas, requirements, partially developed software code – are all vulnerable as wasted inventory as they accumulate. Tasks must be managed to ensure appropriate timing for inventory to be consumed by the development process. The condition or maturity of inventory (i.e. requirements, features) is of prime concern.

Load-Balancing

Synchronization suggests that process steps should be aligned with regard to task *start times*, such that a cadence is established in the process sequence. Once start times are aligned, it becomes evident that variability exists in process step *duration* (some tasks are completing before others), resulting in idling of process participants while waiting for the completion of other steps. Minimizing wastes due to waiting involves an optimization by which process step boundaries are relocated to manipulate the volume of work completed during execution of the process step. In this manner the duration times of process steps are impacted, with the goal being to bring the duration time in line with other process steps in the sequence. This effort is referred to as *load-balancing*. Load-balancing can be considered as a fine tuning, or an optimization of a process which is best performed after a synchronous cadence has been established.

Recalling the analogy of the synchronous digital circuit presented above, to begin to achieve synchronization we first created a work breakdown of small and manageable tasks, and then applied a takt-time at the rate of the slowest task. Implementing load-balancing in development processes is analogous to reevaluating the circuit for optimization. Optimization is achieved in the circuit when the amount of work done between the synchronous elements is balanced between clock edges. Only after balance is achieved can the maximum frequency of the clock be determined.

To optimize a synchronous development process a mechanism must be provided to permit load-balancing. In product development, load-balancing is achieved through flexibility in the workload assigned to individuals or teams, within the cadence already established through takt-timing of the process. Flexibility can be achieved through task management or by allowing individuals to use discretion as to the scope of tasks to be completed per iteration, or per revision. In effect, load-balancing compensates for the inherent difficulty (indeed, inability) to predict product development task duration, by continually manipulating the volume of tasks assigned per iteration. Bauch (2004) discussed wastes due to waiting in waterfall projects and cited that while developers are waiting for a system release, "stable data" is not moved downstream for further processing (i.e. testing, customer feedback). Load-balancing permits that the actual contents of the system release may vary.

Prioritization of Tasks in Product Development

Careful and deliberate prioritization of tasks is a critical factor affecting the success of a product development project. Considerations to be made when prioritizing tasks are maturity of requirements and features, various risk factors (technical or market) related to requirements and features, and knowledge of highly valued features most closely related to value proposition. The careful prioritization of tasks provides a check against development of non-value added features (Wake, 2003), resulting from lack of discipline of team members or lack of awareness of value proposition.

5.2.1.4. Design Alternatives

This fundamental Lean behavior generally includes design options, trade-space exploration, delayed decisions, and set-based design. Ward and Liker et al. (1996) described Toyota's development approach as a "paradox", in that Toyota achieved remarkably high efficiencies in development approaches with seemingly problematic development approaches. The research found that concurrent engineering teams simultaneously sought out a set of design alternatives, narrowing and refining each of these over the course of the design cycle, and delaying critical design decisions as long as possible (in stark contrast to an early 'freeze' of system requirements).

5.2.1.5. Process Fluidity

On the manufacturing floor, when workspaces are not collocated, when sequential process steps occur across separate facilities, companies, or dispersed geographies, materials must be transported. Lean initiatives attempt to minimize the amount of material in transport, the time material is in transport, and to minimize the costs associated with transporting (Bauch, 2004). Wastes due to transportation occur when the interfaces of process steps are misaligned (i.e. due to company jurisdictions, facility limits, shop floor set-up). Promoting fluidity and flow in development processes involves alignment and manipulation of task boundaries so as to create seamless transitions from one development task to the next. Manipulation of the boundaries between process steps affects the nature by which these interfaces are crossed by materials, information, and individuals involved in the process. As the process is manipulated for better alignment, and as infrastructure is put in place to provide for more efficient exchange, the cost impact of the boundary itself becomes negligible.

The Lean enterprise is an integrated enterprise composed of processes and personnel which span functional, departmental, and even corporate boundaries in a coordinated and communicative effort to achieve and end. Process fluidity encompasses all activities traversing the Lean enterprise.

Concurrent Engineering and Integrated Process Teams

Concurrent engineering and integrated process teams break down the communication barriers between functional disciplines and promote teamwork and pull (Bauch, 2004) in product development process. Concurrency in the engineering processes implies a coordinated simultaneous activity of upstream and downstream tasks and processes. Concurrent activities were documented well by Takeuchi and Nonaka (1986) and Umemoto et al. (2004). Coordinated, concurrent development activities imply effective communication between upstream and downstream processes. Integrated process teams imply cross-functional participation and representation on development teams. Browning

(1996) presented principles regarding effective formation and utilization of integrated process teams, and their Lean applicability.

Integrated Enterprise

Toyota's embodiment of an integrated philosophy is evident in their approach to supplier commitment, investment, and incubation (Womack & Jones, 1996). Poppendieck (2001) noted Toyota's nurturing, long-term approach to supplier relationships and cited them as a "paradigm shift". The notion of the integrated Lean Enterprise challenges traditional supplier relationships and traditional corporate boundaries in general. Indeed, other operational sciences include thorough analysis of these boundaries as well. In many of today's sophisticated supply-chains, manipulation of enterprise boundaries goes hand-in-hand with the "tearing down" of traditional company barriers.

5.2.1.6. Employee Value and Empowerment

Lean stresses the value of employees, and challenges and empowers employees to continuously perfect their work approaches and methods (Womack & Jones, 1996). Team members must be kept abreast of market conditions, and system and project progress so that a holistic understanding of roles, responsibilities, and contributions is achieved. Lean promotes recognition of employee efforts, and alignment of rewards and compensation with achievements made in job skills and continuous improvement activities.

5.2.1.7. Customer Focus

Lean Product Development stresses the advantages of involving the customer iteratively in the product development process (Poppendieck, 2001). Intimate customer involvement improves the upfront development of requirements and specifications, improves the interpretation of each downstream, and generally improves information quality process-wide. Customer focus in product development processes reduces wastes resulting from ambiguity of customer value preferences and improves the likelihood of value delivery.

5.2.1.8. Information Strategy/Knowledge Management

A solid information strategy and simple access to quality information is considered a prerequisite foundation for any functioning Lean Product Development organization. Waste occurs when there are miscommunications, no communication, or data incompatibilities among individuals on the development team (McManus, 2004; Bauch, 2004).

Product development processes produce information (Slack, 1999). Defects in product development are defects in the information itself (McManus & Millard, 2002). Product development defects are incorrect specifications, or other oversight, that leads to less than acceptable product or production performance. Lean emphasizes rigor and accuracy of information flows in product development organizations as every process step in a development process is dependent on the quality and timeliness of information received from a process step upstream. The accuracy and integrity of information is particularly vulnerable when customer requirements and specifications are translated and rewritten at team, department, or corporate boundaries.

5.2.1.9. Monuments

Lean discourages the use of monuments in production and enterprise processes. Monuments are resources (i.e. infrastructure, tools, team structure, even processes and methods) which are inappropriate due to capability or capacity – typically employed on current projects simply because they are familiar from or available due to use on past projects. Inappropriate resources lead to excessive processing and verification (McManus & Millard, 2002), and are a symptom of inflexibility of process (the "wrong tool" for the job). Minimizing wastes in manufacturing involves a process by which the most appropriate tool for a repetitious task is identified.

Monuments in Product Development

Crawley (2005) noted that product-alignment of teams can bias future product architectures, warning that "organizations can become *'fossilized'* by [their] first product". A common example occurs when teams select components based on familiarity as opposed to fit – either due to ignorance or for the purposes of risk reduction. According to Crawley, product-teams "impose organization on product" and then "need 'tiger teams' to mend the misalignment of product and architecture." Team structure and organization, and indeed *methods* used must be kept current per the needs of the project at hand.

In product development, where new challenges are confronted daily, the concept of "right tool for the job" suggests the need for flexibility in the chosen process or methods used. Flexibility requires *awareness* that a chosen method or approach may not be applicable in a certain state for a certain task, and the *freedom* to address problem solving with alternate methods.

5.2.1.10. Continuous Improvement

Lean initiatives are well-known for emphasis on continuous improvement activities, which includes relentless scrutiny of tools, processes, and methods for waste identification and elimination. Improvement efforts are participant driven by developers and designers who are empowered to control their own tasks. Continuous improvement is encouraged, recognized and rewarded by management. Continuous improvement includes value placed on training and education. Morgan (2006) showed that Toyota's continuous improvement philosophy includes incentives and performance evaluation criteria to promote real-time and reflective learning events for process participants.

5.3. *Linking Lean Fundamentals to Process Objectives*

A brief analysis was completed to judge the alignment of Lean Product Development fundamentals with the objectives for stakeholder value delivery synthesized in the previous chapter. The analysis paired the 39 categories of product development process objectives with eight categories of Lean fundamentals. The purpose of this analysis was to challenge the utility of Lean Product Development for fulfillment of the process objectives. As mentioned above, the expectation was that Lean would address many, but perhaps not all of the needs of the process architect in fulfillment of process objectives. The following table presents an excerpt from the Objective/Behavior comparison table which demonstrates this analysis. The Objective/Behavior table is included in its entirety in Appendix C.

