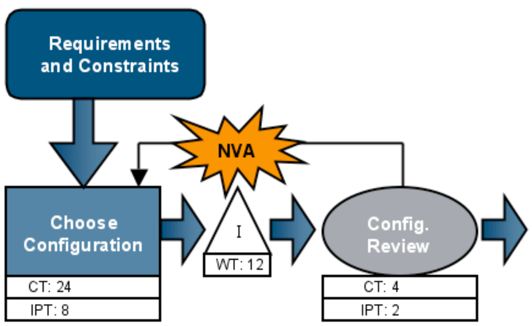
Product Development Value Stream Mapping (PDVSM) Manual



Release 1.0

September 2005

Hugh L. McManus, PhD

LEAN AEROSPACE INITIATIVE

The Lean Aerospace Initiative
Sept. 2005

Product Development Value Stream Mapping (PDVSM) Manual 1.0

Prepared by:

Dr. Hugh L. McManus
Metis Design
222 Third St., Cambridge MA 02142
hmcmanus@alum.mit.edu

for the Lean Aerospace Initiative
Center for Technology, Policy, and Industrial Development
Massachusetts Institute of Technology
77 Massachusetts Avenue • Room 41-205
Cambridge, MA 02139

The author acknowledges the financial support for this research made available by the Lean Aerospace Initiative (LAI) at MIT, sponsored jointly by the US Air Force and a consortium of aerospace companies. All facts, statements, opinions, and conclusions expressed herein are solely those of the author and do not in any way reflect those of the LAI, the US Air Force, the sponsoring companies and organizations (individually or as a group), or MIT. The latter are absolved from any remaining errors or shortcomings for which the author takes full responsibility.

This document is formatted for double-sided printing and edge binding. Blank pages are inserted for this purpose. Color printing is preferred, but black-and-white printing should still result in a usable document.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

This manual is dedicated to the memory of Joyce Warmkessel, a colleague, mentor, and friend to many in the LAI Product Development and Testing and Space Operations communities.

The ideas in this work were developed by the LAI product development community. The primary compiler and codifier of these ideas was Lt. Rich Millard, in his Master's thesis entitled "Value Stream Analysis and Mapping for Product Development."[1] Other major contributors included LAI students Tyson Browning,[2] Jim Chase,[3] Robert Slack,[4] and Joshua Bernstein[5] and their advisors and committees. Many other LAI students and staff also contributed as well. LAI product development research work has been based on, and grounded by, the activities of a very active group of industry practitioners, too numerous to name, who have selflessly shared their hard-won practical knowledge.

The text of this manual is built on LAI research and member documents. Much of its contents are excerpts, modifications, or paraphrases of existing work. Every effort has been made to correctly attribute all contributions. Word-for-word excerpts are identified with quotes or indented, with citations. Many other excerpts have been edited to varying degrees and are integrated into the text for clarity. Their sources are cited in the text or in endnotes. Any omissions or errors of attribution should be brought to the author's immediate attention for correction.

ABOUT THIS VERSION

This version is similar in form and content to the PDVSM Beta release of April 2004. The entire text is revised, and in many places improved, but users of the Beta version will see few surprises. This version does incorporate improvements and additional material in key areas such as metrics and suggestions for improving the value stream. The major difference, however, is the higher degree of confidence with which we can make recommendations. This is based on a higher level of maturity of our knowledge of PDVSM, gained through both extensive practical experience and further research. The broad conclusion drawn from the additional studies is that the basic method presented here is effective. The details of these additional studies will not be incorporated here, but this version does incorporate many new references and supporting footnotes, including references to the latest LAI research.

1.0 INTRODUCTION AND SCOPE

This manual is intended for product development (PD) personnel working on improving their own processes, and the lean change agents working with them. Its aim is to provide practical guidance for applying lean concepts to PD process improvement—specifically, PD Value Stream Mapping (PDVSM). Although sources will be cited, and further reading suggested, this manual and some basic background in lean should be all that is required to start improving product development processes.

This manual is focused on PD process improvement. It is based directly on Lean Aerospace Initiative (LAI) research and member experience. The content is well grounded in research, and (where it exists) theory, and extensive endnotes and references are included. However, it is *not* an academic monograph. It is an attempt to capture the knowledge and best practices of the LAI PD community to date. PD process improvement is an active and rapidly advancing area of practice, and it is expected that periodic updates to this manual will be required. The LAI PD community would be delighted to know if your practices reinforce, contradict, or have moved beyond the content of this manual.

Scope

The scope of this manual is limited to the mapping and improvement of a single, definable product development process, as shown in Figure 1-1. It will show you how to map this process "door to door." It will then help you find the wastes, inefficiencies, and non-value-added actions that can be eliminated quickly to achieve a "future state," and give you insight into the possible "ideal state" that should serve as a longer-term goal.

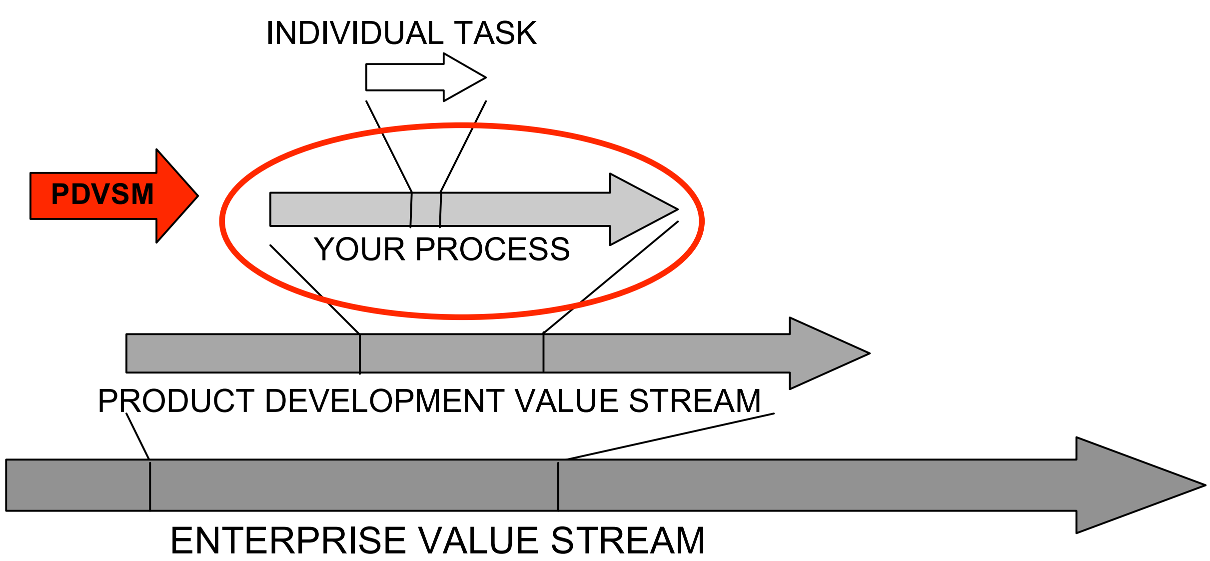


Figure 1-1. PDVSM applies to a definable process within the product development value stream

The manual will concentrate on processes on the scale of component-level hardware development. The wording of the text and examples is hardware-centric, but the intent is to be useful for any development process with a similar scale. The method should apply to many other off-the-factory-floor processes, including but not limited to software development, service or maintenance processes, information-handling "office" processes, and decision-making processes such as requirements generation. Some translation of terms and recasting of examples will be necessary in these contexts, however. Some of the wording is also cast in terms of new product development. This does *not* imply a bias towards new development; on the contrary, much of the member experience in PDVSM is in program-support areas such as drawing changes initiated on the factory floor, or modifications of existing products.

Using this Manual

This manual is not intended to instruct you in basic lean principles—Womack and Jones' *Lean Thinking*[6] is the seminal introductory work, and basic familiarity with the ideas therein is assumed. The basic concept of Value Stream Mapping (VSM) is introduced in *Lean Thinking*, with examples based on the production of physical products. VSM as practiced in lean manufacturing, is expanded in detail in Rother and Shook's *Learning to See.*[7] Familiarity with this work is highly recommended. Further preparation in VSM,[8] process mapping for business process improvement (available from many sources, the work of Trischler[9] is particularly relevant) and Design Structure Matrix (DSM) techniques[10] would be valuable for lean change agents.

This manual begins with a basic primer on the application of lean to product development processes. It is intended for general background for those unaccustomed to applying lean ideas to product development. The next chapter covers getting started—forming a team, defining the boundaries and interfaces of the process under study, and defining the value you want the process to create. Impatient readers may wish to skip to Chapter 4, which covers mapping the tasks and information flows of the process, collecting process data, and evaluating value creation (or lack of it). Chapter 5 introduces the info-wastes, the wastes of information and time that are the prime causes of inefficiency in product development processes. Chapter 6 includes member experience and best practices for eliminating these wastes and improving the process to achieve a future state. Chapter 7 covers ideas for further improvement, including LAI research-derived improvement heuristics and reference to other sources of ideas for radical improvement. The manual concludes with an exhortation to strive for perfection through continuous improvement. Throughout, a running example clarifies the ideas and shows how they might apply to a real process. Appendices include some supporting research work, a sample data collection form, and an additional example which shows how the technique was applied during an actual *Lean Now!* improvement event.

This work stays focused on the improvement of a single value stream, but is intended to work with a number of other LAI and related products. A major product of an LAI-related Lean Forum project is Aerojet's *Learning to Develop*.[11] It is a broader and more general product than this manual. It is highly synergistic with this manual if you need guidance a higher level and/or if you need and introduction to improvement tools other than value stream mapping. The transformation of entire enterprises is the theme of the Lean Aerospace Institute's *Lean*

Enterprise Value book.[12] *Lean Enterprise Value* also provides some important understanding of the need not only to do the job right, but to do the "right job." Implementation of enterprise transformation is covered in LAI's Enterprise Transition-To-Lean (TTL)[13] guides, and in the Enterprise Value Stream Mapping and Analysis guide (EVSMA).[14] Transforming of product development organizations (as opposed to processes—the subject of this work) is covered in LAI's Product Development Transition To Lean guide (PDTTL).[15] Lean change agents would profit from the understanding of the bigger picture that studying and using these tools would provide.

Finally, this manual is best used in the context of a lean improvement event. Chapters 3 through 6 can be used as instructions for a simple "stand alone" PD value stream improvement and mapping event. Better is to embed PDVSM in an existing context. The method has been used in member company *kaizen* events, *Lean Now!* events, and as part of Lean Enterprise Value (LEV) training seminars. Each of these has included a mechanism for selecting the appropriate value stream to map. They have also included explicit support from upper management, trained facilitators, training material, and a structure that allows time for collecting data, creating the value stream map, analyzing it, and planning future improvements or future events based on it. In several of these contexts, PDVSM has been taught with the aid of the LEV training simulation's product development module.[16]

These contexts are different enough (and often location- or company-specific enough) that we will not try to include detailed instructions to replicate them. Instead, we urge the reader to adapt the PDVSM Manual as a tool that will be useful in improvement events in his or her environment.

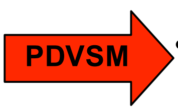
2.0 LEAN ENGINEERING PROCESS IMPROVEMENT

Application of lean techniques to product development processes is underway across the US aerospace industry. The techniques for doing this are not well established, and many practitioners are essentially feeling their way forward, learning by doing, and seeing what works and what doesn't. Over the last five years, the product development community within the Lean Aerospace Initiative has shared the insights gained in this collective effort, and reached a rough consensus on what lean means in the product development context. The term used for these efforts by most industry members, and hence adopted here, is Lean Engineering.[17]

Focus on Engineering Process

Lean Engineering has three goals, representing three very different areas of process improvement. They are:

- Creating the right products—Creating product architectures, families, and designs that increase value for all enterprise stakeholders.
- Effective lifecycle and enterprise integration—Using lean engineering to create value throughout the product lifecycle and the enterprise.



- Efficient engineering processes—Applying lean thinking to eliminate wastes and improve cycle time and quality in engineering.

Product Development Value Stream Mapping is a tool for the last goal—achieving an efficient process. It can be an important enabler for the other goals, but the sole focus of this work will be on process improvement. It is where most lean implementers need to get started on their journey to Lean Engineering. The emphasis will be on the "heavy lifting" of engineering work. Figure 2-1 shows a conceptual view of the entire product development process, adapted from Ulrich and Eppinger.[18]

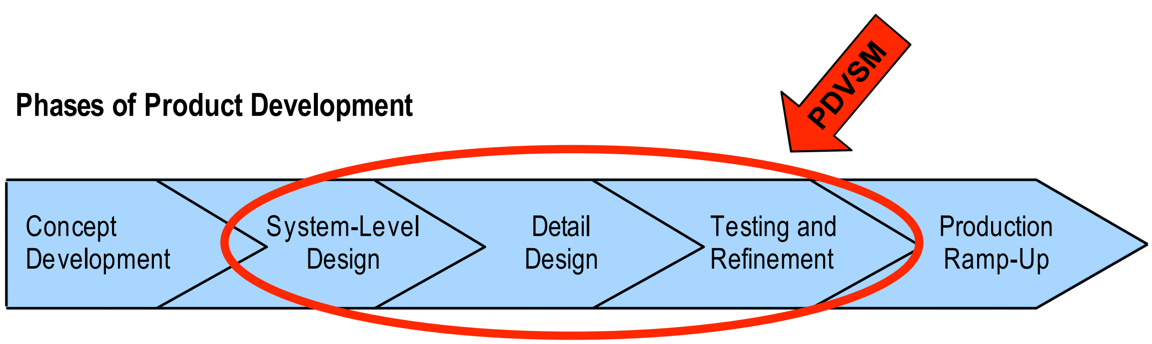


Figure 2-1. PDVSM focuses on core engineering processes

The focus here is on the processes involved in the core phases, such as preliminary and detailed design, analysis, process design, review, validation, and verification. The "front end," where user needs are collected, concepts brainstormed, and requirements set, has its own problems and may well benefit from lean analysis.[19] The transition to production is also very important, and is a key area where enterprise integration is required. These phases will certainly benefit from lean processes, and may well be targets for value stream improvements, but we will not address the special circumstances and needs of these phases.

Figure 2-1 is drawn for new product development. Much aerospace engineering work is currently concerned with modifications or support, and it should NOT be concluded that this sort of work is excluded. On the contrary, many of the examples we will use are taken from support processes such as drawing changes initiated on the factory floor. The emphasis will, however, be on the *engineering process* associated with this work.

Engineering Processes Need Improvement

LAI research and the reports of member companies paint a clear picture of current practice in engineering process. A formal process is required for almost all aerospace engineering activities, to satisfy quality, safety, and regulatory concerns, and to allow management of the complexity of aerospace systems. However, these processes are often poorly defined. Existing process definition can refer to obsolete practices (e.g., paper drawing when CAD is used), contain detail that is not relevant to most jobs, or miss key practices (e.g., appropriate ways to handle new materials or technologies). They may also capture practices that once were critically important, but have become irrelevant over time—in lean terms, they may be *monuments*.[20] Partly as a result, they are often followed only loosely. Quality work is nevertheless accomplished, thanks to the professionalism of the engineers and managers involved, and conservative review and verification practices. This situation has, however, a high potential for process inefficiency.[*]

Not surprisingly, this inefficiency is observed. Figure 2-2 shows typical survey results.[21] Engineers were asked to assess how much of their effort (in time-card hours) was spent adding value directly to the tasks at hand, how much time was spent in necessary support tasks, such as set up, and how much of their time was wasted. Forty percent of their effort was described as pure waste, and only thirty percent value added. Limited formal study[22] data confirms that 30%–40% of engineering effort is typically wasted. Much worse, from a lean point of view, is the data collected on work packages shown in Figure 2-3. When engineering work packages (or "jobs") were tracked (as opposed to engineers), they were found to be inactive 60% of the time![23] No effort was being expended on them and hence, by some measures, they were not costing anything. But assuming the work packages have some value and level of urgency, it is clearly a waste that their cycle time is more than doubled simply because they spend a lot of time sitting on desks and not being worked on. In a recent LAI study, Kato notes this information inventory has "striking similarities [to] manufacturing processes a few decades ago." While the engineers are busy all the time, the valuable information is idle and becoming, in Kato's words, "rotten" as it becomes obsolete with time.[24]

---

[*] LAI members have lately done extensive work on process definition, including in PD. Typically, these definitions are lists of things to do, sometimes but not always including inputs and outputs. They are an excellent starting point for a PDVSM exercise, but do not equate a list of tasks with the value-creating flow of the product(s) through them!

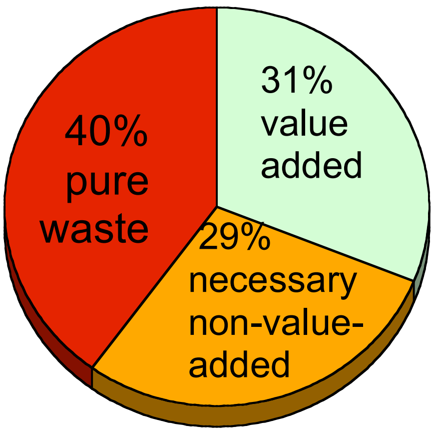


Figure 2-2. Value assessment of aerospace engineering *activity* (as % of charged hours)

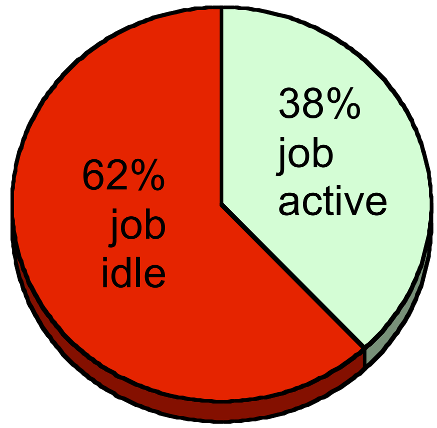


Figure 2-3. Assessment of activity on aerospace engineering work packages (as % of actual hours)

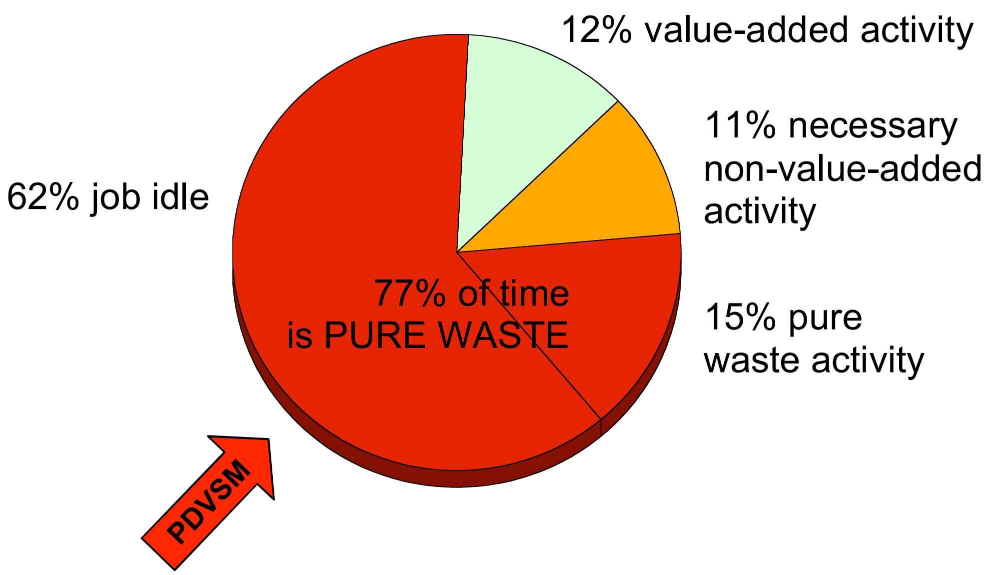


Figure 2-4. *Notional* combined value assessment of a typical aerospace engineering job (% time)

Figure 2-4 combines these data, yielding an alarming picture.[25] A typical work package is undergoing value-adding activity only 12 percent of the time! Conversely, 77 percent of the time it is in a state of pure waste—either idle or undergoing useless processing. PDVSM is aimed at eliminating this waste.

This alarming picture is difficult to accept at first, but is confirmed by multiple independent sources. Anecdotally, but very consistently, LAI member *kaizen* process improvement events reveal 75%–90% job idle times in the "bottleneck" processes selected for improvement. Another interesting verification comes from a process simulation training game run in LAI seminars.[16] The simulation includes a simple model of a typical batch-and-queue engineering process, with unsynchronized functional tasks and "mixed model" jobs (some easy, some hard). The simulation consistently yields more than 50 percent job idle time *due solely to queuing effects* if allowed to run without active management of the workflow. The cumulative picture from these sources is that one can *expect* to find 60%–90% time waste in a typical engineering process, even if all of the steps are value added. This level of waste exists even if all of the personnel involved are busy all of the time, and the majority (60% in Figure 2-2) of their time is spent on value added or at least enabling activities.