	Objectives and Goals for Stakeholder Value Delivery	Lean Behavior	Identified as Waste Objective	Identified as Value Objective	Addresses Objective with Beh.	Behavior Explicit	Flexible without Behavior	Rigid without Behavior
Upstream Value Delivery	Maximize Information Quality	Information Strategy/Knowledge Management	1	1	1	Y	N/A	N/A
	Upstream Team Goals							
	Participant Education – Market Awareness	Employee Value Information Strategy/Knowledge Management Continous Improvement.training	-1	0	0	N	Y	N
	Participant Education – Technical Proficiency	Employee Value Information Strategy/Knowledge Management Continous Improvement.training	1	1	1	Y	N/A	N/A
	Individual/Product Structure	Monuments flexibility of organization Process Fluidity IPTs Continuous Improvement training	1	1	1	N	Y	N/A
	Team/Product Structure	Monuments flexibility of organization Process Fluidity.IPTs	1	1	0	N	0	0
	Specialized Expertise	Process Fluidity.CE & IPTs Continuous improvement.training Continuous Improvement.incentives	0	0	0	N	0	0
	Minimize Perceived Complexity	Process Fluidity.all Task Management.all Continuous Improvement.training	0	1	1	N	Y	N
	Upstream Product Goals							
	Reduce Technical Uncertainty	Task Management.prioritization Design Alternatives Continuous Improvement.training	1	1	1	Y	N/A	N/A
	Minimize Product Costs	Task Management.prioritization Continuous Improvement training	1	1	0	N	Y	N
	Minimize Production Costs	Communication Process Fluidity.all Continuous Improvement.training	1	1	1	Y	N/A	N/A
	Prioritization of Product Architecture Development	Synchronization Communication Task Management prioritization	1	1	1	Y	N/A	N/A
	Midstream Market Goals							
		Synchronization						

Table 5.3 Objective/Behavior Comparison (excerpt)

5.3.1. Discussion

In this analysis, the process objectives synthesized in the previous chapter were considered independently. For each objective, Lean fundamentals were selected which, if implemented in design and development processes, would guide the development team towards fulfillment of the objective in question. Afterwards, the objective/behavior pairs were analyzed to make further determinations: if the objective in question was identified by waste-based or value-based research approaches[12]; the extent to which the Lean

[12] Common research approaches were introduced in Section 2.4.1.

fundamentals chosen fully addressed the objective; whether or not the behaviors chosen were designed to address the objective explicitly; and finally, where objectives were not fully addressed using the Lean fundamentals, whether or not Lean Product Development is flexible enough to accommodate other methods to satisfy the objective. A simple {-1 (not well), 0 (neutral), +1 (well)} scoring mechanism was chosen to rate how well the research had identified the objectives, and for judging how well the Lean fundamentals satisfied the objectives.

5.3.2. Results

Lean Product Development guides the process architect well in determination of behaviors and activities for stakeholder value delivery. Many of the objectives and goals synthesized in the previous chapter were directly addressed by the behaviors of Lean Product Development, others were accommodated indirectly. Only minor deficiencies were found in the application of Lean Product Development fundamentals.

As evidenced in the table, very typically objectives are satisfied through the use of combinations of Lean fundamentals working together in unison for value delivery, rather than a single Lean behavior working alone.

5.3.2.1. Synergies

Goals and Objectives Directly Addressed by Lean Fundamentals

The majority of process objectives and goals were satisfied quite directly through the behaviors of Lean Product Development. In these instances a strong correlation was found between the stated objective and the chosen Lean fundamentals (both through historical research as well as Lean Product Development research). Lifecycle Objectives satisfied were all goals related to Enterprise Connectedness, several Process Capabilities, including High Process Efficiency, Risk Management & Mitigation, Enterprise Learning and Knowledge Management, and Process Flexibility. Also satisfied were Acceptable Process – Product and Participant Perspectives. Upstream phase specific objectives satisfied directly were all market objectives, Participant Technical Education, Reduction of Technical Uncertainty, Minimize Production Costs, and Prioritization. Also directly addressed were the Midstream goal of Accommodating Change and Downstream objectives of Generating Feedback and Customer Satisfaction, as well as Architect's Satisfaction and Ultraquality.

Lean initiatives are widely accepted for their strengths in enterprise connectedness, process efficiencies and capabilities, and strong focus on customer value. As such, Lean's direct application and fulfillment of many of the process objectives identified above was not surprising and indeed expected prior to performing the analysis.

Goals and Objectives Indirectly Addressed by Lean Fundamentals

The majority of the remaining objectives for stakeholder value delivery were addressed indirectly utilizing the behaviors of Lean Product Development. In these instances, the benefits of Lean became more evident as it was discovered that Lean behaviors were not intentionally designed to accommodate these objectives. Lifecycle objectives satisfied were Value Management and Process Transparency, as well as Acceptable Process – Enterprise Perspective. Upstream objectives satisfied were Individual/Product Structure, and Minimize Perceived Complexity. Midstream objectives satisfied were Accommodate Market Uncertainty, Exploit Certainty, and Iterate Well. Downstream objectives satisfied were Deliver the Architecture to the Enterprise, and Direct Feedback on User-Interfaces.

5.3.2.2. Challenges

Minimal deficiencies were discovered using the analysis as suggested above. The following notations briefly address gaps in the objective/behavior comparison study, perhaps suggesting areas for further research.

High-Pressure/Urgency

Lean initiatives have shown enormous strengths in terms of process efficiencies and improvements. Such achievements in process efficiencies hold perhaps the greatest promise of improved times-to-market in product development processes. Still, high-pressure and urgent environments require extraordinary qualities from project leadership and process participants, and typically require additional resources. High-pressure environments require energetic and motivational leadership, individuals with intrinsic motivational drive, and specially designed incentive programs to reward extra effort put forth by employees. Lean leadership has generally centered on Lean implementation, change, and transformation (Bozdogan et al., 2000). Lean product development research could benefit from studies that comprehensively identify the canonical and ideal traits of *Lean Individuals*, both process participants and process management.

In regards to a formal analysis of Lean Individuals, it became evident in this analysis that fulfillment of many of the process objectives required extensive employee training and education. The diversity of subject matter process participants should be exposed to is of particular interest relative to the success of the project. This analysis suggests a subset of this exposure: market as well as specialized technical knowledge, system knowledge, schedule knowledge, and project knowledge including awareness of the individual's and others' task assignments and progress.

Team Organization & Specialized Functional Expertise

Lean advocates for cross-functional and inter-disciplinary development teams, and extends these principles to challenge functional as well as traditional departmental and corporate boundaries where "the horizontal axis dominates the organizational structure" (Bozdogan et al., 2000). Still, many product architectural contexts require extraordinary functional specialization (Allen, 2001). While Lean does advocate for functional training and rewards based on functional performance, it seems that there may be some residual tension where the product calls for very functional behavior.

Lean's aversion to specialists originates from Ohno's analysis of transportation wastes on Toyota's production floor. Challenging the need for high-capacity, specialized machinery, the set up of machines was changed to improve flow (from a grouping of like machines requiring specialists, to a flow of distinct machines along a process line). The new setups changed the nature of tasks performed per individual, requiring that individuals gained proficiency on several distinct machines. Ohno recorded these observations and concluded that all specialized expertise (human or machine) drives waste. Process steps are delineated by the specialized expertise (human or machine) of those involved in the process, therefore leading to backlogs or underutilization of assets.

It seems that Lean will always contend with functional organizations and specialized expertise to an extent. Lean Product Development research can benefit from a study of team organizational structures which appropriately complement product architectural tasks with highly specialized functional needs.

Acceptable Process – Market Perspective

The market perspective of the Acceptable Process objectives is designed to capture certain contexts in which many aspects of the process may be non-negotiable. For example, in certain industries the contents, format, and timing of deliverables (i.e. required reviews, specifications, test plans, and acceptance procedures) is contractually agreed upon at the start of the project. The objective suggests that the development process should accommodate the customer's preferences in this regard. To satisfy the objective of accommodating industry norms, the process architect *can innovate*, but ultimately all customers' desires must be met (including those aligned with industry norms). This research suggests that process innovation must occur "in between" deliverables.

Lean practitioners generally suggest that highly bureaucratic contexts are wrought with wastes and should be redesigned from ground up, would remain at odds with this objective. While Lean is flexible to literally any structure, a VSM analysis would undoubtedly identify non value-add activities and move quickly towards waste elimination. The tension of course is that Lean's strengths are in creating and maximizing process efficiencies, which may often be at odds with industry norms.

6. Summary

This research presented a review of Lean Product Development, introduced and explored the role of the process architect, and proposed an analysis procedure to aid the process architect in the determination of appropriate behaviors in product development organizations. A model was formulated for characterizing the context of a product development project as a critical initial step in the analysis procedure. Analyses were completed based on the context model. Product development objectives and goals were synthesized for value delivery. Finally, the utility of Lean Product Development was examined in relation to goals and objectives for stakeholder value delivery.

6.1. Product Development Process Architect

The Product Development Process Architect is a critical, full-time organizational role in the high-technology enterprise. The role of the process architect is to advise, design and advocate for product development environments that evolve to deliver value to all process stakeholders, consequently satisfying the dynamic, strategic needs of the enterprise.

The product development process architect must be a holistic, systems thinker with broad and deep understanding of a variety of influential factors. First, the process architect must have market knowledge and business training. The process architect must have a comprehensive knowledge of the enterprise: its organization, supporting processes, and resources. The process architect must have comprehensive knowledge of the development staff: their personalities, work preferences, and their experience. Finally, the process architect has knowledge and experience with products, training in systems and architectures, and must maintain a working relationship with the *product architects* themselves. The process architect must communicate with the enterprise, its leadership, and development staff. The architect must have credibility with, and the respect of the development team, and must have the ability to persuade and influence behaviors, and indeed effect cultural change where necessary. The process architect must therefore be a leader as well as a listener, and strive for common language within the development organization.

The role of the process architect should be a relatively independent role - the architect should not be a participant in the process, nor a strict representative of the customer or the enterprise – such that the role should not be biased too heavily by an individual stakeholder or by the current organization of the enterprise, business unit, or functional department. Still, the process architect must be a permanent member of the enterprise, due to the required level of intimacy with the organization and the staff.

6.1.1. Proposed Steps for Architecting Processes

The following procedure is proposed as an analysis methodology for determination of appropriate process architectures. The process architect utilizes such a procedure to inform decision making with regards to process definition and implementation.

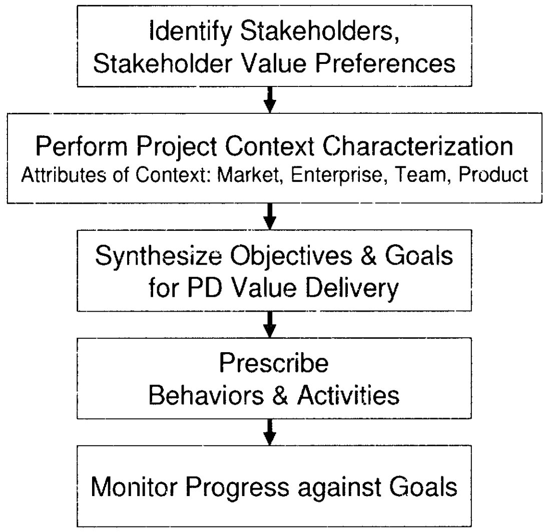


Figure 6.1 Analysis Procedure for Value Delivery

6.2. Product Development Context

Four perspectives were chosen as relevant perspectives for exploration of the environment surrounding the product development effort. Forces contributing to define the product development environment were categorized and used explicitly in the form of a model for the purpose of characterizing the context of a development effort.

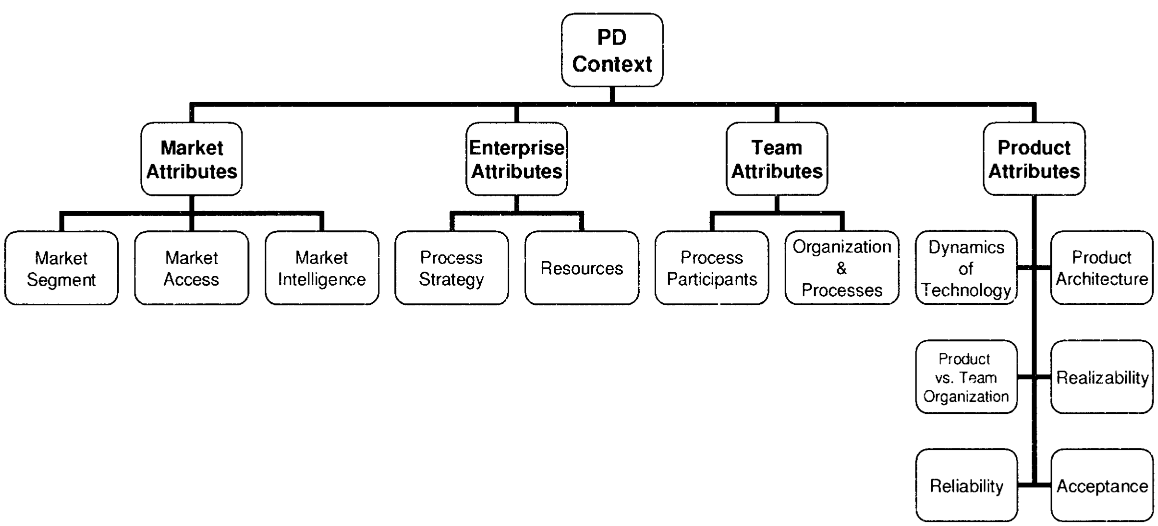


Figure 6.2 Model for Context Characterization

The context model was validated with case study interviews performed at product development organizations from separate market, enterprise, team and product contexts.

6.2.1. Project Context

Using the context evaluation, the process architect evaluates the context of a *specific* product development effort, the *project context* -- market, enterprise, team, and product factors – not all of which are applicable to the entire product development organization. In lieu of publishing a single, common, enterprise process for all projects, the process architect will likely advise any of a variety of approaches, depending on the unique circumstances of a given development project.

6.3. Using Project Context to Determine Goals

To lead the process architect along a focused path towards value delivery, the process architect explores stakeholder value delivery within the various contextual scenarios captured using the context model. The synthesis of product development goals and objectives using context analysis will help the process architect to concisely target value delivery and maximization to various stakeholders.

The process architect uses a Process Objective Template to formulate concise objectives for stakeholder value delivery.

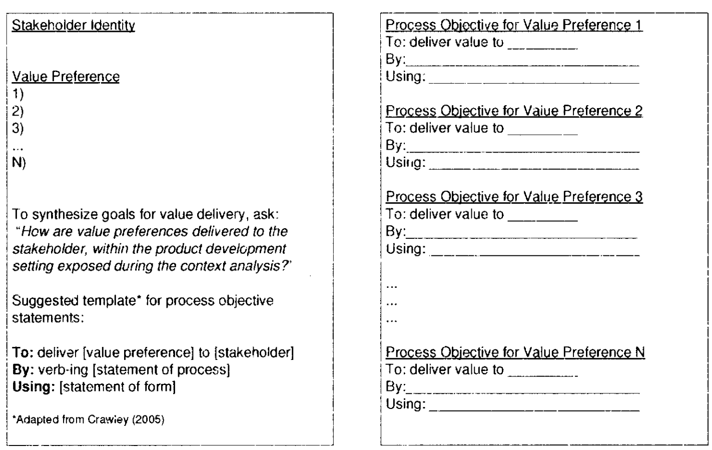


Figure 6.3 Process Objectives Template

The process architect will use synthesized goals and objectives in concert with the results of the context analysis to prescribe the appropriate activities for the product development team.

6.4. Behaviors and Activities for Stakeholder Value Delivery

To complete the analysis procedure, the process architect is responsible for the prescription of behaviors and activities for the development team and its management, in order to realize goals and objectives for stakeholder value delivery. In fulfillment of this responsibility the product architect must retain knowledge of stakeholder identity, value preferences, and context.

Lean Product Development guides the process architect well in determination of behaviors and activities for stakeholder value delivery. Lean Product Development addresses the process goals and objectives for stakeholder value delivery synthesized using the Process Objective Template. Typically, objectives are satisfied through the use of combinations of Lean fundamentals working together in unison for value delivery, rather than a single Lean behavior working alone.

References

Allen, T.J., (2001), "Organizing for Product Development," MIT Sloan Working Paper No. 4229-01. Available at SSRN: http://ssrn.com/abstract=297304.

Anderson, A.R. and M.H. Atkins, (2001), "Business Strategies for Entrepreneurial Small Firms," *Strategic Change,* September/October, pp. 311-324.

Barney, J.A., (1996), *Gaining and Sustaining Competitive Advantage,* Addison-Wesley Longman, Pearson Education, NJ.

Bauch, C., (2004), "Lean Product Development: Making Waste Transparent," Diploma Thesis, Massachusetts Institute of Technology and Technical University of Munich.

Benabou, R. and J. Tirole, (2003), "Intrinsic and Extrinsic Motivation," *Review of Economic Studies,* Blackwell Publishing, vol. 70(3), pp. 489-520.

Boehm, B. and R. Turner, (2004), *Balancing Agility and Discipline: A Guide for the Perplexed,* Pearson Education.

Bowen, D.E. and E. Lawler, III, (1995), "Empowering Service Employees," *Sloan Management Review,* Summer, 36, 4, p. 73.

Bozdogan, K., R. Milauskas, J. Mize, D. Nightingale, A. Taneja and D. Tonaszuck, (2000), Transition to Lean (TTL) Roadmap, Lean Aerospace Initiative, MIT, http://lean.mit.edu.

Brooks, Jr., F.P., (1995), *The Mythical Man-Month –Anniversary Edition,* Addison Wesley Longman, Inc.

Browning, T.R., (1996), "Systematic IPT Integration in Lean Development Programs," Diploma Thesis, MIT, Cambridge, MA.

Browning, T.R., (1999), "Complex System Product Development: Adding Value by Creating Information and Reducing Risk," *LAI Report,* WP99-03.

Browning, T.R., (2000), "Value-Based Product Development: Refocusing Lean," *IEEE,* 0-7803-6442-2.

Chase, J.P., (2000), "Measuring Value in Product Development," *LAI Working Paper Series,* WP00-05.

Chase, J.P., (2001), "Value Creation in the Product Development Process," Diploma Thesis, MIT, Cambridge, MA.

Crawley, E., (2005), *System Architecture: ESD.34 Lecture Notes (Rev 2.0),* MIT, Fall.

Cusumano, M.A. and K. Nobeoka, (1998), *Thinking Beyond Lean: How Multi-Project Management is Transforming Product Development at Toyota and Other Companies,* Free Press.

DoD Architecture Framework (DoDAF), (2004), Volumes 1-3. Available at: http://www.defenselink.mil/cio-nii/docs.

Hallowell, D.L., (2005), "Software Development Convergence: Six Sigma-Lean-Agile," *iSixSigma.com.* Available at: http://software.isixsigma.com/library/content/c050302b.asp

Hamel, G., (2002), *Leading the Revolution*, Harvard Business School Press.

Herzberg, F., (1968), "One more time: how do you motivate employees?," *Harvard Business Review,* Volume 46, Issue 1, pp. 53-62.

Kato, J., (2005), "Development of a Process for Continuous Creation of Lean Value in Product Development Organizations," Diploma Thesis, Massachusetts Institute of Technology.

Kiraka, R. and K. Manning, (2005), "Managing Organizations Through a Process-Based Perspective: Its Challenges and Benefits," *Knowledge and Process Management*, Volume 12, pp. 288-298.

Kraemer, K. and J. Dedrick, (2002), "Dell Computer: Organization of a Global Production Network," Center for Research on Information Technology and Organizations, UC Irvine.

Lehto, J. and P. Marttiin, (2005), "Experiences in System Architecture Evaluation: A Communication View for Architectural Design," *Proceedings of the 38th Annual International Conference on System Sciences,* IEEE, 0-7695-2268-8.

Liker, J., (2004), *The Toyota Way,* McGraw-Hill.

MacCormack, A., (2001), "Product-Development Practices That Work: How Internet Companies Build Software," *MIT Sloan Management Review,* Winter, pp. 75-84.

McConnell, S., (1997), "Achieving Leaner Software," *IEEE Software,* November/December.

McManus, H., (2004), "Product Development Value Stream Mapping (PDVSM) Manual," Lean Aerospace Initiative, MIT, Cambridge, MA.

McManus, H.L. and R.L. Millard, (2002), "Value Stream Mapping and Analysis for Product Development," *Proceedings of the International Council of the Aeronautical Sciences,* 23rd ICAS Congress, 8-13 September.

Miller, P. and R. Palmer, (2001), "Turning Innovation into Profit," *Strategic Change,* March/April, pp. 87-93.

Morgan, J., (2006), "Applying Lean Principles to Product Development," *SAE International.*

Oppenhiem, B.W., (2004), "Lean Product Development Flow," *System Engineering,* Vol 7-4.

Poppendieck, M. and T. Poppendieck, (2003), *Lean Software Development - An Agile Toolkit,* Addison Wesley.

Rebentisch, E.S., E.F. Crawley, G. Loureiro, J.Q. Dickmann and S.N. Catanzaro, (2005), "Using Stakeholder Value Analysis to Build Exploration Sustainability", AIAA-2005-2553, *1st Space Exploration Conference: Continuing the Voyage of Discovery,* Orlando, FL.

Rebentisch, E.S., (2007), "Lean Product Development," Presented at the LAI Annual Conference, Cambridge, MD, April, 17.

Rechtin, E.S. and M. Maier, (2001), *The Art of Systems Architecting 2nd Edition,* CRC Press.

Rothwell, R. and W. Zegveld, (1985), *Reindustrialization and Technology*, Longman, Harlow.