The final proof of the potential for process improvement is the results of process improvement efforts. Typically, *kaizen* events that identify 75 percent task idle times propose eliminating that idle time, resulting in a quick 75 percent reduction in cycle time on paper. Actually locking in this improvement is not so simple, but several member efforts have achieved such improvements in practice, sustained them over time, and documented them well enough to be convincing.[26] Figure 2-5 shows "before" and "after" data on a drawing release process.[27] The process was rearranged and coordinated so that single-piece flow could be achieved. This allowed elimination of almost all waiting and delays. Although required signatures were reduced by 63 percent, no significant engineering or review/verification steps were removed. The cycle time results speak for themselves. In addition to a dramatic decrease in the mean cycle time, the standard deviation of the cycle time is reduced even further, which allows confident planning and scheduling. In this case, the savings in charged engineering hours were small; typical results for value-stream oriented improvements are in the 0%–25% range.[28]

What is not evident in the figure is the reduction in defects requiring rework from 66 percent of released drawings to less than 3 percent. This replicates the experience in the factory—that lean processes reduce cycle times without sacrificing quality. Indeed, on the contrary, the control and discipline introduced into the process improves quality without the need for checking, reviews, or other "quality" activities. As Womack observed in the case of lean automobile assembly, the traditional linkage between high quality and low efficiency is broken by lean—the quality is "free."[29]

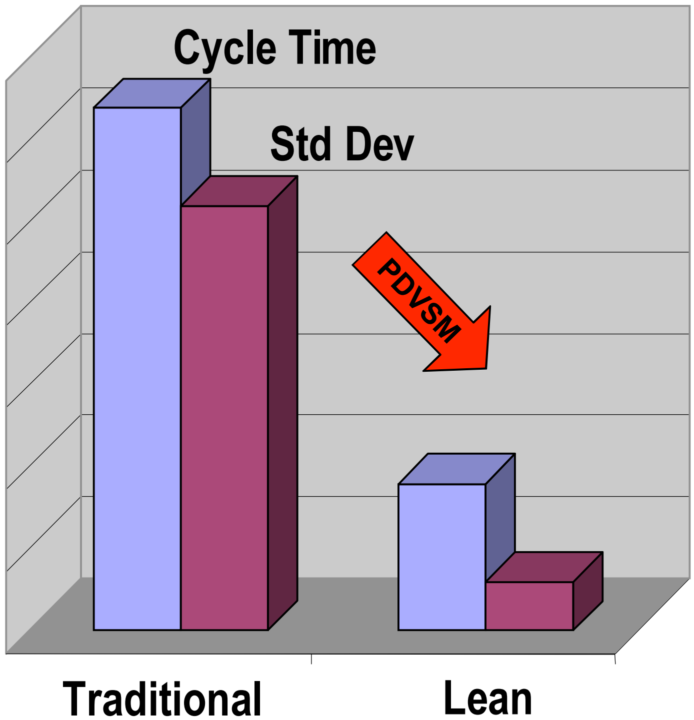


Figure 2-5. Sustained improvements in cycle time possible

Engineering is Different

The first objection to the application of lean techniques developed in the factory to engineering processes is that "engineering is different." This is true. It is *not*, however, a reason to reject lean as a method for engineering process improvement.

Engineering processes differ in fundamental ways from factory processes. Most of the differences are driven by the fundamental *uncertainty* of product development processes—at the beginning of the process, the exact content of the output is not known. This is in stark contrast to factory operations, where the ideal is to make a part precisely the same as the last one. The product development process is also acting upon *information* more than physical material—the ultimate output is the specification of a product rather than the product itself. Finally, most product development processes are acting on a mix of jobs, of greater or lesser difficulty or complication. This is not a fundamental difference; it is analogous to a factory working on *mixed-model production*. It does, however, complicate the application of process improvements. The similarities are driven by another fundamental—although the *outcome* may be uncertain at the beginning of a product development job, the *process* should be repeatable.

Now let us consider Womack and Jones' 5 steps to lean:

- Precisely specify *value* by specific product
- Identify the *value stream* for each product
- Make value *flow* without interruptions
- Let the customer *pull* value from the producer
- Pursue *perfection*

We find we must re-imagine how the concepts of value, value stream, flow, and pull apply. This re-imagining is experience based; a summary of LAI member experience is given in Table 2-1.Value, especially as the process is underway, is harder to see, and the definition of value-added is more complex. The value stream consists of information and knowledge, not the easy-to-track material flows of the factory. Due to uncertainties or interdependencies (e.g., between different analytical steps), branching or iterative flows may be beneficial, which is rarely if ever true in the factory. The "pull" to which the system should respond is also rarely a simple customer demand that can be used to calculate a *takt* time (a metronome-like beat that paces the process; at each beat a product is created). Product development operations are usually intermediate steps in an overall enterprise effort to create value.[†] Finally, perfection is even harder to reach, as simply doing the process very fast and perfectly with minimal resource used is NOT the final goal; efficient product development process is simply an enabler of better enterprise performance and better products.

Table 2-1. Applying the five lean steps to Engineering

	Manufacturing	**Engineering**
Value	Visible at each step, defined goal	Harder to see, emergent goals
Value Stream	Parts and material	Information and knowledge
Flow	Iterations are waste	Planned iterations must be efficient
Pull	Driven by takt time	Driven by needs of enterprise
Perfection	Process repeatable without errors	Process enables enterprise improvement

[†] However, the idea of *takt* time *can* be applied in a somewhat modified fashion to PD processes, as we shall see in a section to follow.

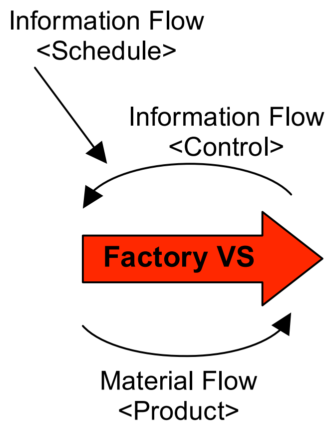


Figure 2-6. Factory flows

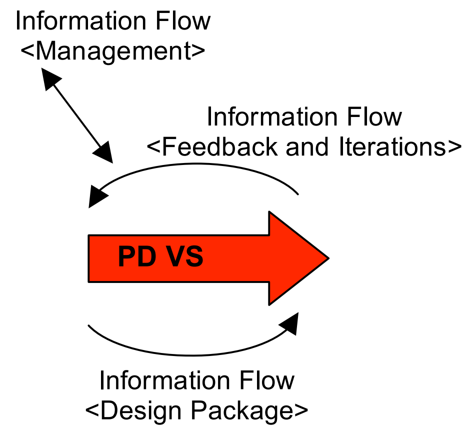


Figure 2-7. Product development flows

We have stated that the product development value stream consists of information and knowledge. Information flows also exist in the factory context. As used in Rother and Shook, for example, the term "information flow" refers to both the flow of control information back up the value stream, and to management or schedule information imposed externally. Figure 2-6 shows the use of this terminology. In product development, all the flows are essentially information flows, so we must be careful to distinguish the different types. In Figure 2-7, the design package or other information "product" flows in the direction of the value stream; feedback, iterations, and perhaps control information flows "upstream," and management information is exchanged with the process. All of these flows need to be mapped. As a default, when we refer to an undifferentiated information flow, we mean the flow "with" the value stream, i.e. the design package in Figure 2-7.[‡]

In one important way, product development and factory value stream mapping are similar. In both cases the leaning out of the local process that value stream mapping enables is *necessary*, but *not sufficient*, to achieving a lean enterprise. In the factory, efficiency in producing a product that the customer wants is clearly desirable. However, at least in aerospace, lean practices *only* in the factory produce only marginal gains—they are *not sufficient*.[30] It is clear that product development has a great deal of leverage on both the creation of the right product, and the enabling of lean production through appropriate design. It is tempting to consider product development process efficiency as a secondary problem. This would be a mistake. Trying to leverage a process that is only 12 percent time-efficient to enable either lean enterprise transformation or the rapid evolution of better products is not going to work. Lean product development processes are *necessary*.

The differences explored in this chapter make direct application of factory-oriented value stream mapping methods to product development practices difficult—hence this document. We will have more to say on these issues as we explore the product development value stream in the succeeding chapters.

‡ Not shown, for clarity, are the "feeder" flows. In the factory, these are parts and materials; in the PD process they are yet another set of information flows.

Summary

Lean engineering requires efficient engineering processes. Without consistent execution of the fundamentals, more ambitious lean improvements are likely to fail. Therefore, improving the value stream is an excellent place to start, and a continuing change enabler. LAI member experience indicates that current practices are moderately inefficient when the metric is wasted effort; they are very inefficient when the metric is wasted time. Much of this waste is time spent in “inventory,” waiting to be picked up from a real or virtual in-box. Improvement efforts aimed at making the value stream flow without waiting and with minimum waste can *typically* expect to identify 50%–75% cycle time improvements, and more modest (0%–25%) savings in engineering effort. Individual results may vary.

Engineering is different from manufacturing. Value may be defined, and will almost certainly be measured, differently. The value stream consists of flows of information and knowledge that are less easily tracked than the material flows of the factory. Uncertainties must be handled and eliminated as part of this process, and hence coupled or iterative processes may be required. Many engineering processes handle “mixed model lines;” they work on jobs of varying size and complexity. These factors complicate the job of value stream mapping.

3.0 GETTING STARTED

Most lean improvement is done as part of an organized site-specific event. Typical is a *kaizen* event: A team of process participants, with experienced facilitators or *sensei*, dedicate a period of time to diagnose and try to correct a specific problem using lean methods. Training in basic lean and the specific tools needed for the job is included. Management support for people's time, access to required data, and permission to change things are granted. Required tools are assembled, and templates for presentations and plans provided. Usually, the effort is pointed at a pre-selected process. The process may be selected based on a known problem, a higher-level analysis such as an Enterprise Value Stream Map, or a previous higher-level iteration using a tool such as this one.

This chapter can be used in a variety of contexts. It can be used stand-alone to set up a mini-*kaizen* to map a product development value stream. In this case, the help of an experienced facilitator is strongly recommended—lean is not a cookbook method, and this chapter is not a cookbook for a *kaizen*! In a pre-planned *kaizen* environment, this chapter should be used as a supplement and check. The pre-planned *kaizen* will have generic rules; this chapter will provide *specific* advice on the needs of a PDVSM effort.

We will assume here that a process has been selected for improvement, and an improvement leader selected. Now, it is time to identify the stakeholders, put together a team, formally define the scope of the process to be tackled, and define the value created by that process.

Identifying Key Stakeholders

Stakeholders are the people who derive value of any sort from a process. Without identifying the stakeholders, we can neither put together the right team, nor define value in a useful way. Stakeholders typically include the participants in the process, the users of the output of the process (sometimes called the internal customers), other indirect users of the output (participants in processes further down the value stream), the customer of the design or other final product development output (who may be internal or external), the external customer for the physical product that will be made from the design, and the end user of that product.[31] Other stakeholders may include suppliers of information or resources to the product development process, suppliers of parts or assemblies to the physical product, the firm or enterprise management or leadership, employees or their representatives, the community in which the process takes place, and the general taxpaying public.

The key stakeholders, and their expectations for the process, its outputs, and the improvement of both, need to be identified. In traditional factory value stream mapping, the only explicit stakeholder of importance is the buyer of the product of the factory (a/k/a "the customer"), and his or her expectations are assumed to be simple—a certain number of units in a given amount of time, on time, and to some standard of quality. Other stakeholders (e.g., owners or shareholders, employees) are implicitly assumed to benefit from the elimination of waste and pursuit of perfection, while implicit demands are placed on a third set (suppliers, workers) to conform to the improved process. In product development processes, it is often the needs of the enterprise that determine pace (e.g., a large-scale development program may need a certain process completed in a given time to stay on schedule), while the downstream processes (or internal customers) define the necessary quality. Explicit expectations may exist for how much waste, cost, or time is eliminated from the process, especially if the process improvement is part of a larger-scale enterprise process improvement. These expectations often come from both the enterprise (which desires cost savings) and the external customer (who desires, or even mandates, price reductions, schedule improvements, or higher quality). It is wise to explicitly include the expectations of, and demands on, process workers, as well as upstream processes that supply or support the process undergoing improvement. Finally, depending on the situation, other stakeholders may have critical needs to be met or contributions to make.

The engineers and other professionals who carry out the process *must* be treated as stakeholders. This is true of any process, including those in the factory, but is particularly important for engineering product development tasks. Most workers have specific skills, personal and professional standards, and ethical responsibilities for the work being done. Some may have professional or even personal liability for it. They are *heavily invested* stakeholders. Fortunately, their interests are generally aligned with those of lean improvements. Doing high quality work, without wasted time or induced errors due to bad policies, inefficient procedures, ineffective information flows, or unneeded reviews, is typically what engineers want to do. This alignment must be communicated and exploited or lean improvements of engineering processes will fail.

Identify the key stakeholders, and, as quantitatively as possible, their expectations for both the process and its improvement. This will be key input into both the definition of the team, and the specification of the value, covered in the next few sections.

Defining the Team[32]

For a simple enough problem, a team of one (you!) may suffice. Typically, however, a variety of perspectives are needed to reveal the often difficult-to-see product development value stream. The team should embody a balance of enterprise perspectives, whether they come from multi-skilled people, or multiple people. These perspectives may include:

> ***Lean Experts and Facilitators:*** for knowledge and experience in Lean theory, as well as the methods and tools used for the process improvement. There are few *sensei* available for product development lean;[33] you will probably have to do your best with combinations of manufacturing lean knowledge and product development improvement expertise.
>
> ***System/Enterprise Thinkers:*** for thinking "outside the box" and considering the needs of the greater enterprise and continuity with upstream and downstream processes.
>
> ***Process Owner(s):*** for knowledge and experience in the process to be improved, and authority to change it. Ideally, the team lead is a lean-savvy process owner.
>
> ***Process Participants:*** for detailed, on-the-ground knowledge of the process, its strengths and weaknesses, and how it is currently executed. They are the best source of both improvement ideas and knowledge of the possible pitfalls of process change.
>
> ***Customer(s) and Supplier(s):*** internal and/or external, for understanding of the value of process outputs; the availability, flexibility, or improvements of process inputs; and the correct diagnosis and improvement of interfaces and handoffs.
>
> ***Other Key Stakeholders:*** Depending on the circumstances, acquirers, end users, representatives of the larger enterprise (e.g., project or company managers), representatives of facilitating organizations (HR, finance, etc.), or representatives of other groups may be key to understanding the value produced by the process, the constraints imposed on it, and the expectations for its improvement.

External consultants may be called upon to fulfill one or more of the team roles. LAI member experience indicates that, in general, external consultants should provide support rather than leadership for the improvement effort.

A final note—make sure you have "top cover." Upper management must be explicitly behind the effort. Ideally, upper managers should be on the team. If this is not possible, clear communication to the team that management is behind the improvement effort and tangible rewards will flow from participation and success is required. The team must own the process, and be empowered to change it with reasonable but minimal management oversight.

Training the Team

The team must be at least minimally trained in Lean and the methods and tools chosen for value stream mapping and analysis. A good start would be some lean fundamentals (e.g., *Lean Thinking*), familiarity with Value Stream concepts (e.g., *Learning to See*), and some translation to product development problems (e.g., this manual). Typical member "best practice" is to kick off an event with a short day (six hours or so) of training based on these materials. Lean training is available from many sources, including emerging materials from LAI.[16,34] Discretion is required, however, in the use of training materials aimed specifically at manufacturing (which most are). Generally, the terms are not translated into the language of product development processes, and some key issues are covered lightly or not at all.

Typically, training beyond basic lean might consist of:

- This manual, assigned as a read-ahead.
- Event kick-off material, featuring knowledge from this manual, local experience and practices, and perhaps a training simulation or exercise such as the Lean Enterprise Value Simulation product development module.[16]
- Introduction to specific tools needed for your event, such as process mapping conventions, capacity or gap analyzers, economic calculators, and/or reporting and planning templates. These tools are site specific.

The purpose of the training is to prepare everyone on the team to be able to contribute constructively, and to enhance communication by establishing a common language and toolset. It is best, therefore, that the whole team train together. Very good results have been recently obtained by the use of customized simulation-based training as a kick-off to the value stream mapping event.[35]

Bounding the Problem

Rother and Shook define a manufacturing value stream bounded by the receiving and shipping "doors" of a single facility—a "door-to-door" value steam. The limits of the product development processes under consideration must be equally clearly defined.

The improvement team must define several critical elements of the value stream.[36] The *process bounds* include the beginning and ending point of the process, and its organizational boundaries. The *product* or products on which the process operates should also be specified. The *owner* will provide the point for direct responsibility for the stream, whether this be a group or an individual. The *output* provides reason for the stream to exist—it is the packaged value generated by the given process. The *customers* then receive the product from the owner at the end of the value stream. These customers do not necessarily represent someone external to the organization; they may include internal customers. The *initial inputs* as well as the additional knowledge and information that may be pulled into the process are the raw materials on which the process operates. The *constraints* place limits of many sorts on the process. Figure 3-1 illustrates the bounds we wish to define in order to apply value stream mapping to the process.

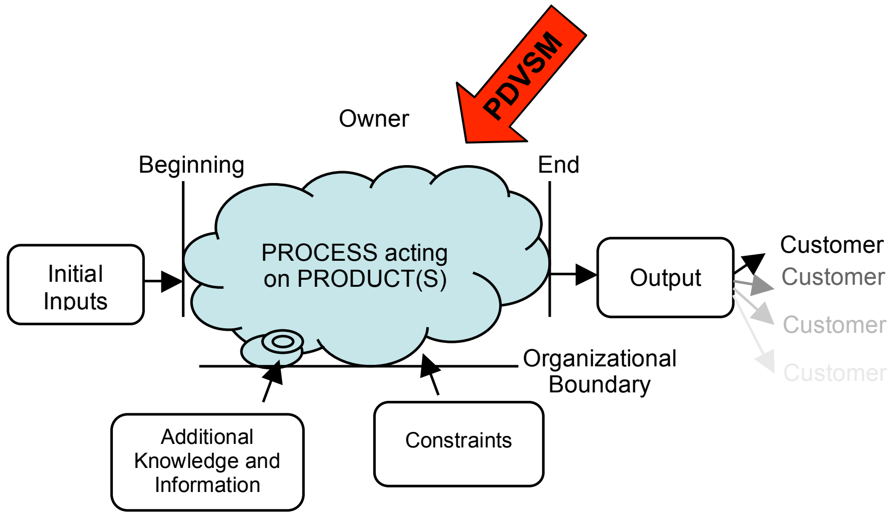


Figure 3-1. Bounding the problem to which PDVSM will be applied

EXAMPLE: Getting Started

To demonstrate this step, and each of the steps that follow, a running example will show the application of the relevant principles. It is a composite of LAI member experiences. As such it will include some missteps and disappointments. Their inclusion is intended to illustrate typical challenge faced by PDVSM teams. This example is based on and uses material from Millard, Chapter 6.[1] It is reinforced by the real value stream mapping done during an alpha test of this product, as recorded in Appendix C, and in several other case studies.

The example is a preliminary design process for a generic aero-structural part such as a fairing or fin. This example has many of the features of product development processes, such interdependent tasks requiring design iteration. The example will be mapped in a simplistic fashion to highlight ideas and principles, rather than to faithfully reproduce a real process.

The key stakeholders for this process are:

- The company and several projects in which the company is involved, which require a substantial improvement in the cycle time of the process to meet accelerated schedules
- The downstream users of the process outputs, which require timely and high-quality work
- The preliminary design team itself. The technical knowledge and experience of the team is a core competency of the company, and its use must be maximized while satisfying the personal and professional goals of the team members
- The suppliers of information to the process

The PDVSM team mirrors the above stakeholder list, with representatives from the preliminary design process itself, key upstream processes that provide information to the team, several downstream users of the team's output, and representatives from both the company management and government program management, all joined by several experienced facilitators. The team is weighted towards the preliminary design process workers, with both supervisors who have a fairly broad picture of the process, and workers with technical depth in all of its major parts, represented.

For the example, the critical process elements are:

- Process: preliminary mechanical design
- Product: fins and fairings
- Bounds: requirements definition to delivery of preliminary design report
- Owner: project managing engineer
- Output: design report and presentation
- Customers: PDR review team, proposal team, detailed design team
- Inputs: system requirements (drag, weight, cost, etc.)
- Constraints: interface geometry, manufacturing standards and capabilities, company standards
- Additional inputs (currently known): Flight envelope, vehicle aerodynamic analysis, material properties

These bounds are shown graphically in Figure 3-2 below.

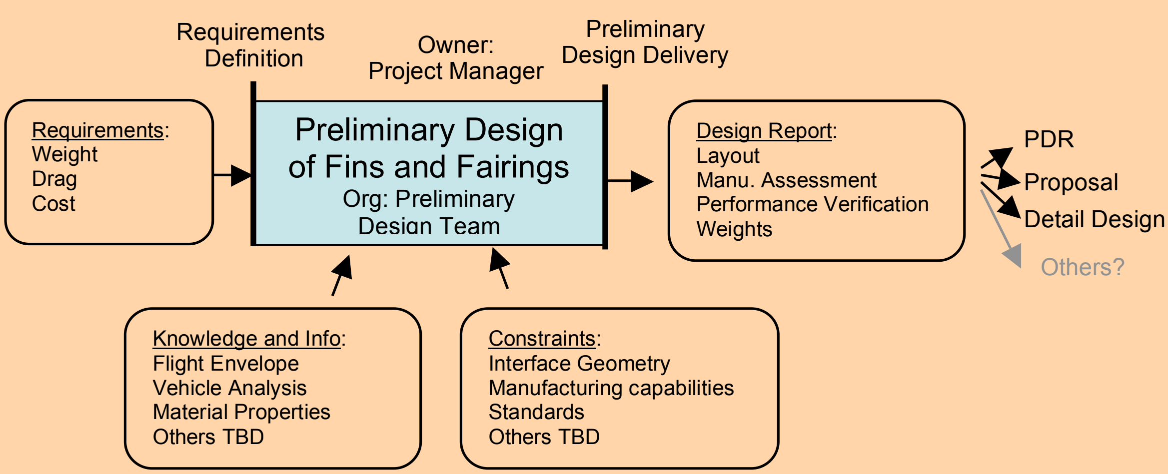


Figure 3-2. Bounded example problem

Defining the Value

Defining the value of product development work in a general way is an ongoing, and probably never-ending, task.[37,38,39] It is important to strike a balance here. Without a working definition of the value created by the process we are mapping, and an appreciation for how that value is created, you cannot guide our improvement effort. However, it is important not to get wrapped up in an essentially philosophical debate over the details of value.[40]

First, assume the overall process you are analyzing is, in fact, value added. If it is not, you are clearly wasting your time. *The aim is to do it efficiently and well, so that the larger enterprise can get on with the task of providing value to the stakeholders*.[41] To do this, value must be understood in two rather different contexts—the value of the process output to the larger enterprise, and the creation of that value during the carrying out of the individual tasks that make up the process.

What is the value of the execution of the process to the key stakeholders? An important part of this value is the value of the process output to its downstream customers, internal or external. It is also usually necessary to consider the value to the larger enterprise of the completion of the entire process. Figure 3-3 illustrates these value questions.

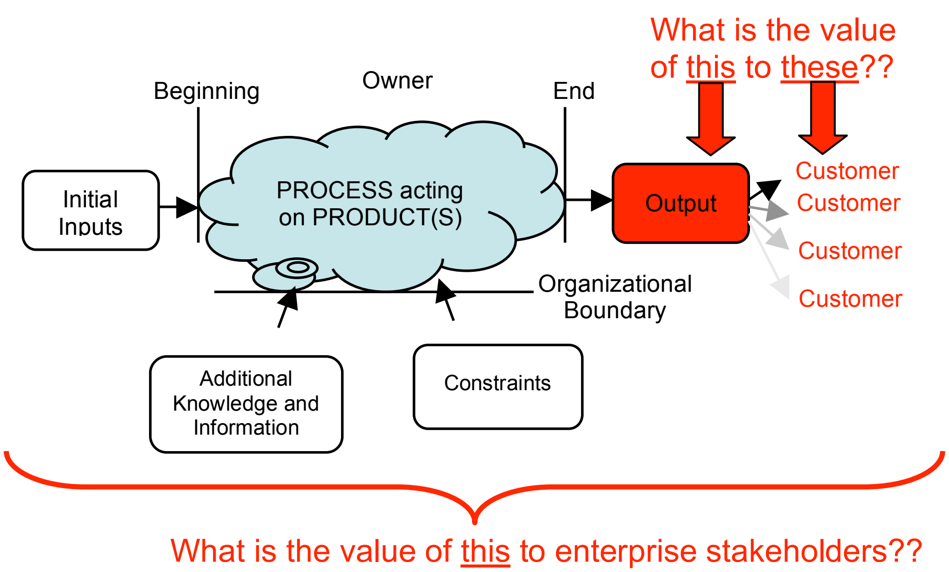


Figure 3-3. Process value questions

If the process you are analyzing is necessary for the larger enterprise to create value, the goal must be to do it well, efficiently, and in a timely fashion. This goal should be made as explicit as possible. A template for such an explicit goal is below.

Produce the *required outputs*, without defects, as efficiently as possible, and *at the right time*.

The words in italics above should be replaced by words specific to your situation.

The *required outputs* are going to be very specific to your situation, but should be checked to assure that they:

- Embody end products that meet performance requirements
- Guide downstream processes
- Assure and document an appropriate level of certainty and/or risk
- Meet constraints

In our example (Figure 3-2) the output is valuable if the product specifications and weight meet requirements, the manufacturing assessment is at least positive (and preferably provides guidance to eventual manufacturing process design), the verification is carried out to a level of uncertainty appropriate for preliminary design, and constraints are met. Note that *unneeded* outputs are waste. If an output is not "pulled" by the larger enterprise, it should not be produced! If outputs do not fall into one of the above categories, they may not be needed. Often archival documentation is required by regulation or to meet the needs of future work. Such requirements should be questioned, however, to assure that the need is truly there and not simply a matter of habit or tradition.[42]

The phrase *at the right time* may, especially in bottleneck processes, mean "as fast as possible." This is not always the case, however. In a perfectly lean process, outputs would be handed off to downstream processes the moment they were needed. Outputs created before they are needed will sit in "intellectual inventory," with some risk that they will become obsolete before they are used.

Note that the phrases "without defects" and "as efficiently as possible" are not italicized. Defective products will always result in rework for downstream processes; inefficient processes will always imply waste. These are not compatible with the goal of a lean process. Defects need to be interpreted in the correct context, however. Defective PD products have mistakes or missed steps in them that cause *unnecessary* rework. A lack of defects in, say, a preliminary design does *not* imply that the design will not need to be revisited and even changed in detailed design. It does imply these changes will be due to the increased knowledge that comes with design maturation, not due to avoidable mistakes in the preliminary design process.

METRICS: Output Value

Measuring progress towards this goal requires metrics. Some metrics to help you track the creation of output value include:

- Cycle time (how long it takes to do an instance of the process)
- Lead time (how long from the time a need exists until the output is delivered)
- Fixed or non-recurring costs (what resources must exist for the process to take place, even if they are not used continuously)
- Cost per job or recurring costs (what resources are expended to do a job)
- Variation in any of the above (how predictable is the process)
- Rework rate (incident of defects in the process output)
- Customer satisfaction (measured many ways, e.g., requirements checklist, survey)

Metrics should be selected based on a combination of usefulness and the availability of data. We will be revisiting the metrics issue in more detail in Chapter 4.

EXAMPLE: Value and Metrics

Our example problem is a bottleneck process in creating proposals and, later, designs for add-on features to existing aircraft. The process currently takes ten weeks, plus or minus four weeks; the enterprise need is for a solid improvement in mean time (to four weeks) and predictability. It also suffers from quality problems, and produces an output that is difficult for downstream processes to use. Therefore, there is motivation to reduce cost and schedule, and address several known formatting and rework problems. The output value that we want to achieve is therefore something like:

Produce the *preliminary design, including all information required by downstream processes in a format usable by them*, without defects, as efficiently as possible, and *in four weeks every time*.

The primary metric selected is process cycle time. This metric is available for previous jobs, and is simple to both estimate (when imagining process improvements) and track. Engineering hours per job and rework rate are selected as secondary metrics, with some issues as to their use to be addressed. The former is desired as a measure of efficiency, but the accounting system is not set up to capture this information explicitly. Therefore it is not available for past jobs; it may be possible to estimate this as part of the mapping process, but the accuracy will be questionable. Some data is available on rework rate (the number of output items such as drawings returned due to errors or ambiguities) and this data will be tracked after the improvement effort, but it is spotty and of questionable statistical significance. Finally, to capture the satisfaction of downstream processes with the output content and format, a semi-quantitative measure of downstream processes satisfaction, based on interviews with managers and engineers in the downstream processes, will be used.

Understanding Value Creation

Assuming we have a goal defined, we need a basis for evaluation how tasks within our process contribute to that goal. The question is expressed graphically in Figure 3-4, which shows the process under consideration decomposed into a series of individual tasks.

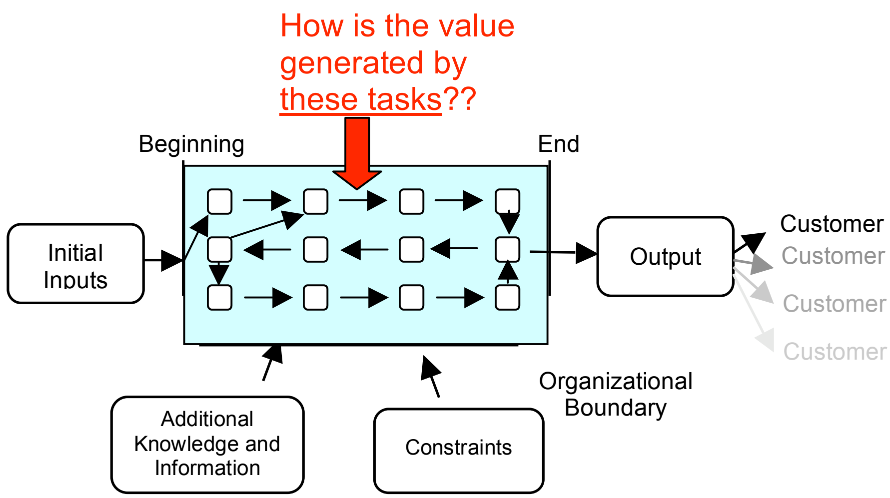


Figure 3-4. Value creation question

In manufacturing, a simple judgment is made as to whether an action contributes to the form, fit, or function of the material flow—if so, the action is judged to be value-added, and if not, waste of either a necessary or unnecessary sort. The form-fit-function metaphor is difficult to map onto product development processes for two reasons:

- The flow in product development is difficult to see, as it consists not of material but of *information*, and
- A key aspect of this information is its *uncertainty*, which implies risk that the product will not ultimately meet user needs.

Browning,[43] elaborating on the ideas of Reinertsen [44] put the resulting difficulty rather well in a design example:

> Designers may start with one design, find it deficient in several ways, and then change it. Especially with novel products, designers learn much along the way about what will and will not work. The desire is to create useful information, which is acted upon by a number of activities and disciplines. The information is valuable if it decreases the risk that the product will be something other than what it is supposed to be—i.e., if it improves confidence in the recipe. Trying, analyzing, evaluating, testing, experimenting, demonstrating, and validating create valuable information.

Chase[45] has elaborated on and generalized this idea, providing us with the conceptual picture in Figure 3-5. As a product development process is performed, each task increases the information that will ultimately contribute to the output package, while hopefully reducing the uncertainty of this information. Value is created continuously, but it is not realized until the process outcome is created. Although value is accumulating, it is very difficult to measure at intermediate steps. Even metrics as simple as percent completion of process have been shown in practice to be difficult to use due to human errors in estimating this percentage.[46]

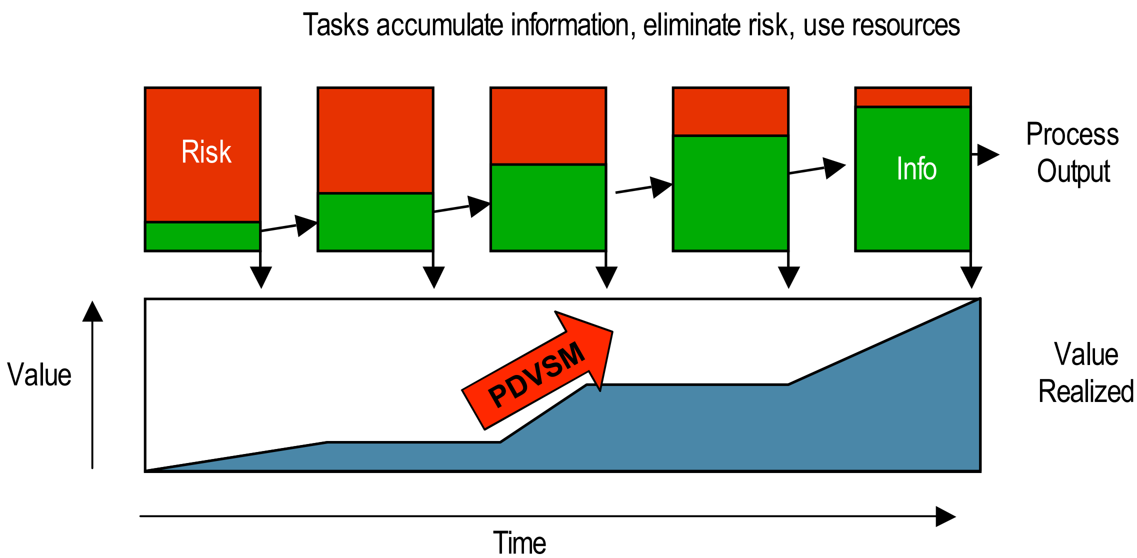


Figure 3-5. Accumulation of value as information is collected and uncertainty reduced

METRICS: Value Creation

Quantitative measurement of the *value* of *individual tasks* has proven elusive. Complex earned value management (EVM) or process tracking software systems have come the closest to enabling this capability, but only at a high cost, especially if starting from scratch.[11] They also provide only a very high-level picture of a complicated value-creation system; see the recent work of Witaker for a complete discussion of the relation between value stream mapping and EVM systems.[47] Simpler metrics such as percent task completion have been shown to be very inaccurate.[46] For many improvement efforts, having criteria to assess value qualitatively is the cost effective solution at the task level. This does NOT mean going without data; task *performance* metrics are discussed further in Chapter 4.

Aspects of Value

Rather than rating tasks in the process value-added or non-value-added, Chase proposed a list of *aspects* of value that a task could contribute. This is presented as Table 3-1. The entire table is presented, as your particular process may require consideration of some or all of the aspects of value presented here (or even definitions of new ones under the "other" category).

Table 3-1. Aspects of Value in Product Development Tasks[48]

Task Contributes to…
V1. Definition of End Product with desired Functional Performance
The task affects the definition and/or functionality of the end product delivered to the customer. It contributes directly to either the function or the form that affects the function. For example, requirements specification, design decisions, material/part/subsystem specification, geometry specification, etc.
V2. Definition of Processes to Deliver Product
The task directly affects the processes necessary to deliver the end product to the customer. It includes the design or procurement of the tools and processes necessary for manufacturing, testing, certification and/or other downstream processes, such as the creation of manufacturing and assembly procedures.
V3. Reduction of Risks and Uncertainties
The task contributes to eliminating uncertainties in performance, cost, and/or schedule. Typically, tasks include the analysis, prototyping, and testing of the product; the testing of tools/production processes, risk analysis, and cost/schedule management.
V4. Forming Final Output
The task directly contributes to the final documentation given to the customer or manufacturer. This typically includes the documentation of the materials, parts, subsystems, and systems, and documentation to meet legal and contractual constraints.
V5. Facilitating Communication
The task aids necessary communication. Typically includes reviews, meetings, and discussions with other company or industry personnel.
V6. Enabling Other Tasks
The task is necessary for other tasks to proceed, although it does not directly contribute to the design, production, or testing of the product
V7. Meeting or Reducing Cost and/or Schedule
The task emphasizes maintaining or improving cost and/or schedule, e.g., many management and process improvement tasks.
V8. Learning or Resource Improvement
The task contributes to the skill base necessary to do future work. This definition includes developing greater knowledge, improving tools or processes, creating new technologies, and communicating this knowledge throughout the team.
V9. Enhancing Employee Job Satisfaction
The task is a positive experience that increases the desire of the employee to do similar tasks; it enhances the professional development or skill base of the employee.
V10. Other
The task performs a necessary or valuable function not covered in the above categories. Examples may include contributions to work environment, environmental impact reduction, satisfying of regulatory or contractual requirements, the following of mandated processes, or the satisfaction other constraints.

Many processes can make do with the reduced set identified by color (or shade) in the table. Creation of the end product, including necessary process definitions, and the reduction of the uncertainty in the end process are highlighted in light green on the table (V1–V3). These are the closest analogy to "direct value added" on the table and should be regarded as such for most processes.

The interpretation of the medium blue value aspects (V4, V6 and V10) is trickier. It is tempting to view these as "necessary non-value-added," and such an interpretation may be correct under some circumstances. However, opportunities exist for direct value addition in these categories. As an example, output format incompatibilities can be a major source of waste.[49] Putting information into a form directly useful to downstream processes (or, better, creating it in such a form) moves the information closer to the customer, and hence can be considered a value-added step. A recent example is creating numerically-controlled milling machine programs direct from 3-D CAD data, instead of creating drawings and then programming by hand. Many kinds of formatting are, however, pure waste. In a similar spirit, other forms of enabling of downstream tasks may be regarded as direct value added if the effort eventually saved as a result exceeds the investment. Tasks in the "Other" category, such as meeting regulatory and contractual requirements, are not usually negotiable. Nevertheless, tasks in this category deserve a closer look as to exactly what value they are adding to the process.

Finally, the dark gray aspects (V5, V7–V9) will fall into a necessary non-value-added (at best) category for a typical process at the level we are considering. Communication is of course necessary, but in a truly lean organization it is seamless; training, knowledge capture, and employee satisfaction building are all necessary, but Womack suggests that these activities be separated from the product value stream although closely coordinated with its needs.[50] Continuous improvement is a value call; one can put it on the direct product value stream, or (like Womack) assume it is pervasive in the lean organization. Special circumstances may exist in your case that require more careful assessment of these value aspects.

Select the key value aspects for your process. Don't worry if this seems difficult now—further clarity will come when we consider waste in Chapter 5. This will also get easier in practice; your team will quickly develop an instinct for knowing value and waste when they see it. Nevertheless, don't get complacent and decide you can just wing value judgments; instead, do your best now, and update the value aspects and definitions as you gain experience.

EXAMPLE: Value Aspects

Our example problem explicitly includes the need for performance specification, manufacturing assessment, and verification. There is also a known problem with handoffs of the information produced. The process (except by its completion) is not a major enabler of other tasks, there are minimal regulatory issues, and training and other issues are not on the direct value stream for this process. Therefore, our initial guess at a value criteria includes V1 through V3, and a qualified V4, from Table 3-1. Tasks are considered valuable to the extent that they contribute to:

V1. Definition of End Product with desired Functional Performance
V2. Definition of Processes to Deliver Product
V3. Reduction of Risks and Uncertainties
V4. Form of Final Output—*to be compatible with downstream processes*

There is a MAJOR caveat that applies to aspects V1 through V3, due to the fact we are doing a preliminary design. All of these goals should be pursued *to the appropriate level of fidelity* for preliminary design, as determined by the customers of the output, and no further.

Summary

A checklist for getting started:

- Have a process chosen for improvement
- Have an improvement process

- Define key stakeholders, and what about the process is valuable to them

- Select and train (as necessary) a mapping team with a variety of perspectives

- Bound the problem:
  Process, product, and process owner
  Start point, end point, and organizational boundary
  Direct inputs, additional inputs needed, constraints
  Outputs and customers

- Decide what aspects of task value creation are key to producing valuable output

Don't get bogged down. Some of these will be estimates or first guesses. Much of this will get easier with experience. Do keep your bounds and value definitions visible, so that the whole team is aware of them; you will be referring to them frequently as the mapping exercise moves forward. Finally, do be flexible to change as things are learned during the mapping exercises. If bounds or value definitions need to change, make sure the whole team knows.

4.0 MAPPING THE CURRENT STATE VALUE STEAM

Now it is time to map out your process. To get your bearings, you may want to quickly draw an overview process map, or even rely on existing process visualizations such as a Gantt chart. But you will have to go deeper to reveal waste and discover opportunities for greater process efficiency and quality. This section breaks down the process of mapping out a value stream into three basic steps: 1) Arranging the process steps (tasks) and information flows 2) collecting performance data on the tasks, and in some cases the information flows as well, 3) evaluating how value is created. We will walk through this process, providing both general guidance and a specific example case. This section ends with some notes on an alternate mapping technique that is complementary to process mapping and particularly useful in cases were coupled or iterative flows exist—the Design Structure Matrix (DSM).