Rothwell, R., (1994), "Towards the Fifth-generation Innovation Process," *International Marketing Review,* Volume 11, Issue 1, pp. 7–31.

Routio, P. (1999), "Arteology, the science of products and professions," *Technical Report,* University of Arts and Design Helsinki. Available at: http://www2.uiah.fi/projects/metodi

Slack, R.A. (1999), "The Application of Lean Principles to the Military Aerospace Product Development Process," Diploma Thesis, MIT, Cambridge, MA.

Takeuchi, H. and I. Nonaka, (1986), "The new new product development game," *Harvard Business Review,* pp. 137-146, January-February.

Ulrich, K.T. and S. Eppinger, (1995), *Product Design and Development,* NY: McGraw-Hill.

Umemoto, K., A. Endo and M. Machado, (2004), "From sashimi to zen-in: the evolution of concurrent engineering at Fuji Xerox", *Journal of Knowledge Management,* 8, 4; p. 89.

Wake, W., (2003), "Lean Manufacturing and Software," *http://xp123.com.* Available at: http://xp123.com/xplor/xp0312/index.shtml

Walton, M., (1999), Strategies for Lean Product Development, *The Lean Aerospace Initiative Working Paper Series,* WP99-01-91.

Ward, A., J.K. Liker, J. Cristiano and D.K. Sobek, II, (1995), "The Second Toyota Paradox: How Delaying Decisions Can Make Better Cars Faster", *Sloan Management Review,* Spring, Vol. 36, No. 3, pp. 43–61.

Wirth, N., (1995), "A Plea for Lean Software," *IEEE Computer*, 0018-9162.

Whitaker, R., (2005), "Value Stream Mapping and Earned Value Management: Two Perspectives on Value in Product Development," Diploma Thesis, MIT, Cambridge, MA.

Womack, J.P. and D.T. Jones, (1996), *Lean Thinking: Banish Waste and Create Wealth in Your Corporation,* New York: Simon & Schuster.

Appendix

A. Model for Product Development Context Characterization

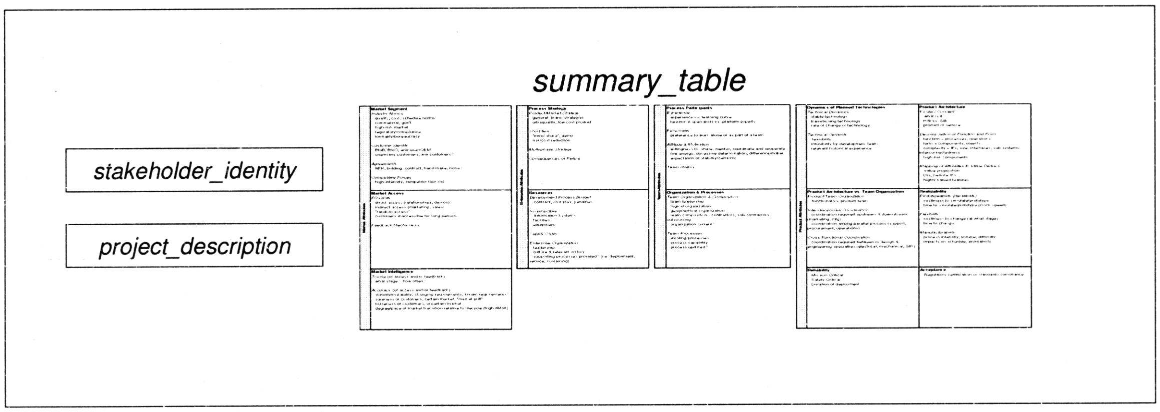


Figure A.1 Context Model for a Development Project

Below the four sections of the summary table are presented in detail. This table represents the updated table after the completion of the case study interviews.

A.1. Market Attributes

Market Attributes

Market Segment

Industry Norms
- quality, cost, schedule norms
- commercial, gov't
- high-risk market
- regulatory/compliance
- formality/bureaucracy

Customer Identity
- BtoB, BtoC, end-user/OEM
- one/many customers, any customers?

Agreements
- RFP, bidding, contract, handshake, none?
- estimation of scope, cost, & schedule

Competitive Forces
- high-intensity, competitor lock-out

Market Access

Proximity
- direct access (relationships, demos)
- indirect access (marketing, sales)
- "random access"
- customers inaccessible for long periods

Feedback Mechanisms

Market Intelligence

Timing (of access and/or feedback)
- what stage? how often?

Accuracy (of access and/or feedback)
- stability/instability, changing requirements, known requirements?
- sureness of customers, certain market, "market pull"
- fickleness of customers, uncertain market
- degree/pace of market transition relative to lifecycle

Figure A.2 Market Attributes of Product Development Context

A.2. Enterprise Attributes

Enterprise Attributes

Process Strategy

Product/Market Strategy
- general, brand strategies
- ultraquality, low cost product

Short term
- "mind share", demo
- risk/cost reduction

Multi-phase Strategy

Relative Strength of Project

Consequences of Failure

Resources

Development Process Budget
- contract, cost plus, penalties

Infrastructure
- Information Systems
- facilities
- equipment

Supply-Chain

Enterprise Organization
- leadership
- culture & relevant history
- vital supportive processes (i.e. deployment, service, sustaining)

Enterprise Dynamics
- internal inconsistencies
- resistance to change

Figure A.3 Enterprise Attributes of Product Development Context

A.3. Team Attributes

Team Attributes	
	Process Participants *Team History* *Experience* experience vs. learning curve functional specialists vs. platform experts difference-makers *Personality* preference to work alone or as part of a team *Attitude & Motivation* willingness to: share, mentor, coordinate and cooperate low-energy, obsessive determination expectation of stability/certainty
	Organization & Processes *Team Organization & Composition* strength of team leadership logical organization geographical organization team composition - contractors, sub-contractors, outsourcing organization current? turnover *Team Processes* management techniques existing processes process capability process formality required deliverables process updated?

Figure A.4 Team Attributes of Product Development Context

A.4. Product Attributes

Product Attributes	**Dynamics of Planned Technologies** *Technical Dynamics* stable technology transitioning technology rate of change of technology *Technical Certainty* feasibility infusibility by development team relevant historical experience	**Product Architecture** *Product Concept* what is it? HW vs. SW product or service *Decomposition of Function and Form* function = processes, operations form = components, objects complexity = #'s, size, interfaces, sub-systems, interconnectedness required architecture high-risk components *Mapping of Attributes to Value Delivery* value proposition UIs, Service IFs highly valued features
	Product Architecture vs. Team Organization *Product/Team Organization* functional vs. product team *Inter-disciplinary Coordination* coordination required upstream & downstream (marketing, mfg) coordination among parallel process (support, procurement, operations) *Cross-Functional Coordination* coordination required between in design & engineering specialties (electrical, mechanical, SW)	**Realizability** *Prototypeability (Iteratibility)* costliness to simulate/prototype time to simulate/prototype (clock speed) *Flexibility* costliness to change (at what stage) time to change *Manufacturability* process intensity, volume, difficulty impacts on schedule, profitability
	Reliability Mission Critical Safety Critical Duration of deployment	**Acceptance** Regulatory certification or standards compliance

Figure A.5 Product Attributes of Product Development Context

B. Case Study Interviews & Results

Expressions recorded in quotations are intended to reflect the spoken words of interview candidates.

B.1. Case Study A: Plexis Gigabit Ethernet Transport Development

The Plexis Gigabit Ethernet Transport development program began in 2000 at a pure startup developing systems (OEM) for deployment in metro-optical, intermediate reach networks such as those deployed in digital cable systems. The company, whose original charter was to develop optical components, was founded by a team of research scientists from a government agency in 1997. The success of the Plexis development program led to the acquisition of the company by a competitor in 2005. The interview was held in-person, in a discussion format with the former Director of Engineering over two sessions: April 3 and April 8, 2008.

B.1.1. Plexis Development Stakeholders

B.1.1.1. Agents

The company was privately financed by traditional venture capital firms, over two rounds of funding. The initial round was financed by a single firm, for approximately $8M in 1997. The second round of financing took place in 2003 by a larger group of 6 firms, for approximately $22.4M. The second round investors formed a board of directors, which also included the CEO and the co-founders of the company. The management team included a CEO, President/co-founder, CTO/co-founder, VPs of Operations, Engineering and Optics, and Directors of Hardware and Software development. The development team (22 developers at its peak) was comprised of hardware, software, optical, and mechanical engineers.

Supportive functions included local marketing and operations personnel, contract-IT, and human resources. Other indirect agents included various suppliers of components and materials, and outsourced manufacturing. The company was not co-dependent on any partners.

B.1.1.2. Beneficiaries

Transport Ethernet is a service enabling product, whose beneficiaries include the MSO purchaser, the MSO evaluator, and the MSO operator (actual "end-user") of the product. The MSO constructs a communications network from gear such as Plexis, and in turn offers subscribers services resulting from the interoperation of all such devices. In this manner, the network subscriber indirectly reaps the rewards of the product quality with high service level enjoyed at home.

B.1.2. Plexis Development Project Context

The project of interest for the interview is the Plexis GbE Transport platform, a chassis based linecard system with several pluggable modules. Plexis modules included data transport, amplifier, combiner/splitter and management capability, all with embedded software applications, firmware code, and remote management capability (human and machine UIs). Development of the product line began in 2000 and continued until the company was acquired in 2005.

B.1.2.1. Market Attributes

Market Segment

The cable services provider market is a market that today offers commercial and residential subscribers a converged television, telephone, and internet service through regional hybrid fiber/cable networks. The converged services lend network operators to the name "multiple system operators" (MSO).

The cable market is a tiered market – upper tiers dominated by few key service providers with local monopolies in major markets, while in lower tiers MSOs are less geographically dispersed, in smaller markets with fewer subscribers and smaller budgets. Plexis was marketed predominantly to tier 1 customers in larger markets. Amid tier 1, a variety of styles were possible among the key players, in terms of customers ranging from liberal to conservative in their tastes for new technology, and their tastes for working with start-ups. MSOs processes for technology evaluation and selection were highly varied, ranging from centralized (technical HQ) to decentralized (regional authority).

Traditionally a cost-oriented industry, the quality level of cable networks is widely understood as less than that of traditional telecommunications, for a variety of reasons including price of services, regulation, and technology limitations, but also for presumed tolerance of video quality from subscribers. With the move to converged services though, the market experienced transitions to higher quality, and is seeing competition and even consolidation with telecommunications networks, companies, and personnel.