The steps help you to build up your value stream map in a coherent way, but there is nothing sacrosanct about them. In all of these steps (and for any of the mapping methods), the basic advice in the following box applies. For your process, you may find it convenient to, for example, collect basic process data at the same time that you build the tasks and flow mapping, or evaluate value at the same time that you lay out the value stream including the process data. *Do not, however, skip steps entirely. It is important to completely understand the current state before starting into waste hunting or jumping to conclusions about improvements.*

You may even wish to use an entirely different mapping format, or more than one format. We will concentrate here on process mapping, but members have used several other methods. The relevant parts of a report on mapping methods used by LAI companies, collected by Millard[51] and reported by McManus and Millard,[52] is included in Appendix A. Note the conclusion of the report was that capability to do a good PD value stream map correlated with success in improvement efforts. However, overall lean capabilities of the site where the PD value stream was done correlated just as strongly with both good PD value stream maps and success. Our interpretation is that good lean training and application practice enables good PD value stream mapping, and both factors lead to success in improvement efforts.

Basic Advice for Mapping the Current State

Follow the work: Womack advises, to uncover the true value stream, strap yourself to the product.[53] Lacking a physical part to ride on, you must strap yourself to the information—the work package that starts with the process inputs and accumulates and transforms until it becomes the process output. Doing this will require some creativity. If time allows, you may be able to follow a real job through the process. In this case, you must beware of altering the subject of your study by your presence, and also be able to relate your single case (which may be exceptional in some ways) to the general one. Following an abstract "typical" job through the process may be easier, and usually quicker. Millard advises you start at the input boundary and continually ask "what happens next?" On the other hand, Rother and Shook advise starting at the conclusion of the process (the shipping dock) and walking the line backwards. One could imagine continually asking "where did this come from?" as you progress backwards in the information stream. In practice, you may need to trace the flow both ways, especially if you have to trace down complex flows with interdependencies and iterations.

Collect the information yourself: To the extent possible, physically walk the process, talk to the actual participants (ideally, include them on the mapping team), and touch the information actually created. If physical walking is impractical (e.g., some work is done on remote sites), talk to the participants by the highest-bandwidth channel available (e.g., telephone in preference to email) and have them send you real examples of the documents or other information products they produce and handle. Always try to uncover what really happens, rather than the documented version. Use forms to collect information, but as part of a conversation, not by sending them out and expecting to get something useful back (you won't).[54]

Exploit existing process information resources, cautiously: Although you need to personally understand the process, if someone else has collected valid data for you, use it. Examples of data that may exist include "timecard" records, sign-in/sign-out sheets or databases, output reports, test records, etc. Often the data you want is not the primary purpose of the existing documents; for example, the dates on reports (as a record of cycle time) may be more valuable to you than the contents.

Map "in pencil": Although Rother and Shook's 11 x 17 sheets may not be sufficient for your process maps, the spirit of their advice still applies. Do not invest a lot in the format of your maps, make sure they are easy to adapt, and do not spawn multiple versions or revisions. Member responses to this challenge have included use of newsprint rolls, large whiteboards, and walls covered with yellow "stickies"; all low-investment, adaptable, and simple formats.

Map the whole value stream: At least at first, have everyone work on the entire value stream rather than partitioning it right away. The goal is a holistic picture of where the value is coming from, not a mechanistic model of the process; this is best achieved if everyone sees the "big picture."

The Zeroth Map (optional)

Representation of the process with a high-level mapping can help to provide a good initial "big picture" of the process. You may already have a planning Gantt chart available. Another high level mapping technique used by member companies was originated by consultant Alan Ward; it shows information similar to the Gantt chart but with a dimension of level of effort included. If not already available, these maps can be "brainstormed" quickly for many processes. Existing work breakdown plans can also be used at this stage. These high level maps should be used to help visualize the process and prepare for more detailed analysis. It is unlikely at this level that wastes will become apparent.

Note that these maps have some interesting strengths—they are particularly good at showing concurrency of tasks, and in the case of the Ward maps, level of effort, which can be very useful for resource planning. If these considerations are particularly important for your process, you may wish to update these representations as more detailed information becomes available.

EXAMPLE: Zeroth Maps

The Gantt chart found in Figure 4-1 below, and the Ward map found in Figure 4-2, illustrate high-level maps or our example process.[55]

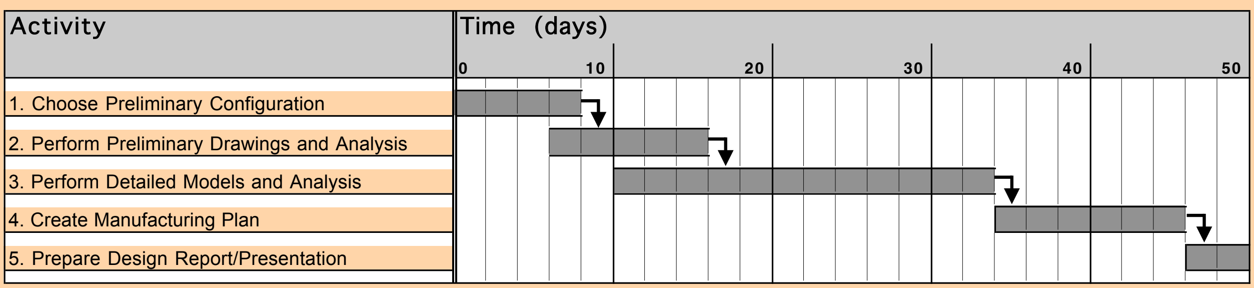


Figure 4-1. Example Gantt chart

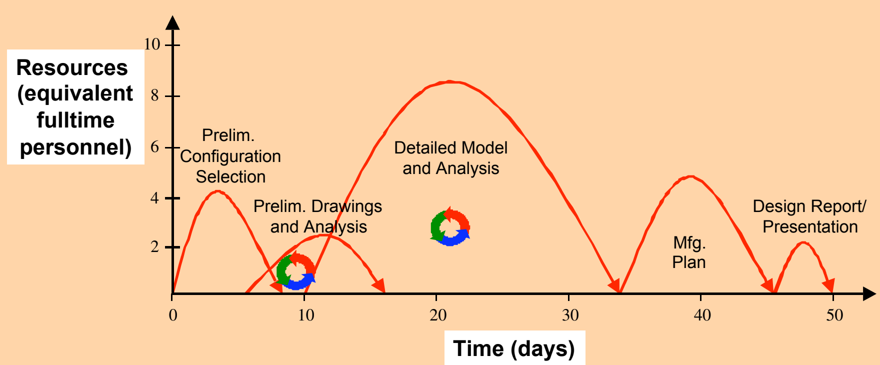


Figure 4-2. Example Ward map

Tasks and Flow

The first step is to identify the tasks that make up the process, and the basic flows of information between them. This map will establish the topology of the value stream: Is it linear? Iterative? Spaghetti-like? The type of map you use, and its symbols and conventions, will depend on the answer to these questions. The next page contains a set of process mapping symbols which should be useful for typical PD processes; they are used in the running example for the rest of this section. Do not be restricted to this set, however. Witaker[47] uses the "swim lane" format, illustrated by an additional example in Appendix C of this document, and several additional mapping symbols, illustrated in Appendix D, to map more complex PD processes. Those looking for further ideas in process mapping symbols should consult Trishler;[9] see Rother and Shook[7] for more symbols for types of flows and lean control concepts.

A decision you will have to make early in this step is how detailed you want your process breakdown to be. Excessively high-level breakdowns (not enough identified tasks) will not provide much insight into the process; excessively detailed ones (too many) will become intractable. Experience indicates that if your breakdown has between 10 and 30 identified tasks it will be both tractable and of sufficient depth to provide useful insight. More tasks may be appropriate if your intention is to build a complex simulation or build a model for long-term study; less may be appropriate if your intention is a "quick and dirty" study of a local process. Do not worry about individual breakdown decisions now—you can always combine tasks or split them out later—but do decide how detailed your overall map will be.

You must also decide how detailed you want the information flow mapping to be. In some engineering processes, almost every task gets *some* information from almost every other task. You should concentrate on critical types of flows—the major flow of information from input to output (which, in an ideal world, would be a linear flow of ever-increasing information), feeder flows of critical information, and those iterative or branching flows that are likely to cause rework or major iterations.

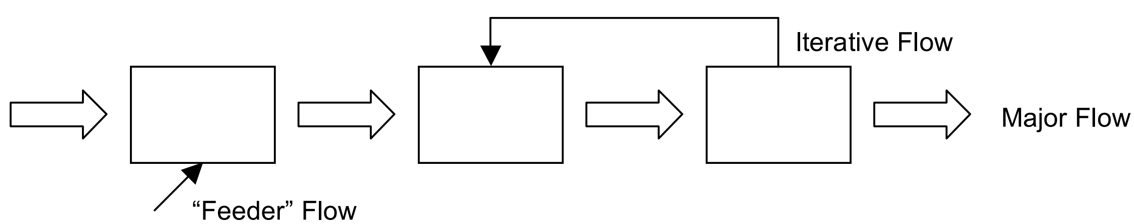
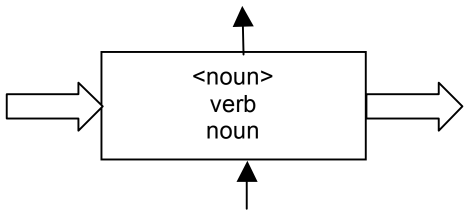
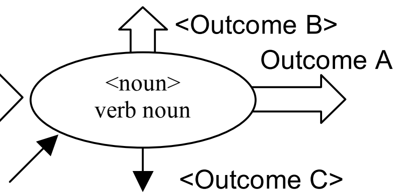
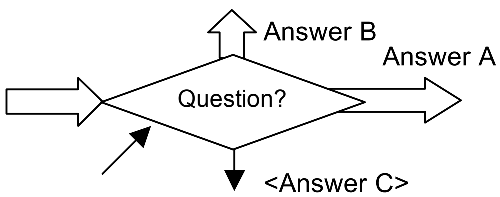
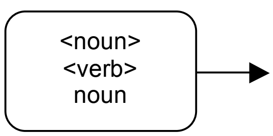
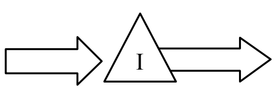
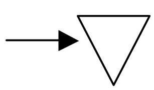
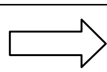
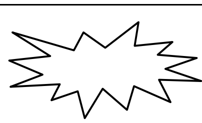


Figure 4-3. Some types of information flows

A common problem when attempting to track information flows is inconsistent answers to two simple questions: "Where does your task output go?" and "Where does your task input come from?" Ideally, each task should get input from the outputs of upstream tasks, and provide outputs that become the inputs of downstream tasks. Experience indicates this is not always the case. When in doubt, it is usually better to rely on answers to the question: "Where does your task input come from?" People are more likely to know where they need to go to get the information they need than to understand what becomes of their task output after they are finished with it.[56,57]

Some Process Mapping Symbols

Angle brackets <like this> indicate optional elements

	**Action Task - Rectangle** Example: Designer Creates 2-D Drawing Inputs: Usually one major, others common Outputs: Usually one major; others possible (e.g., when information created affects other processes incidentally)
	**Review Task - Oval** Example: Panel Reviews 2-D Drawing Inputs: Usually one major, others common Outputs: One or more major outcomes, based on the outcome of the review process *May (or may not) include implicit decisions*
	**Decision Task - Diamond** Example: Drawings Meet Requirements? Inputs: Usually one major, others common Outputs: Two or more major outcomes, based on the answer to the question; others possible (e.g., a minor output could represent a low probability answer)
	**External Factor** Example: Interface Constraints Inputs: Any combination (often none) Outputs: Any combination (often one minor) *Represents tasks or information sources external to the process under study*
	**Inventory** Inputs: One Outputs: Same as input, after delay *Represents waiting, queues*
	**Storage** Inputs: One Outputs: None *Represents archiving, knowledge base addition*
	**Major Flow** *Information on the main flow of the value stream*
	**Minor Flow** *Lower bandwidth, feeder, or rework information flow*
	**Burst** *Draws attention to special feature or problem*

EXAMPLE: Tasks and Flows

The preliminary design process is broken down into the following action tasks:

1. Choose Preliminary Configuration
2. Create External Moldline Drawings
3. Perform Aerodynamic Analysis
4. Create Mechanical and Structural Drawings
5. Determine Structural Requirements
6. Perform Weight Analysis
7. Perform Stability and Control Analysis
8. Perform Loads Analysis
9. Develop Finite Element Model
10. Perform Strength/Stiffness/Life Analysis
11. Create Manufacturing Plan
12. Develop Design Report/Presentation

Within each task, sub-tasks such as formatting, review, software setup, analysis, cross-functional communication, database search, integration, and documentation, were noted. To keep the mapping tractable, however, these subtasks will not be presented on the maps. If necessary, they can be addressed later by creating detailed maps of individual tasks.

Several reviews, and one significant set-up and data formatting tasks with work content comparable with the above tasks, were identified. These tasks *are* included in the map.

To aid in collecting data on the tasks, data collection sheets were created. They are presented in Appendix B. Such sheets may or may not help your improvement effort, and we will not dwell on them in the example, but you may wish to examine them and perhaps modify them for your use.

The results of the first mapping step are presented in Figure 4-4. The topology is typical of a product development value stream. Branching and parallel tasks complicate the flow, and there is considerable dependence on outside information not under the control of the process. The Weight Analysis task gets no input, which needs to be investigated. Reviews are concentrated near the end of the process, and can create rework or design iterations almost anywhere. All possible rework paths are not shown on the map, as it would become too confusing.

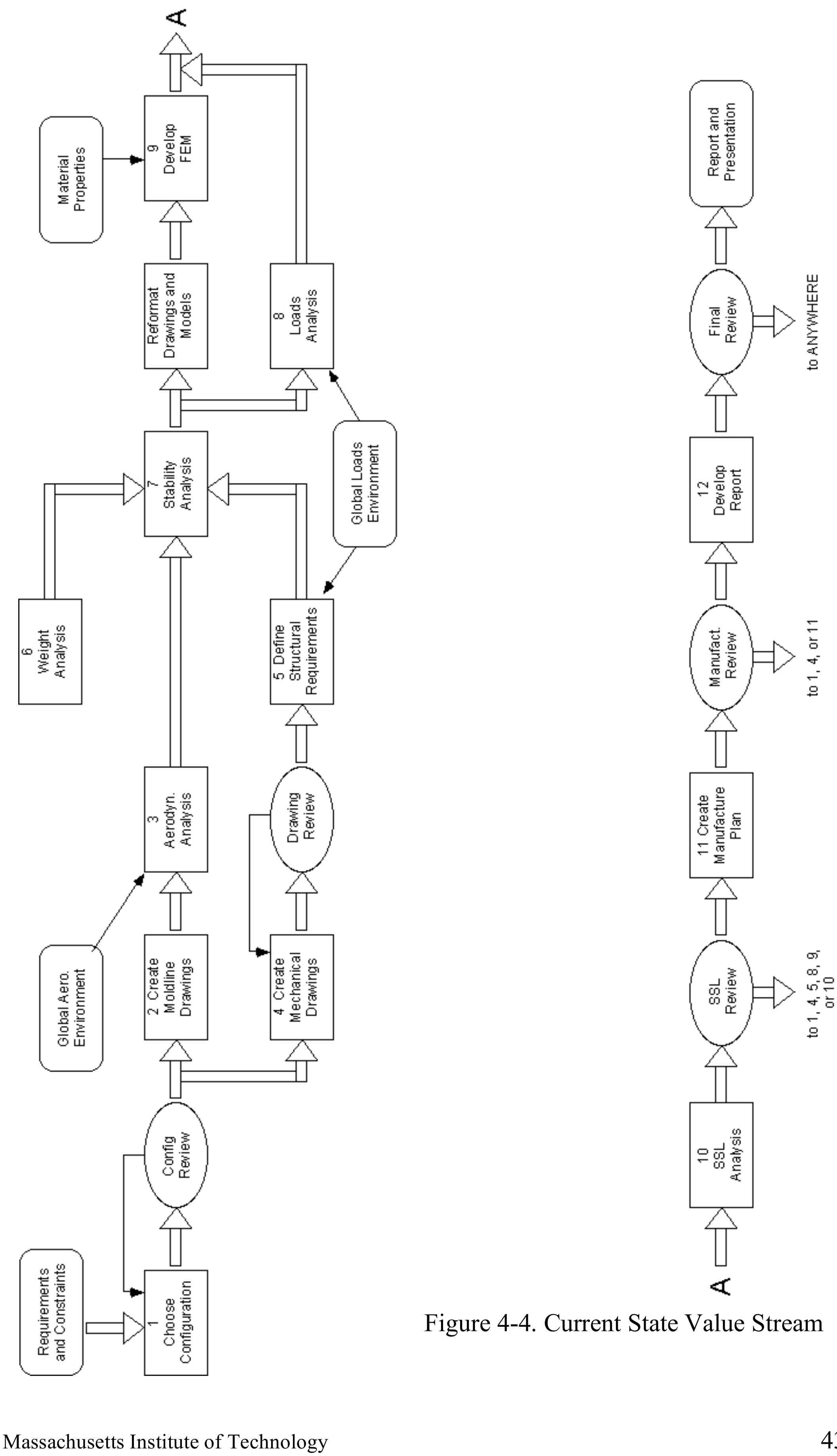


Figure 4-4. Current State Value Stream

Data

Collecting and using data requires a balance between the effort involved and the payoff expected. Although collecting task data is difficult, it is possible to collect too much of it. The point is to achieve a lean process, not build a complete model of the process. Trischler calls the latter tendency "Paralysis by Analysis."[58] The other key mistake identified by Trischler is the tendency to start with a preexisting theory and collect the data necessary to validate it. Avoid this kind of bias. Let the data do the talking.

In that spirit, a few guidelines are collected below. Given what you need and the resources available, collect the data. A form such as the one in Appendix B may come in handy for data collected while walking the process.

The key source of data will be the people on your team. They should be encouraged to brainstorm on where useful data might be. They will know what difficult-to-interpret data means. They will also, almost certainly, have to estimate much of the more detailed process data based on their own experience with the process. There is nothing wrong with this, although one must always guard against biases.

The team members will probably have to put some effort into normalizing data from disparate sources. Watch for simple unit errors—is time in hours or minutes? Be careful of the basis of the data—is the time per person or per process step? When mixing older and current data, make sure nothing has changed between measurements that might alter the relative meaning of the data. A big help is to normalize the data to a "typical" job, perhaps also specifying the range of possibilities, i.e. the time for an average job, with minimum and maximum times also collected or estimated (see Appendix C for an example of this).

The data collected is appended to the value stream map using data boxes. This is illustrated in the example. The aim on the map is to include the data that will provide insight into the process, not simply to display everything. In the example, a Rother-and-Shook-style time line across the bottom captures the time data in compact form. Up to three tasks are proceeding in parallel in the example, so there are up to three timelines; the summed-up cycle times at the end are the sum of the times on the longest path.

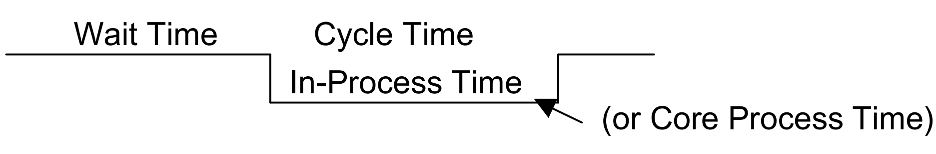


Figure 4-5. Timeline notation used in running example

A strong caveat is appropriate here: there are no standard conventions or definitions for value stream data. Even Rother and Shook (at least in their first edition) used contradictory definitions of "cycle time." The lessons here are 1) see if there are standardized *local* definitions of terms where you work and 2) either way, define the terms and conventions *you* use carefully and apply them consistently. Figure 4-5 shows the convention we will use in the running example in this document.

It is, in any case, difficult to define a set of absolute standards because different metrics and methods of display may be appropriate to different situations and levels of analysis. For example, the timeline shown in Figs. 4-5 and 4-6 is appropriate for higher-level maps where large waits are identified as external. Smaller waits, during which the task is put down only to be picked up by the same group a little later, are buried in the task—this is why the cycle time is not equal to the in-process time. For different maps, different timelines may be appropriate. For example, in portions of the map in Appendix C, only aggregated times for blocks of tasks are reported, as that was the only data that was available.

Basic Advice for Current State Data Collection

Concentrate on what you need: If the problem is process cycle time, collect time data. Task lead times, cycle times, actual work times; "inventory" wait times between tasks, delays due to reviews, resources not being available, etc. If the problem is quality, look for data on errors and rework, and try to correlate errors with root cause factors. If the problem is interfaces with other processes, look at queues at the beginning of the process, reviews at the end, coordination practices, data formats.

Exploit what you can find: Examples of data that may exist include "timecard" records, sign-in/sign-out sheets or databases, output reports, test records, etc. Often the data you want is not the primary purpose of the existing documents; for example, the dates on reports (as a record of cycle time) may be more valuable to you than the contents. In extreme cases, you may build your study around a particularly rich data resource, as Weigel and Warmkessel did when analyzing the satellite testing value stream by examining thousands of existing discrepancy reports.[59]

Make do with what you have: Often your data will be imperfect. Times and error rates are likely to be estimates. Quantitative data will probably exist in "small numbers," making formal statistical analysis dubious. Mean values will thus be properly treated as estimates, and variations as rough estimates. The data will be affected by the act of observing and by the bias of the data takers. All of this is normal, and most of it will not matter. If the data leads you to the critical problems, it is adequate.

Be honest: Identify the sources and likely reliability of your data. Note level of estimation, possible biases, and any other problem with the data. This will make it much more valuable.

Dig deep (only) when you must: The only time you want to do further work to get either greater accuracy or a finer breakdown of the data (by, say collecting enough for statistical analysis) is when you are analyzing the critical problems and looking for solutions. In particular, if your analysis hinges on a small set of data, make sure this data is as accurate as you can practically make it.

METRICS: Task Data

Warning – the definitions here are *not* standardized. Beware of conflicts with local standards and previous work, and carefully define the metrics you use.

Some of the many possible metrics for tasks:

- Cycle time (CT): clock or calendar time it takes to do an instance of the task
- In Process time (IPT): hours of continuous work it take to do the task. Sometimes referred to as Touch Time (TT)
- Core Process time (CPT): hours (or other time units) of continuous work spent on core task, excluding set up, trouble shooting, information gathering, etc. Sometimes called Value Added time (VAT)
- Efficiencies such as IPT/CT, CPT/CT or CPT/IPT
- Lead time: time from known need for task until task completion. Sometimes defined as time from known need until start of active work on task. Use either (but not both!) of these definitions
- Set up or Changeover time: time needed to prepare resources to do or resume task
- Fixed or non-recurring costs (what resources must exist for the task to take place, even if they are not used continuously)
- Cost/job or recurring costs (what resources are expended to do a job)
- Capacity: how many jobs can be done by a given process or sub-process, with available resources, in a given time.
- Utilization: at the current workload, how much of the Capacity is actually needed/used.
- Availability: percentage of time resources are actually available when needed
- Variation in any of the above (how predictable is the process)
- Failure rate: percentage chance a task output will be defective, or that a review is failed, resulting in rework
- Repeat or rework rate: number of times a task needs to be performed in an iterative or re-work prone process
- Downstream task satisfaction (how good is the task output)

Waiting and Inventories can be measured by:

- Inventory part count: number of jobs in queue (usual factory metric)
- Delay time: average time a single job waits in queue (typically more useful in PD)
- Delay time statistics: mean and deviation, or distribution, of wait times (best)

Data can also be collected on the information flows, e.g.:

- Form: report, computer file (what system?), email, phone call, etc.
- Format: standard form? Software specific computer file?
- Size (of file or document)
- Transmission times and/or costs

EXAMPLE: Data Collection

Based on the known problems, data on time, rework rate, and information form and format was desired. The primary source of the data was the team. Where the team lacked personal knowledge, but knew who might have it, structured interviews using the form in Appendix B were carried out.

The following task metrics were selected.

Cycle Time (CT) (hours, based on an eight hour day)
In Process Time (IPT) (hours)
Activity Based Cost ($, or charged hours)
Special Resources Required (varies)
Chance of Rework (percent)
Time Spent on Rework (hours of additional process time)

The time metrics are based on estimates from both the process manager and the performing engineers; they are thought to be reasonably accurate. The cost is a rough estimate expressed in charged engineering hours; a more ambitious desire to include the true activity cost (including overhead and resource costs) was found to be too complex. Special resources were identified as possible constraints. The chance of rework at the task level proved extremely difficult to estimate, and the numbers tended to be badly biased. Some engineers took this as a criticism of their work, while others included normal iterations in this metric, rendering it of doubtful value. Where significant waiting was observed in the flow, a rough estimate of the average wait time was used as an “inventory” metric.

Information was also collected on the form and format of the data flows. These proved sufficiently diverse that no metric could be created. The form of the data flows is identified only in cases were it is a problem (requiring significant clarification or reformatting). This issue will be returned to in the value evaluation, next.

The time data is presented on the process map in Figure 4-6. To clarify the time data, a “castle-wall” time line is drawn across the bottom of the map, in the style of Rother and Shook. Also, inventory boxes have been added where waiting for the task to begin was identified as a major problem. One, in front of FEM development, is due to a resource constraint, and is on the critical path. Others, after loads and structural requirements, *seem* harmless. These tasks tend to get done early, and their results wait to be picked up once other tasks have caught up. Waiting also happens before major reviews due to scheduling trouble. An information-gathering task is added to the map to describe how the weights group collected their inputs.

Overall, the map captures the slowness of the process. There is a major bottleneck at FEM, which has a long cycle time, a major set-up task, and a large average wait time in front of it. There is also a build-up of minor delays due to the many tasks that have to be done sequentially, especially in the review cycle near the end.

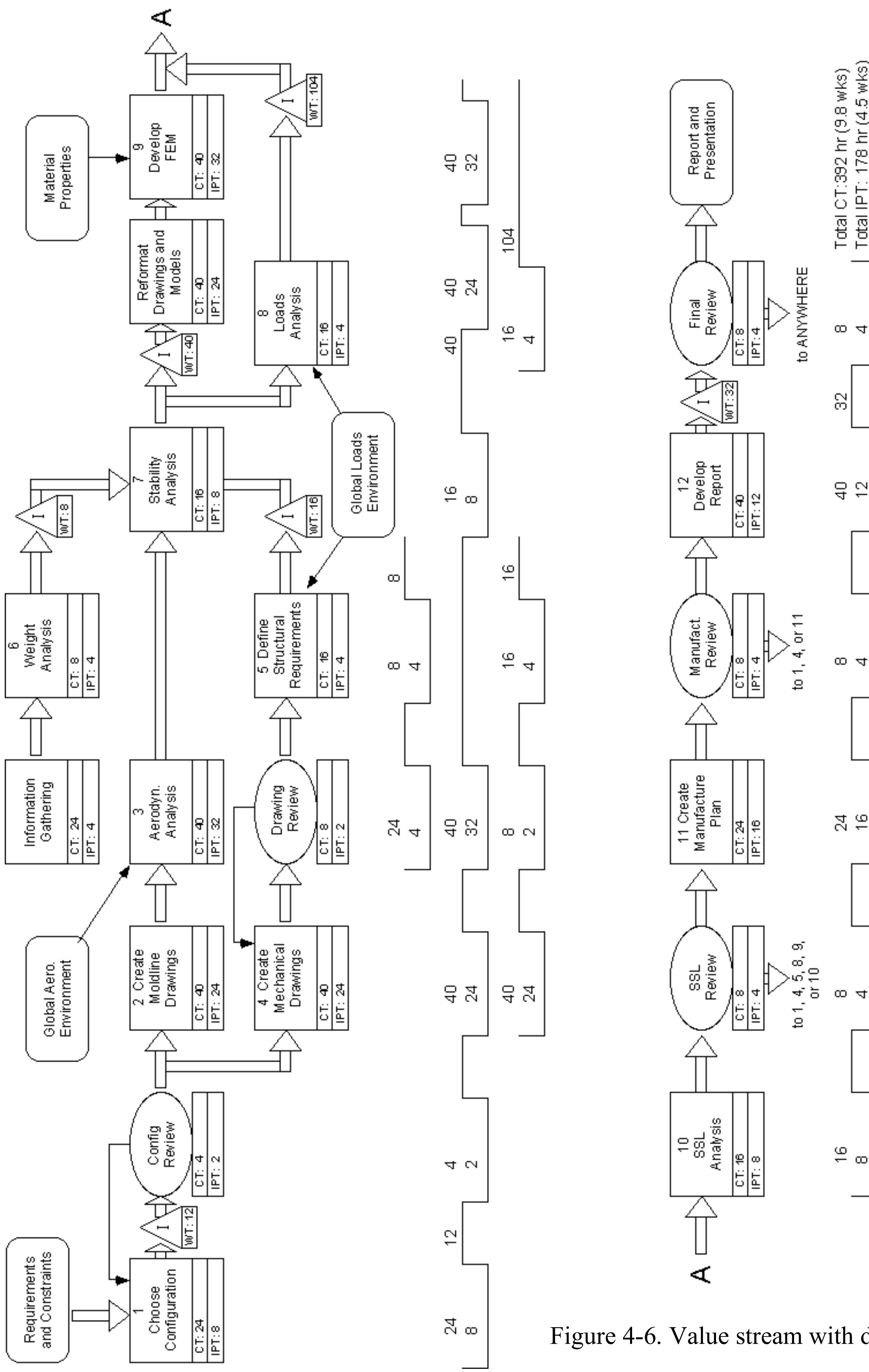


Figure 4-6. Value stream with data

Evaluation of Value

The difficulty of assessing value at the task level was discussed in Chapter 3. Traditional categories of "value-added," "necessary non-value-added" and "pure waste" are not always useful when taking a first look at the value of the current state tasks. Time and effort can be wasted arguing nuances of "value-added" versus "necessary non-value-added" in ways that do not add value to the value stream map!

That said, it is possible that some maps, particularly if they are more detailed, lower level maps, may actually reveal non-value-added tasks. It is worth taking a quick look at your map for such tasks, but do not be surprised if you do not find any. As Browning points out,[60] most non-value-added activity is hiding inside of value-added tasks, but adding more detail to your map *ad infinitum* is probably not the efficient way to find it. Your map may also reveal major waiting (inventory) chokepoints. If any examples of clear "pure waste" do reveal themselves, highlight them now!

Most of the tasks on your map will be necessary for the completion of the overall process, at least in the current state. For now, evaluate *how* they add value. Do they add to the product definition? The process definition? Do they reduce uncertainty in ways that make the definitions more valuable? This will be very useful for eliminating waste within the tasks, and streamlining flow between them, which we will discussion in the next two chapters. Many of these categorizations can be done at a glance—see the example.

You may wish to evaluate how well this value is created, or how much of it is created. The example data collection sheet in Appendix B includes a simple 1 to 5 relative scoring of the effectiveness of each task for creating each of the key value types. This information is most easily collected by a self-assessment of those doing the task, but this sort of data is unavoidably biased, and therefore of limited usefulness. An evaluation based on external observation of the task, on the other hand, will be less biased, but may be inaccurate and/or labor intensive. Decide how the usefulness of the evaluation balances with the effort involved, and the possibilities of inaccuracy and bias, before embarking on this sort of value assessment.

Value assessments can also be made of the information flows between the processes. These assessments are best done by those on the receiving end of the data flow. Again, the first check is for non-value-added information—if useless information is being produced and propagated this is pure waste. More likely the information is necessary, but it may not be of good quality due to incompleteness, errors, or formatting problems. A simple relative ranking of the quality of information may be quite useful in many cases. The data collection sheet in Appendix B includes simple assessments of the perceived completeness, quality, and format of data. This assessment should be made by the users of the data, not its creators.

Value information can be displayed on the map in a variety of ways. Tasks can be "tagged" as Value Added (VA), Required Non-Value-Added (R-NVA) or Non-Value-Added (NVA). If only a few NVA tasks are evident, they should be highlighted by "burst" symbols. Types of value contributions can be identified by acronym codes, color, hatching or other symbols. Problematic information flows can be identified by "bursts." Adding pseudo-quantitative quality assessments to the data boxes is probably not useful. The point here is not to display all the information that has been collected, but to make visually obvious the information you will need to start the next stage of your mapping—creating a future state.

EXAMPLE: Evaluation of Value

In our example, a quick evaluation of the tasks revealed two non-value-added activities. One was the round of data collection performed by the weights group. The other was the drawing review, which was determined to be a formality, and to not reduce risk or uncertainty in any meaningful way. These are identified on the map with "NVA" bursts.

The data collection form in Appendix B was used to identify the relative contributions the tasks made to the key value aspects. The team members scored each task from 1 to 5 on each aspect, and then calculated an overall value score. However, even this relative score proved problematic. The mapping team members often had to rely on self-assessments of the personnel doing the tasks. Biases and inconsistent interpretations further compromised this "data." In the end, the tasks that were not identified as NVA were simply categorized by the primary value aspect they did create: product (V1) or process (V2) definition, risk and uncertainty reduction (V3), or formatting for downstream use (V4).

The information flows were evaluated by the users of the information, using the data collection form. This evaluation identified problems with the quantity, quality, and formatting of data in a number of places. These problems are identified with bursts. In particular, the users of the process output found the report format almost useless. The information in the report was not ready for further use.

The complete current state map is shown in Figure 4-7. A few soft targets can be seen. The NVA tasks should be eliminated, but note that they are not on the critical path—cycle time will not be improved. The low quality of the data flowing into the FEM task, on the other hand, is on the critical path, and at the bottleneck! The other low-quality information flows—the incorrect global loads—seem to be off the critical path. However, there is a problem that can be seen when the map is examined critically. This unstable information is going into requirements and loads which are produced, and then stored. If the input is unstable, there is a good chance the outputs will become obsolete during "storage." This is a classic "inventory" waste: the information has gone bad in the "warehouse."

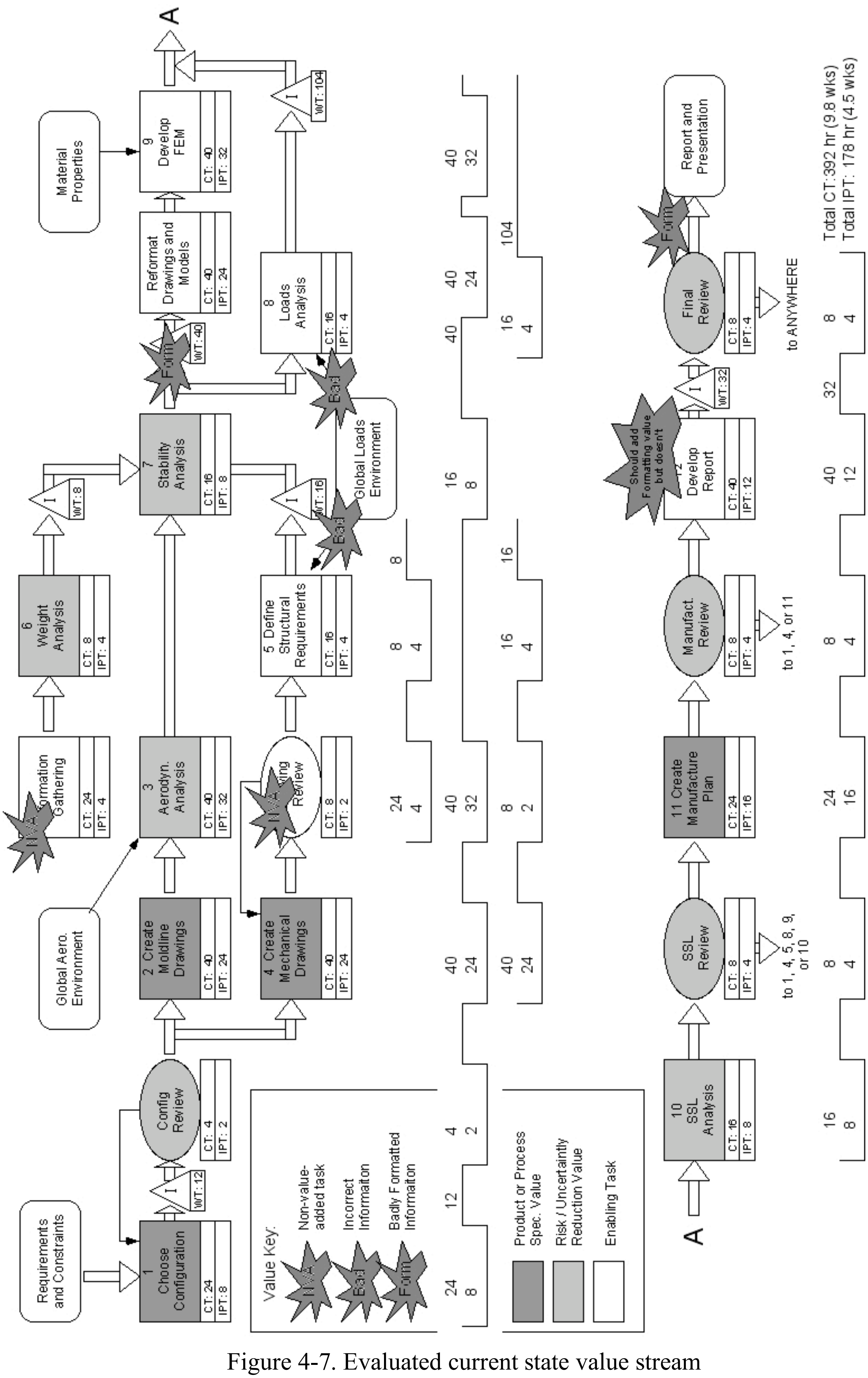


Figure 4-7. Evaluated current state value stream

Understanding Iteration: the DSM

Typically, a product development value stream mapping will have branching and iterative information flows. This can lead to excessive complexity on a traditional process flow map, to the point where its usefulness as a tool for visualizing the process is lost. For these cases, an excellent tool is available: the Design Structure Matrix, (DSM). This tool is presented very well on the MIT DSM website, http://web.mit.edu/DSM. A tutorial is available there, which is required reading if you which to understand this tool. Rather than duplicate the tutorial, we will give the briefest possible description of the DSM and show how it can supplement the process map in our example problem.

The DSM is a square matrix with N rows and columns, one for each of N tasks in the process. Information flow between the tasks is shown by dots in the cells of the matrix. If task m provides information to task n, a dot is placed in the cell in column m, row n. One hopes that information would flow from earlier tasks to later ones. If the tasks are arranged in nominally chronological order, having dots only below the diagonal means information is flowing only from early tasks to later ones. A dot above the diagonal represents information flowing from a later task to an earlier one—an iteration. The further away from the diagonal the dot is, the longer the iteration loop will be. Other aspects of the information flow, such as branching and recombining, processes running independently in parallel, or processes that are fully coupled to each other, can all be seen at a glance on this kind of chart.

Displaying information flow with a DSM loses much of the richness of the process map, but it compactly represents complex information flows in a way that is useful for both at-a-glance evaluation, and quantitative analysis. In the example, you will see the former. Examples of the latter, including codes, can be found on the website.

EXAMPLE: a DSM

Figure 4-8 shows a DSM for our example process. The reviews, and the added tasks (info hunting, waiting) are not shown, and the data (in particular, the time data) are not visible. Much more detail on the information flows is visible, however. Note we have added a feature to the DSM—the very small dots represent information flows that can in theory happen, but usually don't, while the normal dots represent information flows that happen in most cases.

The dots below the diagonal represent "forward" information flow. Where this was shown as a continuous stream with "feeders" on the process map, the DSM reveals how many tasks require information from tasks far up the chain. The obvious example is the final review preparation, which requires input from all previous tasks. In the process map, it is assumed that this information flows down the value stream to its final destination, but this may not actually happen. Without some mechanism for the information to survive the handoffs down the chain, the review preparers may have to go information hunting.

Dots above the diagonal represent iterations or rework. Some of these are nearly harmless. Note the feedback from task 3 to task 2; this is not seen on the process map because it is done "ad hoc" between the aerodynamic analyst and the drawer of the moldline and doesn't involve a formal iteration process. Similarly, the FEM may be debugged during the SSL analysis. (feedback from 10 to 9). This illustrates a general principle, which is that feedback between "adjacent" tasks is common, and relatively easy to deal with (either *ad hoc* or by forming a small working group).

The big problem is the feedback from tasks 10 and 11 to task 4, representing mechanical design modifications uncovered during SSL and manufacturing analyses. This sets the process back almost to the beginning, and is quite common. There are also many unlikely but possible feedback loops that would be just as disruptive were they to occur. The presence of many possible feedbacks also gives one an intuitive feel that the process would be difficult to predict, manage, or schedule. This clarifies some reasons why the current state process is both slow and highly variable.