There was officially an RFP process, but in reality it was highly political and very much a "game more than a legitimate process". Many times, relationships played a highly influential role. Customers played competitors off of each other for price, typically dragging the company along for the ride ultimately to use them as cost pressure against larger companies that they would prefer to do business with. As time went on, other influences added some legitimacy to the process. Regulations such as compliance to Sarbanes/Oxley changed the dynamics of the RFP and actually led to some wins for the company. Again, the market was in flux, experiencing transitions on many fronts – elements of the "old guard" slowly being displaced by the new.

Competition in the cable industry is driven by a variety of factors, both among rival network operators and rival suppliers. The competitive landscape for the MSOs, while traditionally "local monopolies" changed dramatically with the switch to converged services, with increased intensity of competition from traditional telecommunications companies also offering converged services. Among suppliers, a crowded arena of suppliers was getting more crowded. The move to converged services, combined with a major slowdown of telecommunications spending in the early 2000's, also changed the landscape of competition as traditional telecommunications suppliers were now looking to make inroads into the MSO market. The result was a fiercely competitive landscape with strong buyer power, highly cost competitive, with pressure to get products to market. Fueling competition further, MSOs maintained a competitive atmosphere among its suppliers as a strategic initiative to help keep capital expenditures for network build-outs under control.

In summary, the culture of the traditional cable market, when considered in conjunction with macro industry trends, was driving the industry towards increased competition among old and new suppliers and a higher demand for quality gear, while retaining their focus on costs, all at once.

Market Access

Access to customers ranged from very good to very difficult depending on customer. Luckily, with few key customers the company enjoyed early and often access, and with high quality information. This was a result of a general willingness of certain MSOs to interact with new suppliers, but also of hard work establishing relationships and hiring of key individuals with contacts and industry experience. Market access could also be attributable to the fact that cable networks are geographically isolated, "local monopolies" – without regional competition from other cable operated MSOs, customers tended to be highly open with their preferences for both current and planned technologies.

While the marketing function was geared towards downstream marketing activities – demonstrations, trade shows, press releases – the company had access to customers via the technical management: founder/CTO and through VP and Director of Engineering. The company elicited direct contact with engineering management and customers through travel, face-to-face meetings, email, and phone conversations. Customers took advantage, many times requesting features from the "white-coats" directly. A particular regional network operator in Greenbay, Wisconsin once joked that the company should "pay [him] for GUI development."

Market Intelligence

The direct access between MSOs and the company was exploited in both directions. Many times new product concepts and proposed product features were tested in front of "trustworthy customers" during concept development. In fact, many "half-baked concepts" were turned down at this stage. A discussion with a customer in fact drove a key technological shift for the Plexis product line - a switch from QAM modulation to baseband transmission. "Great idea, just not popular. The entire market is going baseband."

While good access to customers permitted high quality customer feedback, the transitioning market from television only to "triple-play" services and merging of cable and telecommunications networks inevitably led to fickleness on behalf of the MSOs as they contemplated their next moves for network transition, convergence, and upgrades. In the case of Plexis, a very large and influential customer expressed strong interest in a particular linecard, such that the company decided to oblige. Ultimately the customer changed course, but as luck would have it, another customer became interested. Much later, the original customer decided to pursue the technology after all.

As a further indication of the accuracy of information, the engineering director described aspects of this market as the "wild west". The market ranged from formal acceptance to rogue regional network architects ignoring the policies of the corporate central technical headquarters.

B.1.2.2. Enterprise Attributes

Process Strategy

The strategy of the Plexis development effort was "customer traction", a flow-down of enterprise acquisition strategy. The idea was to be cost and quality competitive on every bid, creating brand recognition for the company among its larger competitors, particularly by taking margin out of competitors' wins. Generally, the company's strategy was to be enough of a pest to larger competitors such that an acquisition might be preferable. In fulfillment of this strategy, the company was willing to sell gear at considerably lower margins than might otherwise be common.

The company spent almost two years demonstrating the Plexis system at various customer meetings, trade shows, lab trials and demonstrations. (The travel chassis became known as the "battle chassis".) Preparing for a continuum of demonstrations in pursuit of the customer traction strategy "confused product development – are we making a product to sell, or are we making a good looking, slightly functioning prototype?"

Without reliable or predictable revenue streams the survival of Company X was literally at stake. To that end, risk taking was applauded. Where mistakes were made, encouragement followed.

Enterprise Resources

<u>Development Process Budget</u>

The company was funded well considering the troubled times, albeit less than established competitors. After the telecom industry slow-down following the dot.com crash in the late 1990's, the process for getting monies approved became highly bureaucratic, forcing management to campaign aggressively for project needs. Interestingly, the Plexis development project had no formal budget. Monies were requested and were sporadically granted – sometimes requests were filled quickly, sometimes not. At times the board was fickle, but ultimately development efforts were adequately financed.

Additionally, the investors were particularly fickle when it came to head count – once approving a requisition for an embedded software developer, stimulating a month long effort of interviews and selection of a candidate – only to repeal the requisition at the next board meeting. This type of fickle behavior caused uncertainty for management. The director commented, "do I have seven people, or do I have eight?" Of all resources, the board was particularly concerned with headcount, a major contributor of the burn-rate.

Infrastructure

The board supplied many of the functional needs of the organization, including facilities and IT, much of which was provided early on when funding was in plentiful supply. The facilities were older but spacious and pleasant, and under a long-term contract. Test equipment was kept current with project needs. The IT services were part-time and sub-contracted. The engineering director remembers them as "adequate", but that "they were not stakeholders" of the process and "that was a mistake."

Supply Chain

The company wielded little power over suppliers as they were perceived to be a tier 3 player (Plexis had little sales volume to speak of, and very few well-known customers). Generally, the company struggled with lead times, had difficulty receiving allocation of materials, and contended with tier 2/3 pricing and support. Several key component vendors proved difficult to work with – some willing to sell parts to the company, only without support, others *refusing* to sell components to the company altogether. Alternatively, many lesser-known suppliers were quite willing to do business with a small company, some even actually marketing to small companies as a part of their business model. In some cases, vendors were influenced by investor relationships or reputation, in other cases vendors catered to the company as favors to employees from business dealings from previous employment.

Enterprise Organization

A dominant organizational characteristic for development processes at the company was the influence of the board. The board, which met once per month, included junior- and mid-level representatives from the first and second round venture firms. Between monthly meetings the representatives were required to report back to their respective firms, in turn. In this manner a hierarchy of interests was always represented in the board room, except that the views of higher tiers, which were being collected in between meetings, were not expressed until the next meeting. Consequently, many critical decisions were delayed by an extra month. Often the board was inconsistent in behavior, and commonly a source of uncertainty for the team.

The board was very involved from a budget perspective, and was a provider of many enterprise supportive processes. On the whole, the board left the technical operations of the company to the cofounders and the management team.

B.1.2.3. Team Attributes

Process Participants

Experience

Founded from a dedicated research team at a government research laboratory, the original team was highly academic and research oriented, with an unusually high percentage of doctorate degrees. The team was highly specialized mainly in optics, but also with competences in hardware and software. As an optical components manufacturer, such a team was appropriate.

When the company shifted focus from components to systems, specialized expertise in optical components became less relevant. The team was restructured to include individuals with cable industry experience, particularly individuals with backgrounds in chassis/linecard architectures and analog electronics designers with specialized QAM expertise. The original Plexis development team was comprised of hardware, software, optical, and mechanical engineers. The hardware team (3-5 individuals) had expertise in analog (QAM) and digital electronics. The software team (8-10 individuals) had expertise in low-level drivers, embedded CPU, operating systems, application code. The optics team had expertise in laser technology, laser modulation, and receiver technology. The mechanical team was also expanded to include chassis design experience (sheet metal and fabrication). Where learning curves were too high, or specialized experience was needed, the company looked externally (experienced cable and telecommunications personnel) to fill in gaps. Later, these organizations were further adapted to include high-speed digital and RF electronics, and user-interface development.

Personality

The Plexis development team was just large enough such that many varieties of personalities were present at once. The product architecture complemented the team well, also just large enough so that mostly everyone could find their comfort zone in terms of roles, responsibilities, and preferred work styles. With one or two exceptions, no major personality problems or conflicts impeded development.

Attitude and Motivation

In general the company was composed of dedicated, hard working people. The startup work atmosphere is typically energetic and attracts certain types of individuals, many of which are motivated by the potential up-side (stock options and grants). Individuals have a heavier stake in things and are well aware that there is more "on the line". Also, as part of a small team there is typically a high sense of personal contribution, and more peer pressure to perform well. As such, the company was team-oriented with a "rise and fall together" culture. In such an atmosphere there is typically less of a burden on management to motivate employees.

With some exceptions, change was taken in-stride. Most individuals were flexible in terms of personal responsibilities and work. After the switch to systems, much of the team adapted their roles and began to fill in where necessary, diligently coming up to speed on a variety of new technologies ("buying books on SNMP").

The team did experience troubled periods, particularly during times when the team was disrupted during changes in technical direction (from components to systems, from QAM to baseband modulation, and finally to integrated optics). The company expected team members to adapt to new roles, but in certain instances key team members did not see themselves taking on new roles. Elsewhere, the company decided to seek experienced individuals externally, and ultimately let go of several members of the optical and hardware teams. Even without a change in direction the board did their best to inhibit motivation. A week after the company's first purchase order for Plexis, the board decided to cut salary for all employees by 2%!

Organization and Processes

Team Organization and Composition

The board was responsible for significant turnover in management. (In a certain two-year span, the board replaced VPs of Operations and Engineering, a program manager, and a configuration management coordinator.) Despite the variability caused by the board, the team saw one constant in the leadership of a highly motivational and respected company President/co-founder.

As stated above, the Plexis development team was comprised of hardware, software, optical, and mechanical engineers. With the exception of contract manufacturing, the team used a low percentage of contractors. The team organization was kept current, as management "literally tore up the whole thing and started over" after several shifts in technical strategy and direction. On the whole, the Plexis team was highly cooperative and cross-functional, perhaps best described as *flat*. While functionally organized on paper the team was operating with high concurrency across functional disciplines. The Plexis team functioned particularly well at key architectural interfaces: embedded software and hardware, applications and network management software, RF electronics and optics, electrical and mechanical engineering. Team was collocated (same facility, same floor) with several core team members even sharing office space. The director recalled core team members who were making meaningful contributions in multiple functional areas. The director recalled one individual in particular who was qualified in hardware, software and optics. "When you have guys like that on the team it makes divisions between functional groups even hard to distinguish."