		1	2	3	4	5	6	7	8	9	10	11	12
Choose Preliminary Configuration	1	1									•	•	•
Create External Moldline Drawings	2	●	2	●								•	•
Perform Aerodynamic Analysis	3	●	●	3									
Create Mechanical Drawings	4	●			4	●					●	●	•
Determine Structural Requirements	5				●	5					•		
Perform Weight Analysis	6	●	●		●		6						
Perform Stability and Control Analysis	7	●		●		●	●	7					
Perform Loads Analysis	8			●				●	8		•		•
Develop Finite Element Model	9		●		●		●			9	●		
Perform Strength/Stiffness/Life Analysis	10	●				●			●	●	10		
Create Manufacturing Plan	11		●		●							11	•
Develop Design Report and Presentation	12	●	●	●	●	●	●	●	●	●	●	●	12

Figure 4-8. DSM map of example process

Summary

A checklist for mapping the current state:

- <optional> Create or obtain a "Zeroth" map to start the visualization
- Define and follow the work being done by the process

- Define tasks that make up the process
- Arrange the tasks and define the information flows between them. Work "in pencil."

- Collect and interpret process data
- Apply selected process data to the map

- Assess value (or lack thereof) of the tasks and the info-flows between them
- Apply relevant value judgments to the map

- <optional> Use a DSM mapping to better understand information flows and iterations

Be flexible. Adapt the mapping, data collection and display, and value assessment methods to the needs of your problem. *Do not, however, skip steps or proceed to the next chapters until the current state map is complete.* Let the map and the data, not your preconceptions or biases, lead you to the next steps: finding wastes and improving the process.

5.0 IDENTIFYING WASTE

Now that you have a vision of your process, you can set about improving it. The first and most important step is the elimination of waste from the process in all of its forms. The sources of waste in product development processes are many and varied, so there cannot be a cookbook for their identification and elimination. This section will draw on the collective wisdom of the LAI product development community to provide guidelines. They will be directly applicable in many cases. In most cases they will at least provide an illustrative example to get you started.

This section will review the *info-wastes*, a re-interpretation for product development processes of the lean seven wastes. The info-waste concept has grown out of collaborative work in the LAI product development community.[61] Some compiled wisdom and best practices for eliminating the most common of these wastes will follow.

Recent work has shown that some wastes do not fit fully into this framework,[62] and there is value in digging into the root causes of these wastes.[24] However, the waste framework presented here remains a good first step. Look deeper, particularly into the advanced waste framework of Bauch, if your problem does not fit this framework.

The Info-Wastes

The flows in product development are of information rather than physical materials. Therefore we would expect to find wastes associated with the information flows, analogous to the seven wastes identified in the factory. The seven info-wastes include:

> ***Waiting***: Late delivery of information; Delivery too early (leads to rework)
> ***Inventory***: Lack of control; Too much in information; Complicated retrieval; Outdated, obsolete information
> ***Over-Processing***: Unnecessary serial production; Excessive/custom formatting; Too many iterations
> ***Over-Production***: Creation of unnecessary data and information; Information over-dissemination; Pushing, not pulling, data
> ***Transportation***: Information incompatibility; Software incompatibility; Communications failure; Security issues
> ***Unnecessary Movement***: Lack of direct access; "Walking" the process
> ***Defective Products***: Haste; Lack of reviews, tests, verifications; Lack of interpretation (raw data delivered when information or knowledge needed)

Table 5-1 gives examples and causes of information wastes. Study it. Do not hesitate to brainstorm further examples relative to your process and add them to the table. The point is not to create an inclusive list of all possible wastes, but to get the team thinking about waste, and provide them with a common language for categorizing and discussion wastes.

Table 5-1. Information Wastes[63]

Types of Information Waste	Examples	Causes
Waiting *Idle time due to unavailable information*	People waiting for information	• Lack of access • Untimely updating of data bases • Multiple approvals • Poorly designed or executed process to provide information
	Information waiting for people	• Information created too soon may be obsolete by the time it is used
Inventory *Information that is unused or is "work in progress"*	Too much information	• Poor understanding of user needs
	Multiple/redundant sources	• Tendency for everybody to maintain their own files
	Outdated/obsolete information	• Lack of "version control" • Lack of disciplined system for updating new and purging old information • Inadequate archiving standards or practices
	"Just-in-case" information	• Collection, processing and storage of every element of data that process participants can think of, whether or not a specific end use has been identified
Excessive Processing *Information processing beyond requirements*	Excessive/custom formatting	• Lack of standardization
	Numerous, fragmented reports	• Poor output design • Lack of understanding of the needs of the users of process outputs
	Unnecessary serial processing	• Poor system design • Lack of understanding of concurrent processing capabilities
	Excessive approvals for information release	• Stove pipe, command and control mentality • Turf protection
Over Production *Producing, distributing more information than needed*	Unnecessary detail and accuracy	• Tendency to "over-design" • More detail than necessary in early design
	Pushing, not pulling data, information	• Uncontrolled process
	Over-dissemination	• Poor understanding of each participant's needs • "Send all information to everyone," rather than to meet specific needs

Types of Information Waste	Examples	Causes
Transportation *Unnecessary movement of information between people, organizations, or systems*	Information handled by multiple people before arriving at user	• Lack of direct access due to IT system limits, organizational inefficiencies, knowledge hoarding, security issues
	Information hunting	• Lack of clear information flow paths, failure of process to produce information needed
	Data re-formatting or reentry	• Incompatible information types (drawings *vs.* digital descriptions) • Incompatible software systems or tools • Lack of availability, knowledge, or training in conversion and linking systems
	Switching computers (e.g., CAD to PC) to access information	• Software/hardware incompatibilities • IS support
Unnecessary Motion *Unnecessary human movement (physical or user movement between tools or system)*	Walking to information, retrieving printed materials	• Lack of distributed, direct access • Lack of on-line access • Lack of digital versions of heritage information
	Excessive keyboard, mouse operations	• Lack of training • Poorly designed user interfaces • Incompatible software suites • Too much information to sort through
	Poor physical arrangement or organization	• Team members not co-located • Organization structure inhibits formation of right teams
Defects *Erroneous data, information, reports*	Errors in data reporting/entries	• Human error • Poorly designed input templates
	Errors in information provided to customers	• Lack of disciplined reviews, tests, verification
	Information does not make sense to user	• Raw data delivered when user needs derived information, recommendations, or decisions

Slack[37] carried out a survey of waste in product development. Forty-one surveys, of which 25 were returned, were sent out asking for a single example of waste in product development processes. The results were tabulated using the seven waste categories, plus two more used by Slack. The results are shown in Figure 5-1. Waiting and over-processing accounted for roughly half of the answers. More wastes fall into the closely related categories of intellectual inventory (information waiting to be used) and overproduction (of processed information). Note that the two extra categories added by Slack—Complexity and Time Lag—proved less important than all but one of the original seven info-wastes.

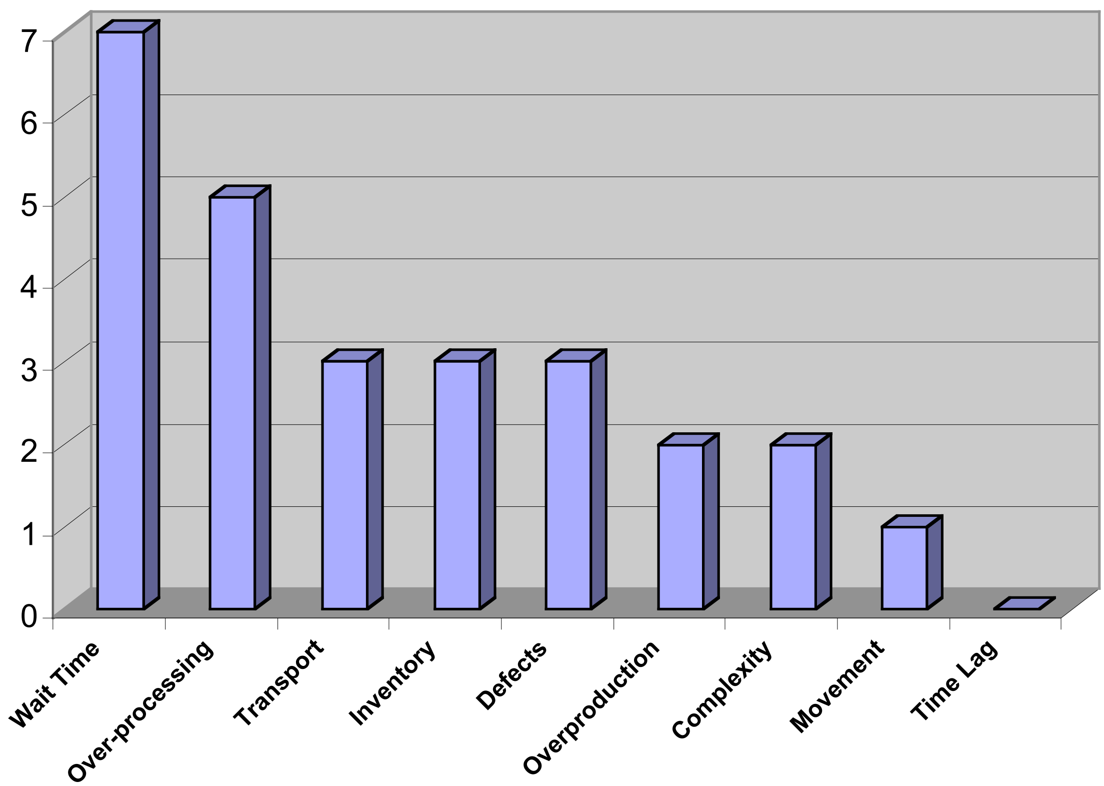


Figure 5-1. Frequency of waste categories found in survey of waste in product development[64]

6.0 IMPROVING THE PROCESS

Figure 5-1, as well as the discussion in Chapter 2 (e.g., Figure 2-4), point to wait time as the leading category of waste in product development processes. This is an inevitable result of the batch-and-queue nature of many product development processes. Each task works until it is complete, then "pushes" its results to the next task. The next task picks up the work when all of its inputs are ready and the resources (usually personnel) to do the job are available. This cycle of task results push, followed by waits for information and resources, is repeated until the process is complete. This virtually guarantees at least some waiting between each task. Worse, this sort of system is vulnerable to instabilities and bottlenecks.

A simple example is the analysis tasks in our example problem. Note the time data in Figure 4-6. Building the finite element model (FEM) is the slowest of all the tasks in the process, and is labor intensive. This causes work to "pile up" waiting for personnel to be available, making a bad situation worse. When the FEM is ready, the loads may not be, resulting in more waiting before the analysis can be carried out. The loads analysis task is straightforward, requiring little time or effort, but is not synchronized with the production of the FEM.

The tendency of batch-and-queue systems to produce bottlenecks is well understood from a factory point of view (see Womack, and Rother and Shook, for simple factory examples). The factory examples can be applied, in many cases, directly to PD operations. A major caveat is that inherent uncertainty, and the possibility of deliberate iterations, may complicate the analysis of PD processes. If a deeper look is needed, formal queuing theory, Goldratt's theory of constraints, or system dynamics modeling may be applied. These approaches are particularly appropriate for more complex processes, for which sufficient data can be cost-effectively collected. They are discussed briefly, with references, in the next chapter.

Here we will concentrate on best practices for *quickly* reducing info-wastes and clearing bottlenecks to flow. The suggestions here are based on member company experience. They basic concepts will sound familiar from factory lean; the are modified for use in a product development environment.

Note the emphasis is *not* on eliminating Non-Value-Added (NVA) tasks. This is rarely as simple as marking them as such on the value stream map and omitting them in future. Most often, NVA activities are hidden within tasks that are value added. Other apparently NVA activities actually should have value, but are executed in an NVA way. We will, however, review some frequently noted classes of NVA activities.

Establish a Takt Time

In factory lean, the single most important concept is achieving a takt time. Applying the concept of takt time to PD processes requires some creativity. Only rarely will a PD process require large numbers of nearly identical jobs to be performed in a limited time. Drawing release on an active development project, and change order processing in support of an active production line, are two examples of processes that may meet this description. In these cases, the factory definition of takt time can be applied almost directly—takt time is the number of units of work demanded by the customer divided by the available time. All tasks in the process must then proceed at this pace.

Average Takt Time: In other cases, an averaged takt time or throughput rate can be useful. If 50 jobs of varying levels of complexity must be done in a year, on *average* about one job a week must be completed, even if the actual release pace may vary with variations in the workload or job complexities. The process must be able to support this average takt time, or the overall demand for work cannot be met (at least without expensive and inefficient expediting and/or overtime).

Psuedo-Takt Time: In many cases, a *pseudo-takt* time needs to be set based on the needs of the enterprise. In our example, the enterprise demand is for a total calendar time of four weeks. We would like to use the pseudo-takt time as a "metronome" to synchronize the process without relying on complicated scheduling or disruptive high-priority interruption of other work.[65] To calculate a takt time, we could calculate an average takt time as defined above, but this might mean that a complex job or several closely spaced jobs would be delayed. An alternate would be to propose a desired "surge" capacity in terms of jobs per unit time, but this may result in excess capacity most of the time. We could also simply divide the cycle time up into convenient increments. Defined this way, the pseudo-takt time is much less rigid than a factory takt, but still serves the function of synchronizing the process. Note that it should be less than or equal to the average takt time defined above or the process will not have the capacity to get the necessary work done![§] Once a takt or pseudo-takt is defined, it can be quickly and intuitively used. The tasks should be, to the extent possible, regrouped or redefined to work to the takt time. The *tasks* that must be improved are the ones that cannot make the desired takt time. Improving any other tasks will not help the process at all—they will simply pile work up at the bottleneck task at a more rapid pace.

Coordinating: The ideal, even for processes that will perform only a small number of jobs, is "factory style" time coordination—all tasks working at takt time, reaching completion and handing off to the next task at predefined and predictable intervals. Coordination, handling of exceptions, and sharing of necessary process information should also be built into the pseudo-takt period. For example, short weekly "standup" meetings (or telecoms) can be assure a common picture of progress, make sure needed information flows, adjust for unexpected events, and plan the next week. Care must be taken, however that these meetings be kept short, and that planning not devolve into rigid scheduling. A common root cause for long process times is that waiting is simply scheduled in by conservative estimates of how long it will take to do things!

[§] Indeed, to handle the variation and unpredictability inherent in the PD process, it is best if the process has 20-30% more capacity than a steady-state analysis would indicate is needed.

EXAMPLE

In our example process, a traditional takt time is hard to set as the demand for preliminary designs varies, as does the difficulty of these jobs.

An *average* takt time is calculated from the current enterprise need for about 35 of these jobs a year. The avaerage takt time is then (50 available weeks/yr) / (35 jobs a year) = 1.4 weeks/job.

A "pseudo-takt" time of 1 week is selected with the following logic:

- A longer time would be too large a fraction of the desired 4 week cycle time to be useful
- A shorter time might not be realistic, given the current FEM in-process time of 7 days
- The theoretical capacity of 50 jobs/year provides plenty of capacity margin for the current demand of 35, both to buffer against uncertainties and to provide an enterprise resource that will be an advantage for future proposal work
- Handoffs at the ends of the work week fit in the current enterprise work planning method
- Failure to make takt time can be recovered by weekend overtime (not desirable, but practical—buffers the new process from the inevitable troubles that accompany change)

A quick examination of Figure 4-7 identifies two challenges—reducing the task time of the FEM preparation (Reformat Drawings and Develop FEM), and organizing the other tasks in logical groups to support the takt time.

Assure the Availability of Information

Waiting due to lack of information, or time wasted hunting for information, are common in member experience. Making information available is a challenge to which several lean solutions can be applied. Possibilities include:

Practice Information 6S: Make sure all information is stored "neatly." Clutter (e.g., obsolete variants, drafts, incorrect work) should be ruthlessly cleared away. Good information should be sorted and stored systematic, standard, and simple ways. Information systems should be simple, safe environments, with standardized applications, and updated firewalls and virus protection. Note the best solutions here are not necessarily high-tech. Keeping all information in a centralized database system is analogous to controlling inventory with an MRP system: potentially effective, but not necessarily the lean solution.

Collocate an IPT: Physical and organizational proximity greatly increase the flow of information. This is also a lean advantage of integrated concurrent engineering. These strategies have many advantages which will be discussed later; free information flow is a major one.

Make information visual: "War rooms" where key information is kept on the walls and constantly updated are a time-honored method for high-priority projects. It can be generalized. Team metrics (progress) and lean metrics (improvement) can be displayed on simple progress boards for all to see. Common libraries of technical data and project documents can be shared at a central location. Although these solutions are in general *not* IT-driven, database systems or specialized *visual* information systems that are currently under development[66] may be appropriate for some projects, especially where co-location is not feasible.

Make information flow physically: A simple trick used by some members is to place all key information pertaining to a job in a folder, binder, or other simple physical form and flow it through the process. This no-cost solution assures instant information availability and version control. It provides a visual marker as to where the process is—it is clear who is holding the folder. It has the added advantage for continuous improvement purposes that process data can be collected as the folder moves through the process—simple sign-in time, sign-out time, and engineering hours used entries can be made every time the folder changes hands.

Pull, don't push, information: Information should flow when it is needed by downstream processes, not when upstream processes are done. Pushed information may not be needed by many of the recipients, and it will almost certainly not flow "at the right time."

These are just a few of the possibilities. Not all will apply to your process, and your imaginations should not be limited by the list.

EXAMPLE

The obvious case in our example is the weights task having to hunt for their information. This can be fixed on the value stream map by requiring the producers of the needed information to supply it! Less visible on the value stream map would be the establishment of a simple data structure on either a server or a website where the latest version of process documents could be stored, and a process folder that would travel with the value stream, containing necessary non-electronic documents. These measures would assure continuous availability of information already developed by the process.

Balance the Line

A first step here is realizing there is a "line" – a value stream that needs to flow. In an uncontrolled process with variations of any sort, queuing effects will tend to build in waits between each task, and pile up work at bottlenecks. Waiting can also be triggered directly by effects external to the process, such as lack of resources, information, or authority to proceed. If resources, typically personnel, are not available to start a task, it waits. If externally-provided information is not available, tasks either wait, or proceed with "bad" information only to rework later. Finally, if external approval (e.g., from a review) is required to proceed, and it is not forthcoming, the task will wait. To minimize this waiting, all of the following are necessary:

Eliminate bottlenecks: Bottlenecks are defined as tasks which take more time to do, and/or have less resources available to them, than the rest of the process. Resources and work must be allocated so that, at least on average, each task (or, as we shall see, group of tasks) takes about the same amount of time. Only when the line is balanced in this sense is there any hope that work will flow through it. Note that shifting work between tasks may be easier said than done: skills (or lack of them), cultural resistance, and silo organizations are all potential barriers.

Assure the timely availability of resources: Sometimes this is as simple as making sure that the bottleneck processes are sufficiently staffed. Often it is more subtle; even staff that is assigned to a processes may not be available when needed if they have too many other priorities. Kato found that a major root cause of delay was staff called away to "fight fires" on other projects.[24] Over-commitment can also create inefficiencies due to "changeover" times as staff switches projects. Finally, overcommitted staff may be expected to attend meetings and other events that are not a burden for any one project, but cumulatively across many projects can be a major time drain.[67]

Minimize and buffer variation: Variation is an intrinsic property of product development processes, as not all challenges faced by the PD team can be anticipated ahead of time. This variation must be minimized by whatever means available. At a high level this means things like taking high-variability activities such as technology development off the critical path, and having fall-back strategies to buffer against technical failures. At lower levels it means having the training necessary to allow simple processes to be standardized, and the support to avoid delays due to equipment or software failures or "crashes." Variations cause changes in workload that must be both actively managed (by switching resources to task that are in unexpected difficulty) and buffered against (by having a reserve of resources available).

Clear external constraints: To the extent possible, clear external causes of waiting. The causes of, and solutions to, externally caused delays are too numerous and varied to be fully explored here. Experience indicates that techniques such as the "five why's" can often get to the root cause of minor delay problems, and once identified they are easy to fix. The toughest class of external delays are those due to resource shortages. Here is where your management "top cover" is vital; if the process is compromised by lack of personnel or other resources, it must get priority access to them.

EXAMPLE

The example has a clear bottleneck at the FEM formatting, setup and analysis. A simple solution (more personnel) is tempting but would be expensive. Cross-training other personnel to do the model and SSL analysis is not practical; these are specialized jobs which *must* be done well, and there are firm "silo" boundaries between the analysts and other engineers working on the project. This problem needs a multi-pronged attack. We will see later in this section various things we can do to address this problem.

Variation, on the other hand, is well handled by our choice of pseudo-takt time and the extra capacity that (if the lean improvement is successful!) it will give us.

Our example value stream shows two clear examples of external constraint. One is obvious—the poor communication of global load data. The answer is communication; the loads data exist, but are hard to find and not actively communicated. The other example is the availability of FEM personnel. This is a more complex issue of prioritizing and scheduling scarce resources at a higher level than this process, but it may have a simple solution. In our example, we propose providing a dedicated person or persons, with associated workstations and software licenses, to this process.

Eliminate Unnecessary or Inefficient Reviews and Approvals

Reviews and approvals are the tasks most frequently deleted outright during LAI member improvement efforts. Again, the issue is what value is created. Reviews can be used to reduce risk. If all they are doing is catching mistakes, they are at best necessary non-value-added quality assurance tasks. If they are used to control iterations and assure convergence on the best solution, they can be key value added steps. In all cases, they should be arranged in so that maximum value is created at minimum cost to the process.

The costs take two major forms: waiting for approvals that are ultimately given, and rework loops when reviews find problems or value-added opportunities. The former is pure waste of waiting, and should be eliminated; the latter may be value added, but must be managed.

A simple example illustrates a technique for streamlining review while preserving or increasing value. Three tasks are followed by reviews in which the task is quality checked and any problems with the task, previous tasks, integration between tasks, or changes in external factors can result in rework. This situation is shown in Figure 6-1. Note the many opportunities for rework to occur and the large number of potential rework paths.

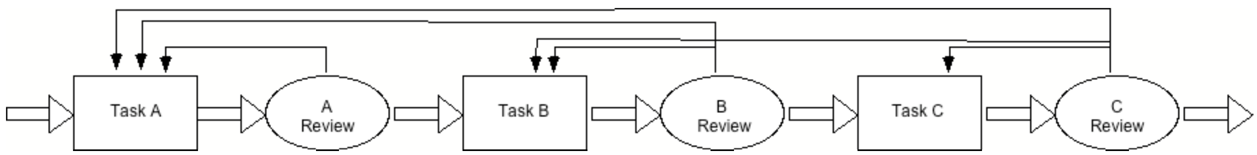


Figure 6-1. Inefficient review cycle

Figure 6-2 illustrates a potentially more efficient review strategy, resembling a local stage-gate. The tasks are carried out with internal quality self-checking. A combined review is carried out to look at issues of integration and changes in external factors. A separate value-based judgment is then made as to the degree of rework that is appropriate to address the concerns brought up in the review. Note that, depending on the process, this may not be the *optimal* review strategy. Although the rework path is simplified, and the probability of rework lowered, any rework results in the entire process being repeated. The point is not dictate any particular review strategy, but rather to encourage you to think about the optimal one *for your process*.

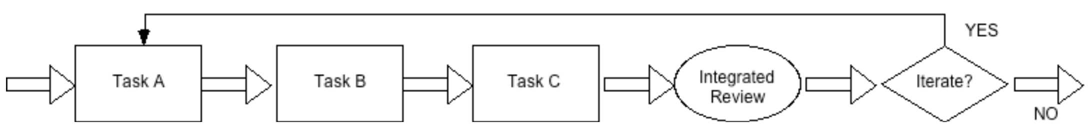


Figure 6-2. Improved review cycle (?)

EXAMPLE

The drawing review has already been identified as NVA. What to do about it will be considered in the next section. The transformation shown in Figures 6-1 and 6-2 also applies almost directly to the review cycle at the end or our process.

On the other hand, we find an opportunity to add value with an informal review of the mechanical design before doing the FEM analysis; waiting until the process is almost concluded to surface manufacturability problems is clearly problematic. A quick review by a team consisting of a designer, manufacturing engineer, and stress analyst is inserted right after the conclusion of mechanical design.

Break Down Monuments

Womack defines a "monument" as any machine or process which is too large to be moved to accommodate dynamic reconfigurations as the value stream changes and whose scale requires operating in a batch and queue mode. Example of monuments are huge presses, centralized paint booths, etc.[68] In *Lean Enterprise Value*, Murman *et al*. expand the definition to include assets, processes, or mindsets that were originally created for a good reason, but which have not adapted to changing circumstances. From the book: "the very institutions, infrastructure, and mindsets that enable success under one set of circumstances can become barriers under a new set of circumstances."[69] Many activities that appear NVA once had an important motivation, but have become impediments under current conditions. Many review or signature requirements fall into this category—they may once have had a legitimate motivation in terms of risk reduction, but do not actual contribute any under current circumstance.

Monuments (according to both of the definitions above) should be eliminated. However, it is important to understand where they came from. Has the functionality of the "monument" activity simply gone away, or does it need to be recreated to fit new circumstances? In PD processes the original motivation was often the mitigation of specific risks. Ask if this risk has become irrelevant due to changing circumstances (e.g., checking for "drawing errors" when digital models are used), is mitigated elsewhere in the process (e.g., self checking for some errors in the digital model tool), or is not mitigated and represents a weakness in the process (e.g. common errors in the digital model that would not show up in a "drawing check" and which are not self-checking!) Make sure the original "value added" intent of eliminated monuments is not lost.

Break down Silos: A subset of monuments, but a very common one, are the silo barriers prevalent in most engineering organizations. Silo barriers can cause delays directly (for example, if work has to enter a new queue every time it crosses a barrier); they can prevent work sharing and balancing (as noted above); and they can prevent cooperative solutions to process problems. They are worst if they are located in parts of the processes that require iteration, as this implies that the silo barrier may need to be crossed many times. Decisions driven by silo management, such as choice of computer tools, can also have major impacts on the ability of workers to efficiently work together.

The usual approach to silo barriers is to form teams to "work around" the barriers. These teams are often in place in an ad-hoc way in legacy processes. Typical past examples include designers and analysts working together informally on design iterations of mechanical hardware, despite having desks in separate buildings and different, incompatible computer tools, or "tiger teams" to do conceptual designs or integrate complex products. Ideally, these teams would be formalized as part of the culture, so that the workers first loyalty would be to their project team, and their rewards and responsibilities primarily under the control of the project manager.

Note however that silo boundaries, as with most monuments, may have some purposes that should not be lost. Quality control is a reason cited for separating designers from analysts, or drafters (or their modern-day scope-user equivalents) from quality checkers. This should not be abandoned; lean methods of self-checking, continuously cross-checking, and mistake proofing should be incorporated to serve the same ends. There are also good reasons for leaving key tasks in the hands of experienced professionals; many engineering disciplines are still at least partly "art." The most value is created when these professionals spend the maximum time at their core function. On the other hand, secondary tasks and tasks on the interface of the silos (data formatting and conversion come immediately to mind) can and should be moved across silo boundaries if necessary.

EXAMPLE

In our example process, the obvious "monument" is the drawing review. The temptation is to simply eliminate it without further thought. The above analysis, however, leads us to consider its purpose—assuring drawing (now 3-D model) quality. Note that the FEM task is complaining of poor quality input, so this purpose is NOT obsolete. Examining the drawing task, the team finds that the modeling software has considerable self-checking capability, but it is used only if the designer is de-bugging known problems. The functionality of the old review will be recaptured by modifying the drawing task to include both running self-checking routines in the software, and completing a manual quality checklist. The checklist will be kept cost-effective by limiting it to a simple paper form that can be completed in a few minutes, and will reside in the process folder. The checklist itself will be monument-proofed by being put under the control of the design manager, who can modify it without management approval.

The silo issues are most clear at the interface between the Mechanical drawing group and the FEM analysis group. There is a major task that currently falls on the FEM group to reformat the drawings for the FEM software. This may be unnecessary; it is certainly a problem that the bottleneck group is doing this work. Also note possible redundancy between the mechanical and moldline drawings; this is not currently on the critical path, but is still a waste.

Eliminate Unnecessary Motion

Unnecessary transportation of information is largely covered in the above sections. Unnecessary motion of people is often associated with either information hunting or "walking the process." The literature, and much practice, overwhelming suggests that the answer to unnecessary motion of people is co-location and shared work experiences. If co-located integrated process teams (IPTs) can be formed as part of your waste elimination effort, excellent. More often larger reorganizations are part of longer-term solutions, so we will consider them in more detail in Chapter 7.

Eliminate Unnecessary Documents and (Re-) Formatting

Sometimes, it will be found that information is created that has no user. This is pure waste and should be eliminated. Much more often, the information will appear to flow from the task creating it to the task using it, but on closer examination will do so very inefficiently. Long, custom-formatted documents are created to transmit a single number. Custom, animated Powerpoint charts are used to get across simple points. These may be simply wastes. As often, they may appear necessary in order to clearly transmit information to reviewers perceived as hostile, or over functional walls to engineers that speak a different jargon, etc. They may also be required in order to archive the process for regulatory or contractual purposes. Rarely, formatting work will actually be value added. The effort saved downstream by, for example, recasting results into a form that is directly useable to a downstream tool, or presenting a decision-maker with a clear choice supported by logically presented reductions of data, may justify non-trivial effort.

The aim in all cases must be to eliminate all non-value-adding documents, formatting, and data handling. If simpler channels (e.g. emails, phone calls, instant messages, collocation) will effectively transmit information they should be consistently used. If documents are actually needed, they should be templated and standardized, such that the necessary information (and only that information) can be dropped in with minimum effort. Two caveats—efforts expended on creation of information exchange systems should be proportional to the expected gains, and they should be regularly updated to avoid becoming "monuments" themselves.

A major issue without a simple answer is the incompatibility between the data structures of numerical tools. Often, considerable effort must be expended to convert data between legacy CAD, analysis, database and/or financial tools. Most simple answers to this problem have proved difficult to implement. Homegrown data conversion routines are often defeated when the tools are updated or new features are exercised. Attempts to implement uniform tool sets across enterprises create cultural resistance and may also cut into real competencies (because the investment in knowledge of specialty tools is considerable) and real capabilities (because generalized tools are sometimes less powerful than specialized ones). For a given process, short term solutions will probably have to be found; for enterprises, this is a continuing challenge.

EXAMPLE

The downstream users of the process outputs do not like the documents they receive, despite all the work done to format them. Templates will be designed *with the downstream users* to allow information to be dropped into them as it is produced. The bottleneck FEM task does not like its input; the mechanical drawing task is assigned the task of pre-formatting the mechanical model output for input into the FEM tools. This will save time on the critical path, while adding some work to a task off the critical path. A longer-term task will be looking into translation tools that would allow direct communication between the drawing and FEM tools. Note that both of these solutions will require relaxation of silo boundaries

Eliminate Unnecessary Analyses, Exploit Underutilized Analyses

The same comments applied to documents above can also be applied to analyses. The purpose of the analyses should be very clear, and the analyses carried out should be those necessary to support the purpose and no more. Typically, analyses are used to eliminate risks. Thus the appropriate filter is to ask what risk is mitigated by the analysis, and what level of risk reduction is appropriate for the stage of design the process is part of. For example, detailed structural analysis is usually required to eliminate the risk of structural failure in the detail design phase; at the conceptual design phase it should only be used if it is required to answer basic questions about the concept feasibility.

Analyses can also represent missed opportunities for value creation. This issue here is not unnecessary analysis, but underutilized analysis. Analyses can be used to improve the design through feedback; if all they are used for is checking that requirements are met this possibility is lost. In the conceptual design phase, were detailed analyses may be inappropriate, "back of the envelope" analysis may be very important for spotting issues that may prove expensive to fix downstream.

The Future State Map

Redraw your value stream with wastes eliminated. By following the advice above, and your own creative solutions to your local problems, you should be able to cut most of the most serious waste, particularly waiting, out of the process. Your solution will be unique to your problem, but will most likely include:

Working to a takt time: A beat that synchronizes the process and eliminates much waiting without requiring complex scheduling.

Visible and explicit information flows: Expectations regarding who will provide what information, to whom, and in what format, will be stated explicitly, to eliminate waiting and information hunting.

Balanced Lines: Resources and work will be balanced such that all tasks can keep up with the takt time. Variations will be minimized and buffered and external constraints to flow cleared.

Streamlined analysis and review processes: Both will be rationalized to assure necessary reduction of risk and uncertainty takes place with minimum effort. Reviews will be managed to assure effective use of iteration.

Minimized formatting: Information will be reformatted only if the reformatting serves the needs of downstream tasks in a cost-effective manner.

Markers for detailed improvements: External constraints that need to be cleared, tasks that need to be improved internally, or data flows that need "fixing" (e.g., with a translation tool) will be identified for further work with bursts.

EXAMPLE

Figure 6-3 shows the future state map for our example process. Key features include:

Working to a takt time: The timeline has been replaced by the proposed one-week takt beat. This takt time can only be maintained with the organization show if the tasks can reduce their cycle times to be at or very close to the In-Process Times.

Visible and explicit information flows: The information flows on the value stream have been rationalized. More importantly, simple information handling systems (e.g. physical job folder plus data exchange web site or server) reduce information hunting and waiting.

Balance the Line: The FEM model tasks have been provided with dependable resources (dedicated personnel) and to the extent currently possible, work has been transferred to other personnel so that they can concentrate on the bottleneck process.

Streamlined analysis and review processes: The final review cycle has been rationalized. The decision to iterate has been separated from the review task. If iteration or rework is required, the first task is to create a plan for the rework, rather than just "sending it back." As importantly, the drawing check has been replaced with a design check, where a group including stress and manufacturing engineers review the design. The hope is to catch design problems early, and avoid large and costly rework loops.

Minimized formatting: The process output is accumulated on a template created to serve the needs of the downstream processes. The reformatting necessary for the FEM model has been moved up to be worked in parallel with other work on week two to clear a bottleneck, but this is far from ideal, and is identified as an area for further work.

Markers for detailed improvements: The formatting problem, issues with bad load data, and the availability of the FEM personnel are all flagged for more work.

This state has eliminated most waiting, and reordered the information flow to alleviate the bottleneck at the FEM model creation. It is dependent on the individual tasks cutting their cycle time to approach their in-process times. Improved information flows and improved personnel availability will help the tasks with this. No actual work is omitted, and in-process times are not assumed to be reduced. The process shown is vulnerable to personnel availability and priorities, as it assumes that the needed personnel are available "on call."

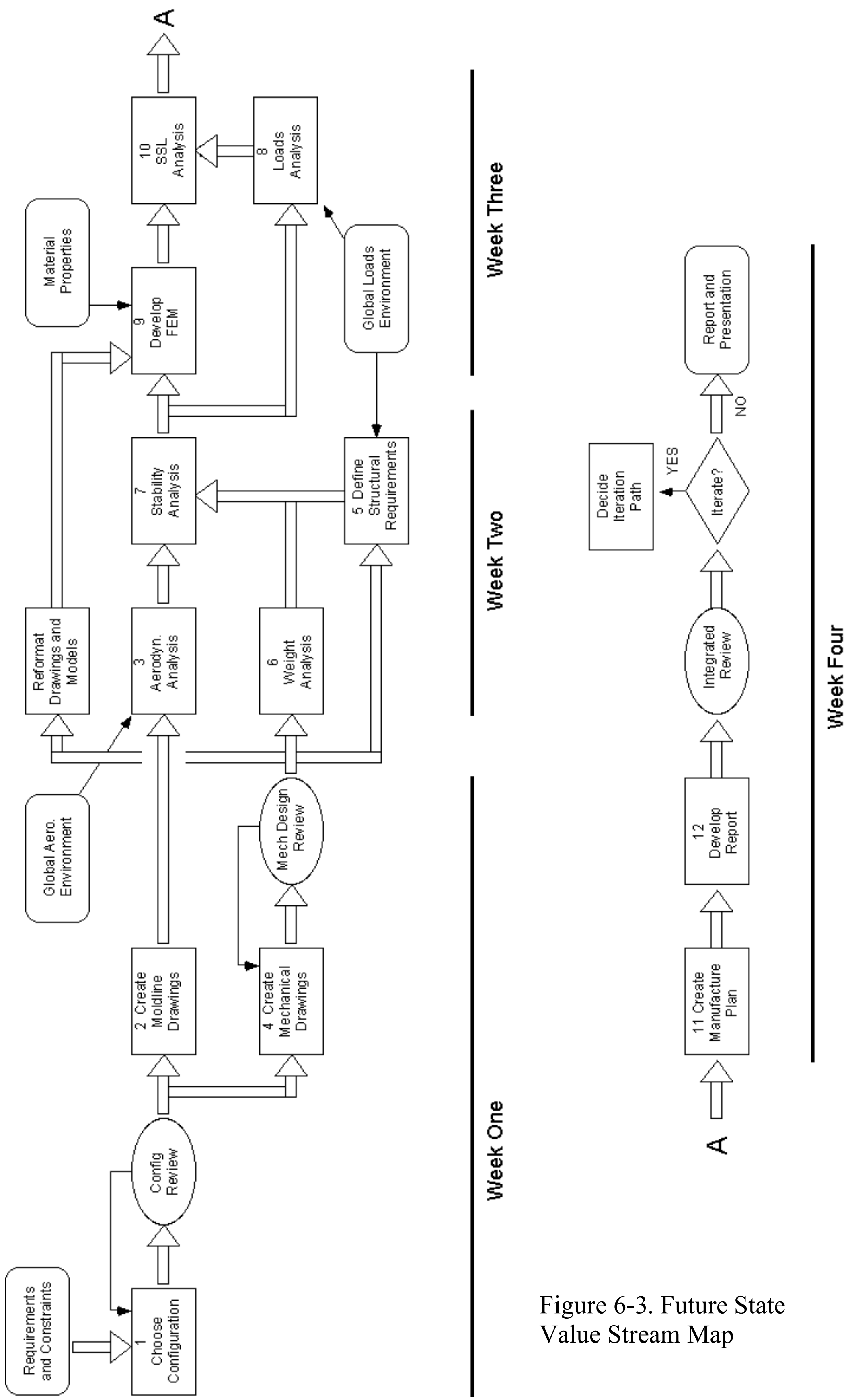


Figure 6-3. Future State Value Stream Map

Summary

A checklist for improving the value stream:

- Identify Waste
  - Waiting
  - Inventory
  - Over-Processing
  - Over-Production
  - Transportation
  - Unnecessary Movement
  - Defective Products

- Establish a Takt Time
  - Average or Psuedo-Takt time
  - Coordination mechanisms
- Assure the Availability of Information
  - Practice Information 6S
  - Collocation and IPTs
  - Make information visual
  - Make information flow physically
  - Pull, don't push, information
- Balance the Line
  - Eliminate bottlenecks
  - Assure the timely availability of resources
  - Reduce and buffer variation
  - Remove external constraints
- Eliminate Unnecessary or Inefficient Reviews and Approvals
- Break Down Monuments
  - Break down silos
- Eliminate Unnecessary Motion
- Eliminate Unnecessary Documents and (Re-)Formatting
- Eliminate Unnecessary Analyses, Exploit Underutilized Analyses

- Draw The Future State Map

Use as many of the above techniques as you need to improve the value stream and meet the improvement goals specified by the key stakeholders (see Chapter 3). Do your best, but remember that the future state is only the first step of the lean journey.

7.0 BEYOND THE FUTURE STATE

Now that you understand your process, and have eliminated (or at least planned the elimination of) the waste in it, you are ready for more radical change. What form this change takes is entirely dependent on your individual process, problems and goals. Therefore, we will only give you general advice in this chapter. Do not fear you will be without guidance, however. There are plenty of experts willing to tell you how to improve your product development process. Now that you have a process you understand, you are ready to listen, and critically assess their advice.

In this section, we will review improvement advice from both the LAI and other sources. The LAI material will be presented in moderate detail, with full references. External sources will be briefly reviewed.

The LEM

The Lean Enterprise Model (LEM) was a product of the first phases of LAI. It is a collection of lean principles and practices, collected from LAI member best practices. The principles and practices are supported by enabling practices, and the entire structure is supported by a very large collection of metrics and research result. Although somewhat dated as a stand-alone product, it has proven surprisingly durable as a set of best practices against which lean improvements can be assessed, and as an organized collection of good ideas which can be browsed when brainstorming new improvement concepts. The basic LEM framework is available to all on line; the supporting database is available only to LAI members.[70]

The LEM's core is formed by twelve Overarching Practices (OAPs). These practices are listed with a brief explanation on the next page. The OAPs remain best practices. Some are "obvious" but are worth reinforcing (*focus on the customer*). Some are worth revisiting often, especially in difficulties (*relationships based on mutual trust*). Some are of clear and direct relevance to PD processes—*assure seamless information flow* clearly applies to the information flows of the PD value stream. Others require some interpretation but are surprisingly powerful—for example, *make decisions at the lowest possible level* is a good guideline for revising a review and oversight structure.

LEM Overarching Practices (OAPs)

1. ***Identify and optimize enterprise flow:*** optimize the flow of products and services, either affecting or within the process, from concept design through point of use.

2. ***Assure seamless information flow:*** provide processes for seamless and timely transfer of and access to pertinent information.

3. ***Optimize capability and utilization of people:*** assure properly trained people are available when needed.

4. ***Make decisions at lowest possible level:*** design the organizational structure and management systems to accelerate and enhance decision making at the point of knowledge, application, and need.

5. ***Implement integrated product and process development:*** create products through an integrated team effort of people and organizations which are knowledgeable of and responsible for all phases of the product's life cycle from concept definition through development, production, deployment, operations, support, and final disposal.