Team Processes

The rhythm of the development atmosphere was fast paced and entrepreneurial, with risk taking encouraged and mistakes quickly overcome. In lieu of formal processes, the team relied on the past experiences of key personnel, many of whom had been brought in from larger companies, some from high-disciplined industries. When discipline was critical these individuals would offer checks to help "pull in the reigns". The director recalls the VP of Optics (referring to the lack of discipline) once literally saying, "This time you've gone too far."

Development efforts at the company were contributing to an emerging knowledge base. The engineering team created a home-grown framework for common drives and configuration management. The process was rudimentary but custom designed, without enterprise systems for database or business processes. The team created its own parts inventories, numbering conventions, and versioning repositories. Many of these processes were strictly followed, while loosely enforced. The process was efficient but highly vulnerable, with read/write capability to all team members on common drives – at least once a developer wiped out the entire parts database. The director recalls that the system was perhaps exceeding its own limitations by the time of the acquisition. "The process worked because the team was small: one layout guy and just a few schematic designers. If we had one more guy – disaster."

The engineering team had to establish their own "internal operations" for prototype stages and final assembly: purchasing (including lead-time mitigation), prototyping, materials tracking, etc. The enterprise did not provide a separate function for components (verification, mechanical, or reliability) and the team worked around it. Sub-assembly manufacturing operations were initially outsourced (out of state) and ultimately full assembly was transitioned to the contract manufacturer. The company relied heavily on the manufacturing organization for supportive processes in production volumes: material handling, purchasing, and inventory.

B.1.2.4. Product Attributes

Dynamics of Planned Technologies

<u>Timeline of significant technological dynamics and transitions:</u>
1996 – Company founded to manufacture optical components for data communications industry.
2000 – Enterprise strategy shifts from optical components to optical networks and systems, Plexis Chassis concept born.
2002 – Plexis product line switch from broadband (QAM) modulation to baseband modulation over fiber.
2003 – Plexis product line switch from "stick-built" optical/RF circuits to integrated, pluggable optics modules.

The Plexis platform experienced an evolution of technological change, beginning as a technology-push product, and ended up as a market-pull product. A brief synopsis of this evolution follows. The company first began as a component vendor, manufacturing optical gratings targeted towards the optical communications market. After the telecom downturn of the late 1990's, needing to reinvent itself the company identified opportunities in the cable MSO market, and transitioned to systems in 2000. The transition to systems saw the birth of the Plexis platform product concept. Plexis was at first a technology-push product. Based on the expertise of the development team, Plexis transport cards utilized QAM modulation techniques to drive data onto the fiber. The

QAM strategy worked well, but was non-standard and not received well by the market. After experimenting with customers for some time, the Plexis changed from QAM modulation to baseband transmission (GbE/SONET) – effectively transitioning to a market-pull product. A final transition came about when the team decided to use off the shelf, integrated, pluggable optical modules, in lieu of custom circuits which provided better, but unnecessary improvements in performance. In the early days of Plexis, the optical strategy was to build custom circuitry to launch and receive optical signals (i.e. distinct laser, driver, TEC, and receiver diode components).

While easing the burden of development on the team, the selection of integrated modules presented challenges. The technology integrated optics had been evolving for some time – well established vendors offered reliable (but bulky) modules (MSA 300), but were transitioning to emerging standards with compact form-factors (XFP, xPAK). These new technologies were desirable and considered low-risk technology, but were costly (upwards of 1/3 of total BOM), available only in low quantity, and still very new to the team. The selection of optical module and other key components were particularly critical to the final costs as the choices made were principal determinants of final product cost.

Product Architecture

The Plexis platform is a chassis based, backplane/linecard system with several replaceable linecard modules – data transport, amplifier, combiner/splitter and management capability – all with embedded software applications, firmware code, and remote management capability (human and machine UIs). The Plexis system was composed of a network of interconnected chassis, geographically dispersed, such that a regional metropolitan communications system was created. The network created provides reliable regional transport of video, telephone, and data services to subscribers of the MSO's service offerings.

The primary architectural components of the Plexis system included the chassis, backplane, network manager linecard, transceiver linecard, and a software-only network management system (NMS). The Plexis chassis was available in two form factors, 1 and 4 RU. These chassis provided mechanical stability and accommodated functional partitioning of the major architectural assemblies.

The chassis backplane distributed power to all linecards, and provided control plane communications paths from the network manager linecard to all data forwarding (transceiver, amplifier, etc.) linecards. No data forwarding mechanism was built into the Plexis backplane.

The network manager linecard architecture was composed of a COTS single-board computer with access to the chassis backplane. The single-board computer ran chassis application code which included linecard provisioning (including upgrades), health, and monitoring, and also the network management application which provided communications to the NMS. The NMS was a desktop application and served as the control and monitoring interface to the user-operator of the chassis. The NMS provided remote connectivity to all chassis in the Plexis network.

The transport linecard architectures are logically broken into two planes, control plane and data forwarding plane. The control plane included hardware and software components: embedded microprocessor with access to data forwarding components (for provisioning, operations, and maintenance) and access to the backplane for communications to the network manager. The data forwarding plane included optical transmitters and receivers, optical to electrical converters, and data processing ASICs.

The Plexis system was marketed to customers as an enabler for extension of high-dollar services to existing networks, standardized and interoperable for ease of deployment, at a reasonable quality-to-cost ratio. Plexis was backed by a high service level from the technical team and with customizable features per requests from customers. The main architectural features related to the value proposition were standardized interfaces, data grooming and formatting ASICs and transceiver components, general flexibility of the linecard system and overall system stability. While highly overlooked at the onset, perhaps the most critical determinant of Plexis success was the NMS software, which ultimately became the most heavily trafficked user interface in the system, but was in reality an afterthought of the product system architecture.

Product Architecture vs. Team Organization

As a linecard system, Plexis was highly modular with discrete boundaries between functional hardware and software elements. This architecture drove the organization of the team. After the transition to systems rippled through the team structure and composition, the team quickly adapted roles and hired individuals to fit within the Plexis architecture. When following technological transitions occurred, the team adapted again. Integral role for individuals with respect to the product architecture were backplane communications designers, middle-ware developers (communications between linecards and single board computers, and between single-board computer and NMS), and finally team leaders on linecard development. Finally, the software team developing the NMS desktop application became intimately involved with the customer, but also highly visible to and critiqued by management.

Realizability

The manufacturability processes are well established for embedded electronics assembly, but the team shared a burden with coordinating with outsourced manufacturing. Prototypes were costly but generally built with ease – perhaps the biggest challenges were procurement of materials, particularly long-lead items. In production, initially only sub-assembly manufacturing was outsourced, while sub-assembly test and optical assembly performed by the team at the local facility. Eventually, even these processes were off-loaded to the contract manufacturer. While coordination was necessary, in general standard process and low-volumes were such that manufacturability did not drive the process.

Reliability & Acceptance
In the cable space there are formal guidelines for quality and acceptance, but generally these guidelines were achievable for the company. Plexis was neither safety nor mission critical, but was expected to survive five to ten year deployments. Depending on the customer, qualification and acceptance ranged from customer visits to "kick the tires", to formal qualification at MSO technical headquarters. The director notes that in many cases customers recognized the company as a startup and "they took it easy on us."

B.2. Case Study B: ProjectX Development

ProjectX is an ongoing development effort which began in 2006 at a government contractor developing subsystems for deployment in satellite-based communications applications. The company, which commonly participates in the prime/subcontractor hierarchy of government and DoD contract work, typically is involved in several development efforts for a variety of products at any given time. The interview was held in a discussion format with the ProjectX Program Manager during a single session on April 27th, 2008.

B.2.1. ProjectX Development Stakeholders

B.2.1.1. Agents

Indirect agents include the prime contractor, and its internal management of the subcontractor providing ProjectX. The prime is accountable to the customer (the government) for a completed system, meeting all requirements, on schedule. Also, the subcontractor enterprise and its shareholders are considered indirect agents, providing all resources to the team. The enterprise desires successful completion of subcontracts to secure future business and to provide a return to shareholders. Direct agents include management and technical stakeholders at every level of the subcontractor hierarchy.

B.2.1.2. Beneficiaries

Beneficiaries include the government customer, and presumably its citizens, as well as the prime for its strategic business purposes and its shareholders.

B.2.2. ProjectX Development Project Context

The project of interest for the interview is ProjectX, an electro-mechanical controller and satellite telemetry subsystem, targeted for low-power spaceflight applications. Development of the system began in 2006 and continues in late project phases at the time of this writing. The interview was conducted from the perspective of the embedded software design team (3-5 developers), which covers 24 months of software development, concept through qualification.

B.2.2.1. Market Attributes

Market Segment
The market segment for ProjectX is the American aerospace/government contracting industry, which is widely known to be a highly formal, reliability driven, and bureaucratic marketplace.

The industry has seen significant consolidation over the last several decades, leaving a small number of very large enterprises controlling the lion's share of complex systems integration contracts. Commonly, the government issues RFPs to potential primary contractors who are evaluated on systems integration, product architecture, and process capabilities, as well as cost and schedule projections. Once a primary contractor is selected (the "prime") as the systems integrator, the system is partitioned into components of varying complexity. These sub-systems are then contracted out to various subcontractors, who either produce the sub-system in its entirety, or may in turn sub-divide components once again, passing work down to sub-subcontractors. Multi-level subcontracts are commonplace in this context.

Catering to the government or to prime contractors in this industry requires adherence to an enormous bureaucracy of specifications governing processes, standards, testing, and compliance.

Agreements
Contract terms, which have evolved to protect suppliers from the inherent risk of costly, complex systems, are generally structured to guarantee certain funds to vendors provided that broad contract terms are met. In the case of ProjectX, the contract terms are cost-plus with a specified profit margin. In this type of agreement, the supplier agrees to a set of deliverables over a specified time frame, the costs of which are guaranteed by the government. In addition to these costs, a profit margin is guaranteed, but only as a percentage estimated cost at the time of contract signing. In this manner the supplier is protected from cost overruns, but is not guaranteed a certain profit margin (as a percentage of cost). Such contracts guarantee set funds to suppliers even in the event that the project is cancelled by the government.

Market Access
In a contract situation the customer is highly accessible, whether it's a subcontractor, the prime, or the government itself. A regulated process ensures regularly scheduled (weekly) meetings, review, and deliverables, typically conducted formally with open dialogue in face-to-face discussions. For ProjectX, access to the prime was not of concern; in fact, the prime assigned a full-time program manager to oversee the project first hand.

Market Intelligence
Every level of the subcontractor hierarchy inherently represents a separation of development teams. The primary mode of communication across these boundaries occurs in the form of systems specifications – every time a subcontract is created, a new

specification is written as a flow-down of higher-level requirements relevant to the subsystem. Consequently, misinterpretation is likely within each level of translation, especially considering that such boundaries commonly are inter-enterprise with geographical separation. Many facets of the software architecture are largely dependent upon systems specifications regarding interfaces. Due to the hierarchy of subcontractors, some churn of systems specifications should be anticipated as requirements are scrutinized.

B.2.2.2. Enterprise Attributes

Process Strategy
The greater technical system within which ProjectX operates was actually bid to the government as a strategic development project by the prime, who was attempting to adapt current capabilities into a new business area. Consequently, the prime underbid the system to win the new business. Alternatively, ProjectX was an established product capability for the subcontractor, who was not initially consulted about the strategic intentions of the prime. Later, to help compensate for its underestimation, the prime resigned to offload some of the burden of documentation to allow the subcontractor to move forward more efficiently.

Enterprise Resources

Development Process Budget
Widely understood as over-budget, the financial resources available for headcount on ProjectX were managed tightly, while the budget for equipment and materials remained largely unconstrained. Drivers of the project budget were the terms of a cost-plus contract and by the project strategy, but also largely driven by headcount as opposed to other project needs. The interview candidate recalled instances where purchasing decisions were weighed against time, citing that for $1k-10k items (occasionally even $50k) decisions were made relatively easily. Conversely, the burn-rate due to headcount was monitored closely, as software developers ran $200/hr factoring overhead for such a large company.

Infrastructure
The subcontractor supplied the development team with an abundance of generic infrastructure (i.e. office and facilities, IT resources, CAE tools). For specific infrastructure, particularly high-end test fixtures, cost restrictions constrained test capabilities, driving test and qualification strategies.

Supply Chain
The subcontractor's brand power, along with that of the prime, wielded significant influence over suppliers. Consequently, access to products and technologies were not a concern as the project proceeded. The space industry does suffer however from a component and services selection perspective in that special materials and capabilities are required (i.e. RAD hardened, low-power, light-weight, high temperature). Additionally,

volumes are typically low (ProjectX is a one-off deliverable) such that pricing in such instances is typically high. In this context, manufacturers of systems typically are not going to pressure suppliers from a cost perspective, but will demand excellence from suppliers from a test, lot, and certification perspective.

<u>*Enterprise Organization*</u>
The subcontractor is a large organization, with multiple projects running simultaneously, many of them much larger than the ProjectX development effort. The company provided all of the vital supportive processes necessary for ProjectX, including procurement, qualification test, operations, delivery and deployment.

Ironically, in such a supportive culture with large development teams, "engineers become very functional, and very pigeon-holed. It's interesting trying to run a small program in that environment – a small project lacks broad based expertise because with a lot of people no one ever had to learn [general expertise]."

B.2.2.3. Team Attributes

Process Participants

<u>*Team History*</u>
Approximately six months into development, the ProjectX development team was reorganized after schedule slips and cost overruns had accumulated project wide. News of the difficulties has risen to VP level, and by request of the contract prime the program manager at the company was replaced. The new program manager, known to the prime as "the Wolf" for his abilities and reputation as a closer, was a senior level engineer with vast knowledge, experience, and power within the organization. Consequently, the Wolf restructured the development staff, replacing 50% of the designers. The software team was restructured during the transition, having its lead software engineer replaced by the functional manager of the software engineering group.

<u>*Experience & Personality*</u>
The software team was responsible for hardware driver, register level coding through upper layer command and control interfaces. To conserve memory space, the team designed a single-threaded application (with no operating system). At higher layer functions, each designer was assigned a major function and saw it through from inception to delivery.

The lead designer had superb experience, mentoring, and professionalism to the project. This individual was actually the functional software manager, whom the Wolf managed to have temporarily reassigned as the software lead on ProjectX. The lead designer's experience became particularly critical when it came to the various paperwork deliverables, who really accelerated the writing of the various documents (i.e. SDD, SRS). The team's other developers were relatively younger and less experienced, approximately

5-6 years of experience each, but talented and energetic. Together, the team managed design and coding efforts with efficiency and competence.

Attitude and Motivation

Design work on satellite hardware is particularly challenging for engineers due to the many constraints particular to radiation hardened components and rugged materials. Often, suitable components are generations behind equivalent commercial technologies in terms of speed, density, and capacity. Space companies typically attract engineers who, if not for the challenge of working around all of those constraints, enjoy the industry due to the allure of working on spaceflight systems (which tends to compensate for many of the aforementioned difficulties).

As a result of the nuances of contract negotiations in the aerospace industry (as touched on above), design teams inevitably commence efforts behind schedule. In the case of ProjectX, shortfalls in project planning resulted in two years of extended overtime. Morale was a major concern for management, and factored into weekly decision making as time progressed. The team suffered from low morale up until the restructuring, but changed for the positive when the Wolf came on board – even while the Wolf stepped up the pressure on the team. The Wolf held a mutual respect with the engineers and protected developers from management. As a result, the team served him well with positive attitudes and perseverance.

Organization and Processes

Team Organization and Composition

After the reorganization, they benefited from strong leadership, both at the project management and team lead levels. The project was lead by an individual who had a history and reputation for success within the company. The Wolf knew the company well, was aware of who the strong individuals were, and was powerful enough to assemble a team of above average talent. The software development team was led by a highly experienced individual, who was actually reassigned out of functional management temporarily to lead the ProjectX effort.

The software team was structured as a single lead developer, and 2-3 full-time software engineers depending on development phase. In later stages, a quality engineer was added to the team for review and monitoring of test administration for acceptance and compliance purposes. The software team was collocated, and shared lab space with the hardware team. The company did not sub out any of their electrical engineering or embedded hardware.

Team Processes

Broadly, ProjectX and other similar complex high-reliability efforts operate such that project related efforts (people, processes) have cost and schedule responsibility, while functional efforts (people, processes) have quality responsibility. For example, the project manager drives cost and schedule targets while the functional manager drives quality targets, each effort working as a "checks and balances" complement to the other.

The project is highly managed, and highly tracked for progress, cost, and risk mitigation over time. Technical risk is identified up front, with attributes of probability and impact. A "burn-down plan" is created to tackle probability and impact while mitigation results are closely tracked as the process evolves. Cost and schedule risk, or program risk, is also closely monitored and evaluated as time progresses. All metrics are under constant scrutiny by the management team during weekly "page and line" progress meetings. Project tracking data is used to identify problem areas and critical paths, and for comparison of actual to estimated costs and schedule.

Guided by CMMI and DoD specifications commonly found in aerospace and military contracts, the company designs formal processes for development projects. These processes are published to the prime during the bidding process as a capability offered by the subcontractor. The prime reviews such processes as part of the bidding and review process, and agrees in contract that the subcontractor will follow their published processes.

The process environment of ProjectX is highly rigid and bureaucratic, with a comprehensive schedule of reviews, tests, and deliverables (paper and product). Requirements are derived, specified, decomposed, and flowed-down, then linked through the subcontractor hierarchy and audited by the customer. Formal design reviews are held at multiple stages, after which an "action item committee" forms to enter action items into a database for formal tracking. Shall's, should's, must's, and will's abound, followed by tests, analyses and/or demonstrations. A tremendous amount of time is consumed writing, reviewing, in preparation for, and in adherence to such a process. Such a process becomes culture for the organization. Acronyms and process jargon dominate regular conversation among individuals.

B.2.2.4. Product Attributes

Dynamics of Planned Technologies

<u>Technical Dynamics</u>
ProjectX was originally conceived as a hardware only box running state machines encoded in field-programmable gate-arrays (FPGAs) with hard coded steps, and did not include and software. As various complexities were uncovered during system design, the product architects decided to improve the flexibility of the design by using an embedded processor, requiring software development.

For software development, the detailed work of coding up an architecture is hinged largely on electrical engineering decisions about component selection, namely choice of microprocessor, complex ASICs, and other configurable hardware elements. To mitigate risk, the team chose to reuse a proven processor core, one that had been flight proven on past projects. Once this decision was made, there was very little expectation of change going forward. Indeed, the software team has not encountered any changes in hardware components or hardware architecture on the project to date.

The software design had been reworked occasionally, but mostly to negotiate bugs as part of routine problem solving and detailed design. For example, the team found that a requirement of 50ms interrupt timing was not long enough to process all of the telemetry data acquired – an oversight in the flow-down of system requirements. Luckily, the interrupt timing was adjustable in the code, and the remaining architecture quickly adapted to the elongated timing.

Downstream, in the context of high reliability there is a possibility that the product could fail in the harsh conditions of environmental testing. ProjectX is currently in environmental testing at the time of this writing, and the candidate expressed some fear that failures in test could drive technical change.

Technical Certainty

The ProjectX platform design was highly conservative in its use of components and technologies. As stated above, a heritage processor core was selected, along with a known 1553 ASIC. In most critical component selections, hardware was selected first based on test qualification risk aversion.

Ironically, the processor core was of an older vintage, somewhat slow, with limited memory. Cost-benefit analyses revealed that qualification of a new core would have cost ~$1M, versus a projected $75k/month on software development on the old core for 24 months ($1.8M) – leaving $800k residual for software development on a new core. It was determined that the old core would be used. Consequently, feasibility analyses were performed on the software side to see if new code would fit, but ultimately the cost impact of a new core was primary driver for decision making. Ultimately, significant hours of development were spent writing efficient code such that the software requirements were fulfilled on the heritage processor core.

Product Architecture

Product Concept

ProjectX is a packaged electronics assembly which is mechanically attached to host satellite structure for space flight. The product controls electromechanical devices and collects satellite telemetry data. Internally, the product utilizes an embedded processor core, and associated circuitry to run software algorithms which process, collect, and report mission data. Externally, the product interfaces electronically to electromechanical devices and to a host satellite computer.

Decomposition of Function and Form

ProjectX is an embedded processor sub-assembly packaged in a ruggedized mechanical chassis. The chassis has external machine interfaces for electro-mechanical control and system bus (1553) interfaces for relaying of telemetry data to the host satellite computer.

Internally the ProjectX architecture can be decomposed to hardware and software subsystems, each partitioned into logical functions. The hardware subsystem is composed of a power plane, control plane (embedded Intel i286 processor core, 1553 communications ASIC), and a data plane (telemetry processing). The embedded software subsystem can be decomposed into boot, board support (hardware level device drivers) and application code (three primary functions – fault response, conops, and the mission function). Alternatively, the software architecture can be viewed as external interfaces and internal functions. A large fraction of the software functionality was specified explicitly in systems design (particularly, fault response). Other functions were more loosely defined through a flow-down of systems specifications (primary interfaces, i.e. conops).

Mapping of Attributes to Value Delivery
The value of the architecture is manifested high quality telemetry data, high reliability in harsh environments, low power, CPU system stability, and interoperation within the satellite system architecture. Externally delivered functions through machine interfaces (1553 operation, conops standards compliance) are critical and highly specified. Internally, the mechanisms used to deliver value are highly transparent to the customer. While an embedded CPU was chosen over FPGA algorithms for capacity and versatility, the exact balance between software and hardware features from a customer perspective is quite arbitrary.

Product Architecture vs. Team Organization
The ProjectX development team was functional and specialized. The interview candidate felt this was mostly appropriate per the product architecture. To compensate for the emphasis on functional organization, the subcontractor assigned a systems engineer as a liaison between functional groups on development teams. In fact, the subcontractor had a systems organization whose primary responsibility was serving this role. All negotiating among functional teams of development features, selected components, and functions were done on paper and managed by the systems engineer.

As a valued feature, high quality telemetry data was particularly critical. Delivering this quality relied on coordination and communication between embedded hardware and software teams. The interview candidate felt that this coordination was vastly improved by the individuals (hardware, software, and systems) put in place by the Wolf.

Realizability
A common challenge of satellite systems manufacturing is that mechanical structures optimized for Zero-G atmospheric pressure can commonly not be assembled in their entirety while on the ground. This limitation is a strong driver of the process from a test planning, test step, and simulation perspective.

ProjectX is a custom product slated for very low volume production. The low volume afforded the team many opportunities with regards to production, for example, "hand-wiring" was a common step in the production process. In this manner, no unusual intensity of relationship was required between manufacturing and development engineers.

The interview candidate expressed that extreme measures were taken to ensure *reworkability* – components were chosen such that they could be replaced easily if found to have latent failures during environmental testing. All flight hardware underwent harsh-environment testing to weed out infant mortality. Therefore, the hardware team maintained an aversion to ball-grid arrays (BGAs) and other components difficult to rework.

Reliability & Acceptance

Software testing is governed by process, beginning with a draft software test plan required at the preliminary design review (PDR). The draft test plan specifies what kind of hardware configuration will be used for software test, the number of developers and quality personnel required to administer the tests, and the general test philosophy and approach. Later, SRS requirements are flowed down into the test plan, which ultimately becomes a procedure that is performed by a developer and initialed by quality person. The process requires 100% coverage of SRS requirements, which actually only represents a subset of tests, as fault testing includes every possible branch of software execution. In the case of ProjectX, which was a 24 month development effort overall, three to four months are spent in software qualification and test. Acceptance tests are performed locally at the subcontractor's site, with the prime present during test time. This practice is common along the entire subcontractor supply-chain.

The hardware test methodology is driven by a desire to expose infant mortality of components. Test processes employed by the subcontractor are formally defined (MIL-STD 1540c – Test Requirements for Launch)

C. Objective/Behavior Comparison Tables

C.1. Lifecycle Comparison

	Objectives and Goals for Stakeholder Value Delivery	Lean Behavior	Identified as Waste Objective	Identified as Value Objective	Addresses Objective with Behavior	Behavior Explicit	Flexible without Behavior	Rigid without Behavior
Lifecycle Value Delivery	*Enterprise Connectedness*							
	Coordination with Vital Supportive Processes	Synchronization Communication Task Management.prioritization Process Fluidity.all	1	1	1		N/A	N/A
	Cross-Functional Communication and Cooperation	Synchronization Communication Task Management.prioritization Process Fluidity.all	1	1	1		N/A	N/A
	Inter-Disciplinary Communication and Cooperation	Synchronization Communication Task Management.prioritization Process Fluidity.all	1	1	1		N/A	N/A
	High Coordination with Manufacturing	Synchronization Communication Task Management.prioritization Process Fluidity.all	1	1	1		N/A	N/A
	Process Capability							
	High Process Efficiency	Sychronization Communication Task Management.all Process Fluidity Information Strategy/Knowledge Management Process Fluidity Process Fluidity.Concurrent Engineering Monuments	1	1	1		N/A	N/A
	Accommodate Pressure/Urgency	SEE Efficiency	0	0	0		Y	N/A
	Value Management & Process Transparency	Synchronization Communication Task Management.all	0	1	0		N/A	N/A
	Risk Management & Mitigation	Task Management.prioritization Design Alternatives	1	1	1		N/A	N/A
	Enterprise Learning and Knowledge Management	Employee Value Information Strategy/Knowledge Management Continuous Improvement.training	1	1	1		N/A	N/A
	Process Flexibility	Synchronization Task Management.prioritization Design Alternatives Process Fluidity.IPTs Customer Focus Monuments.flexibility of methods	1	1	1	Y	Y	0
	Process Scalability	SEE Efficiency SEE Flexiblity	0	0	0	N	Y	N
	Acceptable Process							
	Acceptable Process – Market Perspective	Synchronization Communication Task Management.prioritization Monuments.flexibility of organization Monuments.flexibility of methods	-1	0	-1	N	Y	N
	Acceptable Process – Product Perspective	Process Fluidity.all Monuments.flexibility of organization Monuments.flexibility of methods Continuous Improvement.training	0	1	1	Y	0	N/A
	Acceptable Process – Enterprise Perspective	SEE Efficiency Employee Value Task Management.prioritization Monuments.flexibility of methods	0	1	1	N	Y	N
	Acceptable Process – Participant Perspective	Communication Design Alternatives Employee Value and Empowerment Continuous Improvement.all	1	1	1	Y	N/A	N/A

Table C.1 Lifecycle Objectives/Lean PD Behavior Comparison

C.2. Upstream Comparison

	Objectives and Goals for Stakeholder Value Delivery	Lean Behavior	Identified as Waste Objective	Identified as Value Objective	Addresses Objective with Behavior	Behavior Explicit	Flexible without Behavior	Rigid without Behavior
Upstream Value Delivery	*Upstream Market Goals*							
	Reduce Market Uncertainty	Synchronization Task Management.prioritization Customer Focus Continuous Improvement.training	1	1	1	Y	Y	N/A
	Reduce Market Ambiguity	Synchronization Task Management.prioritization Customer Focus Continuous Improvement.training	1	1	1	Y	Y	N/A
	Maximize Information Quality	Information Strategy/Knowledge Management	1	1	1	Y	N/A	N/A
	Upstream Team Goals							
	Participant Education – Market Awareness	Employee Value Information Strategy/Knowledge Management Continous Improvement.training	-1	0	0	N	Y	N
	Participant Education – Technical Proficiency	Employee Value Information Strategy/Knowledge Management Continous Improvement.training	1	1	1	Y	N/A	N/A
	Individual/Product Structure	Monuments.flexibility of organization Process Fluidity.IPTs Continuous Improvement.training	1	1	1	N	Y	N/A
	Team/Product Structure	Monuments.flexibility of organization Process Fluidity.IPTs	1	1	0	N	0	0
	Specialized Expertise	Process Fluidity.CE & IPTs Continuous Improvement.training Continuous Improvement.incentives	0	0	0	N	0	0
	Minimize Perceived Complexity	Process Fluidity.all Task Management.all Continuous Improvement.training	0	1	1	N	Y	N
	Upstream Product Goals							
	Reduce Technical Uncertainty	Task Management.prioritization Design Alternatives Continuous Improvement.training	1	1	1	Y	N/A	N/A
	Minimize Product Costs	Task Management.prioritization Continuous Improvement.training	1	1	0	N	Y	N
	Minimize Production Costs	Communication Process Fluidity.all Continuous Improvement.training	1	1	1	Y	N/A	N/A
	Prioritization of Product Architecture Development	Synchronization Communication Task Management.prioritization	1	1	1	Y	N/A	N/A

Table C.2 Upstream Objectives/Lean PD Behavior Comparison

C.3. *Mid- and Downstream Comparison*

	Objectives and Goals for Stakeholder Value Delivery	Lean Behavior	Identified as Waste Objective	Identified as Value Objective	Addresses Objective with Behavior	Behavior Explicit	Flexible without Behavior	Rigid without Behavior
Midstream Value Delivery	*Midstream Market Goals*							
	Accommodate Market Uncertainty	Synchronization Design Alternatives Task Management.all	0	0	1	N	Y	N
	Accommodate Changes in Customers' Desires	SEE Flexibility	1	1	1	Y	Y	0
	Midstream Product Goals							
	Accommodate Changes in Technology	SEE Flexibility	1	1	1	Y	Y	0
	Exploit Certainty	SEE Efficiency	1	0	1	N	Y	N
	Iterate Well	Synchronization Communication Task Management.all Process Fluidity.all	1	0	1	N	Y	N
Downstream Value Delivery	*Downstream Market Goals*							
	Generate Feedback	Synchronization Task Mangement.prioritization Customer Focus	1	1	1	Y	N/A	N/A
	Deliver the Architecture to the Customer	Customer Focus Information Strategy/Knowledge Management Continuous Improvement.training	1	1	1	Y	N/A	N/A
	Downstream Enterprise Goals							
	Deliver the Architecture to the Enterprise	Sychronization Communication Task Management.prioritization Continuous Improvement.training	0	0	1	N	Y	N
	Downstream Product Goals							
	Deliver the Architecture to the Architect	Information Strategy Communication Empowerment Task Management.prioritization Design Alternatives Information Strategy/Knowledge Management Continuous Improvement.training	1	1	1	Y	N/A	N/A
	Direct Feedback on User-Interfaces	Communication Task Management.prioritization Customer Focus Information Strategy/Knowledge Management	0	0	1	N	Y	N
	Deliver Ultraquality	Communication Customer Focus Process Fluidity.Concurrent Engineering Information Strategy/Knowledge Management	1	1	1	Y	N/A	N/A

Table C.3 Mid- and Downstream Objectives/Lean PD Behavior Comparison