6. ***Develop relationships based on mutual trust and commitment:*** establish stable and on-going cooperative relationships within the extended enterprise, encompassing both customers and suppliers.

7. ***Continuously focus on the customer:*** proactively understand and respond to the needs of the internal and external customers.

8. ***Promote lean leadership at all levels***: align and involve all stakeholders to achieve the enterprise's lean vision.

9. ***Maintain challenge of existing processes:*** ensure a culture and systems that use quantitative measurement and analysis to continuously improve processes.

10. ***Nurture a learning environment:*** provide for the development and growth of both organizations' and individuals' support of attaining lean enterprise goals.

11. ***Ensure process capability and maturation:*** establish and maintain processes capable of consistently designing and producing the key characteristics of the product or service.

12. ***Maximize stability in a changing environment:*** establish strategies to maintain program stability in a changing customer-driven environment.

Millard's Value Stream Analysis Heuristics

Lt. Rich Millard synthesized a set of improvement heuristics using the LEM and the member experience captured in his case studies. They are excerpted from his thesis,[71] slightly modified, below:

Redundancy, Simplification, Standardization involves the self-explanatory confrontation of work that is completed more than once, is more complicated than necessary, or fails to take advantage of potential standardization in the development, documentation, or presentation of information. As part of this category, automation of tasks where it is both viable and beneficial can provide for one method of simplifying and standardizing the overall process.[72]

Flow Continuity attends to the disconnects and misdirections in the flow of the current process. Continuity requires the availability of all input information and tools, and the correct direction of all outputs.

Information Handoffs (or Transfers) stems from the assertion that every time an idea or concept is passed from one party to another, it is impossible for the originator to fully convey to the receiver all the information required to comprehend the idea in the exact same manner. The amount, quality and context of the information suffer with every transfer.

Balanced Review and Responsibility focuses on requiring the correct amount of review to ensure the quality of a product without imparting the unnecessary bureaucracy of signatures and presentations that are not really needed. It is not uncommon for upwards of a third of all process steps to be reviews. Not only is time spent reviewing value rather than creating value, but reviews also require preparing for presentations and reports that often find no direct audience.

Communication Systems can aid in establishing pull systems for information and enable more efficient flow within the process. Web-based environments, common directories for information storage, and electronic transfer of data, can enable efficiencies not possible under many paper-based, push systems. Bear in mind, however, that the effort involved in establishing and learning tools must be justified by results.

Integrated Product and Process Development addresses the opportunity for synergy in a multi-disciplinary team environment. This principle seeks to improve the decisions made about how to design and manufacturing a product through shared communication of available options and capabilities.

Concurrent Processing involves collapsing the process into as high a level of concurrency as necessary when the more sequential process has been improved. This idea serves to reduce cycle time, when necessary, to match relevant business cycles and customer needs.

Collocation, Boundary Objects, and Shared Experiences

Much of the advice in this section is experience-based and anecdotal. There is, however, a subset of practices for which there is strong quantitative evidence. This subset includes co-location and the use of boundary objects and shared experiences to increase the efficiency of information flow.

Co-location: Allen[73] has shown conclusively that co-location greatly increases communication between team members. The definition of co-location is, however, strict—communication falls off dramatically with separation distances measured in yards. Down the hall may not be close enough, and personnel in other buildings, even on the same plant, may as well be in Japan. We have anecdotal evidence that virtual co-location is feasible, but difficult. A multi-university collaborative design study[74] carried out recently used high-bandwidth, multi-mode communications to create a virtual team. Closed circuit and web video, telephone and web audio, netmeeting, websites, email and instant messaging were all used. Still, communication was difficult, and on the order of twice as much time was spent on communications-related tasks than would have been in a true co-location.

Boundary objects:[75] Henderson[76] and Carlile[77] have shown that prototypes and other physical and/or virtual representations of design problems are excellent mechanisms to facilitate team discussion and focus efforts on problem solving. Prototypes help specialists from different groups share their ideas by allowing for the creation of shared languages, based specifically on the models themselves, rather than idiosyncratic references particular to only to one group. More than simply allowing for the validation of design features, therefore, prototypes are important for their role as *communication tools*. Given their ability to help move knowledge between organizational groups and boundaries, some researchers refer to prototypes and similar artifacts as "boundary objects."

Shared experiences: Bernstein[5] showed that the existence of boundary objects was most helpful when engineers from different functional groups worked on them together. This generated a shared experience that was a much better aid to cross-functional communication than the existence of a static boundary object. There is anecdotal but strong evidence that 3-D CAD models, in particular those that allow investigation of cross-functional issues (such as assembly simulations), function as boundary objects. When teams all work on the same model, a shared experience is created. Boundary objects need not be as complex and hard to implement as a 3-D CAD system. Simpler boundary objects (such as the value stream map you are creating!) can be used with the same effect.

A Few Good Books

We will not attempt to review all of the massive literature on PD process improvement here. We will point out again that, with a process that is well understood and cured of pathological inefficiencies, you are in a good position to understand how various methods can be used to further improve your process. You can also make good decisions given the sometimes-contradictory advice that can be found in the literature. Below are a few methods that LAI members have used with some success, and which are particularly compatible with lean-based improvements.

This list is not mandatory reading, and should not be used as an excuse for delay or a source of confusion. Many of these books are written from the perspective of the entire product development process, which is *not* the perspective we are addressing here. Understanding the larger context is not the goal at this point; getting ideas to help you past your particular and unique problems is. The concepts, and in some cases the specific advice, in these books may help you solve problems that your value stream analysis have unearthed.

This list is currently only a sampler, from detailed and specific process improvement tools to grand ideas that present themselves as alternatives to lean. It is intended to grow; please contribute your favorites to later editions by contacting the LAI staff.

Learning to Develop[11] is a wide-ranging look at lean methods applied to product development processes. It includes tools that can be used before this manual (e.g. the waste questionnaire which helps locate the processes to analyze) and short descriptions of many process improvement ideas that can be used in conjunction with this work or after you are done with it. Concurrent engineering, databases, design for excellence, and process failure modes analysis are among the topics covered.

Developing Products in Half the Time[78] and ***Managing the Design Factory***,[79] are collections of good ideas for product development improvement. They are clearly and practically stated.

Building a Project-Driven Enterprise[80] is a new collection of lean-derived ideas for efficient project management, covering a wide range of level of detail, from rules for emails and efficient progress meetings to strategy issues.

Lean Six Sigma[81] links lean and six-sigma concepts. The coupling of lean and six-sigma has proven a powerful combination in factory operations; the applicability to PD process improvement is (to the author's knowledge at the time of writing) untested, but intriguing.

Critical Chain[82] is Goldratt's application of his Theory of Constraints (TOC) to development processes. Knowledgeable users have probably already spotted TOC-based ideas in some of the discussion in this manual. The interaction between lean and TOC is discussed in an interesting forum on the LEI website.[83]

Some Advanced Ideas

Ideas for the total redesign of product development processes are under active development. Implementation of these ideas in member companies is limited, but ongoing. For now, they should be considered by local improvement teams in cases were either a radical change is necessary due to extreme circumstances, and/or the team has a charter to pursue new ideas. In the near future, as more experience and documentation becomes available, these ideas will become more generally applicable.

Lean Product Development Flow (LPDF): This method is an attempt to redesign the product development process from the ground up based on lean principles. It is intended for "smaller design programs based on a high degree of legacy knowledge, with predominantly mature technologies," with short design cycles and with high confidence that major design uncertainties can be avoided. Under these conditions, it is proposed that once the value stream is understood, the detail design process can be radically restructured to eliminate most waste. The suggestions for improving the processes in Chapter 6 of this document are advanced to their logical conclusion by this method. An implementation plan has been published by Oppenheim.[84]

Integrated concurrent engineering (ICE): This method goes by several other acronyms as well. The essence of the method is the linking of disciplinary tools to a central database so that data exchange is truly seamless, *and* the collocation (or virtual collocation) of all involved personnel in a properly equipped design center, so that decisions requiring human interactions (e.g. design changes) can be done quickly. This allows extremely rapid execution of complex, coupled analyses and design trades.[85] It is ideal for preliminary designs, as it allows rapid iterations towards optimal solutions. There is good experience with this method for preliminary design of space systems.[86] It is less clear that the method is suited for detailed or final design, and there are issues in implementation and with the coupling of this new preliminary design method with more traditional detailed design methods.[87]

Multi-Attribute Trade-Space Exploration (MATE): This method is intended as a "front end" process for finding the right designs, and understanding the key trades, at the beginning of a product development process. The method is related to ICE in that analyses are coupled through a database. The analyses are deliberately made simple enough that they can be executed automatically, and a large "trade-space" of hundreds or thousands of potential designs created. The results are evaluated using Multi-Attribute Utility Theory to find optimal design concepts, and to understand the key trade-offs in their most basic forms (e.g., utility vs. cost). MATE results can be used to direct ICE sessions in an integrated process known as MATE-CON.[88] This method has been used successfully for trade-off and feasibility studies.[89]

It is interesting to note, in closing, that a convergence is starting to appear between these "radical" ideas, developed using a clean slate, and the results of a truly lean conventional process. If information flows seamlessly, personnel work in effective integrated product teams, review and decision making is integral and delay-free, and the process moves through a well defined value stream to a takt time beat, the results may be difficult to tell apart from those of a designed-from-scratch environment.

The Ideal State Map

What would a perfect version of your process look like? The point here is not to be bound by practical constraints, resource limits, or current systems. An exercise in stretching our collective imaginations at an LAI PD workshop indicated that a x20 (!) improvement in cycle time might someday be possible. You need to set a very high mark here if you are going to strive for perfection.

EXAMPLE

Figure 7-1 shows *an* ideal state map for our example process. This vision is lifted directly from Millard, pp. 96–97. The preliminary configuration is chosen, perhaps with the aid of a MATE-like trade space exploration, although the simplicity of the part in our example probably precludes that. The specification and analysis steps are done using integrated engineering teams and tools. The aerodynamic and stability analysis is called out separately, as the tools are more difficult to integrate, but the tasks are done concurrently with strong interactions. The entire design is then deliberately iterated to clear up any issues remaining from the review, and improve the design as necessary. The final task is the creation of an integrated "design-to" package, containing not just the preliminary design specifications, but also the analyses and models in a form that is ready for use by downstream processes such as detailed design.

Note that Figure 7-1 looks like a higher level mapping than our previous maps, because each task accomplishes so much more. Figure 7-2 (from Millard[90]) shows an expanded look at what the integrated teams actually do; ultimately they cover the same ground as in the old process, but with a cycle time one could imagine approaching 2 weeks.

Finally, note that there are actually many paths to this state. The integrated engineering could be the result of traditional tools, co-location, and a team approach; it could be the result of an ICE approach using existing or new tools, or it could be enabled by a new, integrated toolset incorporating design, FEM modeling, weight estimating, and manufacturing planning and simulation.

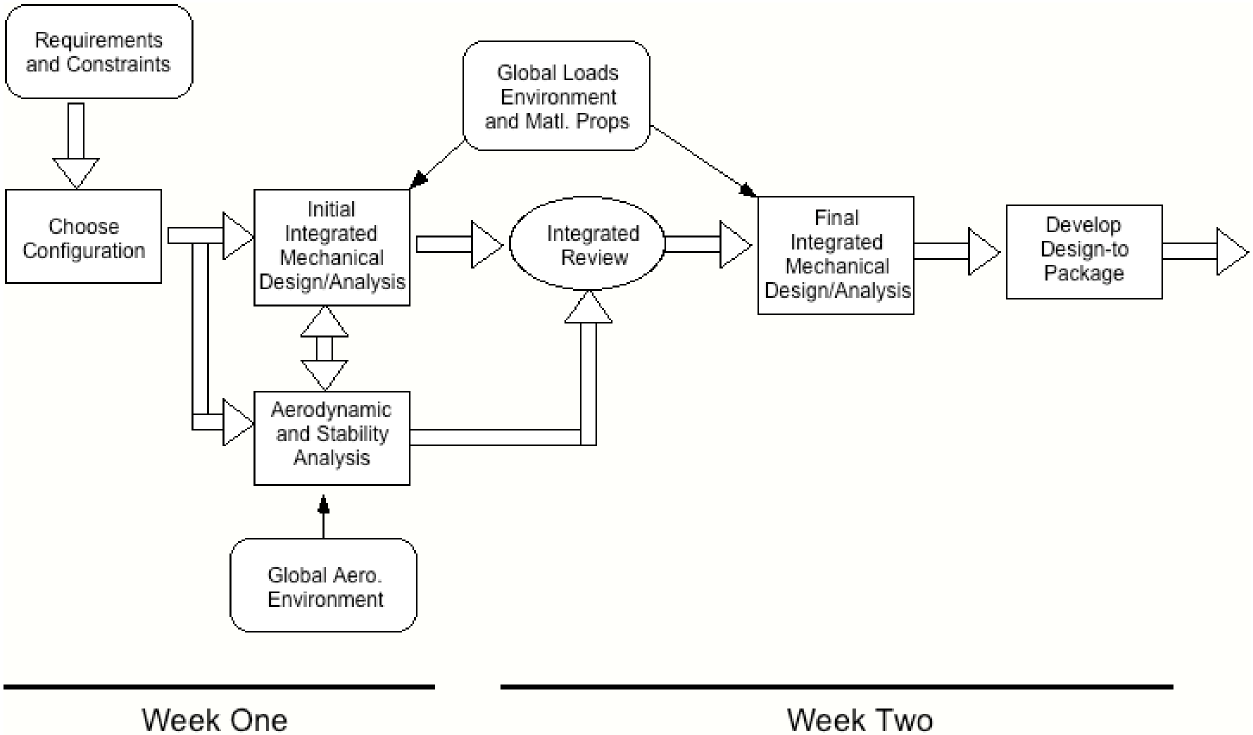


Figure 7-1. Ideal State Map

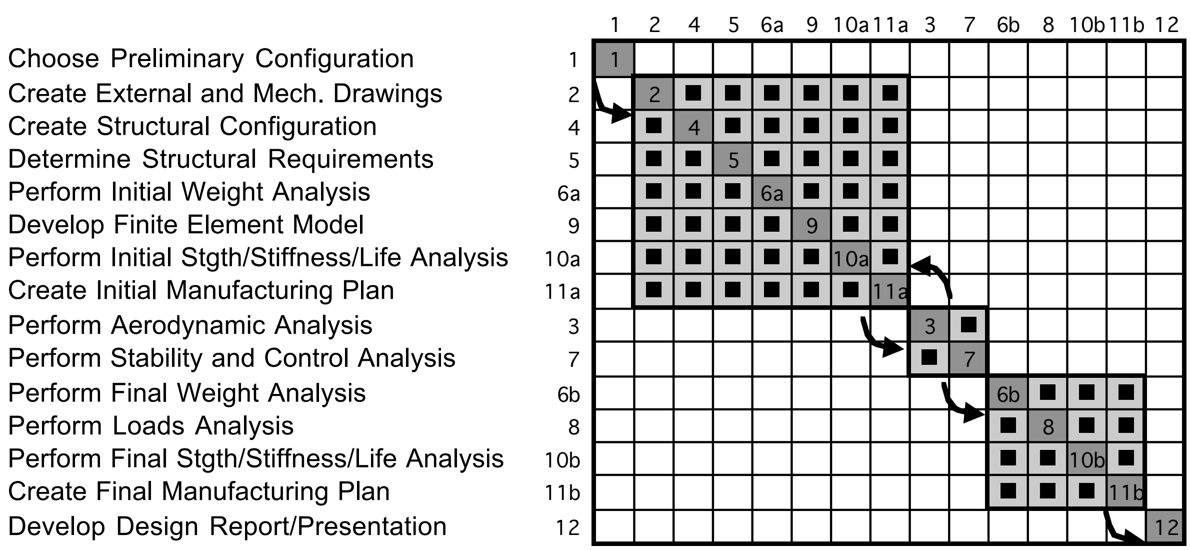


Figure 7-2. Possible expansion of Ideal State Map in DSM form

8.0 STRIVING FOR PERFECTION

The drawing of the ideal state map should not mark the end of your consideration of the value stream. The lean journey is a continuous one, and the value stream map is a vital tool throughout the journey. Ideally, the value stream map will remain a "living document," updated as a continuous record of your progress to lean.

The creation of a continuous-improvement culture is beyond the scope of this little manual, but here are some ideas on how your VSM can be used to foster continuous improvements.

Use the VSM as a visual control for longer-term improvements: Most *kaizen*-type events conclude with some action items for longer-term improvements. Failing to actually carry these out is common, and a leading cause of failure of improvement efforts. Use the Future State VSM as a visual control. Keep it visible, and get team commitment to achieve the future state in a finite time. Track progress visually by noting the progress of the updated current state towards the future state. Use persistent differences to diagnose implementation problems, and follow up.

Use a multi-kaizen strategy: It is an irony that the word "kaizen" is used in the US in a sense almost opposite to its Japanese meaning. Improvements are concentrated in rapid, work-interrupting events, with relatively weak follow-up. In most US work cultures, however, it is more effective to get people's attention with concentrated events than to hope they will fit continuous improvements into their busy day-to-day schedules. An effective compromise used in member companies is to schedule multiple *kaizens* on the same process. Sometimes this is planned from the start; other times it emerges as necessary. For example, the VSM developed in the first event may reveal the need for collecting real-time process data before an improvement strategy can be devised. The development and improvement of the VSM can serve as a tool for both decomposing the problem into multiple events, and to tying multiple events back together into a logical whole.

Use the VSM as a process control tool: Once you understand it, use it. A VSM is an excellent way to introduce a process to new participants. It can be used as a visual control tool, tracking the progress of various jobs through the process. Bottlenecks, pileups, missing resources, etc., can be marked on the VSM in real time, and their consequences and urgencies understood at a glance. A shared vision of the VSM can allow rapid adaptation of the process to shifting needs and priorities. It can also allow quick understanding of the effects of on-the-fly adaptations; for example, the handling of urgent work and its effect on other jobs.

Used in these ways, the VSM can be a useful process management tool as well as a motivator and enabler of continuous improvement.

APPENDIX A: METHODS AND EFFECTIVENESS

This appendix is a verbatim excerpt of two sections of a paper by McManus and Millard.[52]

Research Design

Visits were made to nine major U.S. aerospace development sites to determine the current practices and maturity of VSA/M within the industry. These visits took place from January to August 2000. The research methodology involved collection of data by several means, including interviews, participatory research, workshops, presentations, and both formal and informal discussions concerning the topic. Each of the nine sites was engaged in a PD process improvement effort. A total of 31 interviews were conducted; and a total of 48 subjects participated in the study. The data collection focused on the three themes:

- The process mapping and VSA/M tools used at each of the sites
- The context surrounding their use
- The self-assessed success of the respective process improvement efforts

The collected data was reduced as follows. The various methods used were evaluated for their ability to describe process characteristics, including:

- Time: concurrency, task duration, and start/stop times
- Work: decision branching, feedback, flow, inputs/outputs, iteration, metrics, task precedence, resources, tasks, and value
- Structure: geography, grouping/teaming, milestones, and organizations

The evaluations were summarized as ratings of the ability of a method to: 1) represent a process, and 2) assist in the analysis of the process for improvement. Both the basic methods, and the methods as used at the study sites (which were usually modified) were rated in this way.

The context information was reduced to a rating of "lean context" by considering the following factors:

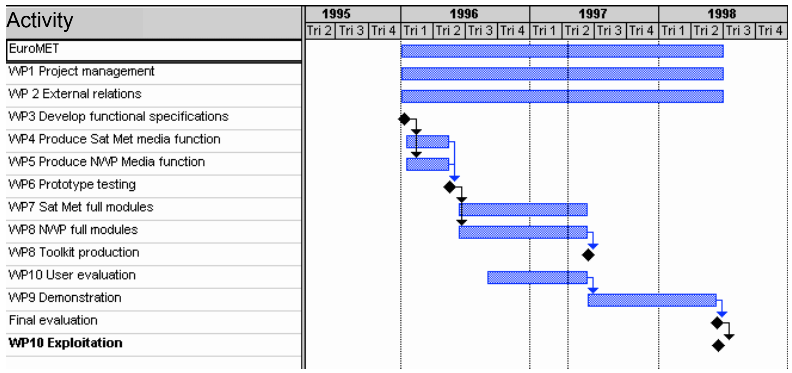
- Opportunity for lean education and training
- General resource allocation for business improvement efforts
- Leadership involvement in business improvement efforts
- Organizational integration of lean principles
- Lean vision or goal

Results: Available Tools

A number of existing process analysis tools will be briefly reviewed here, some of which were not originally intended for use in the mapping of a value stream. This list is not intended to be exhaustive, but rather reflects the tools actually found in use in the U.S. aerospace industry.

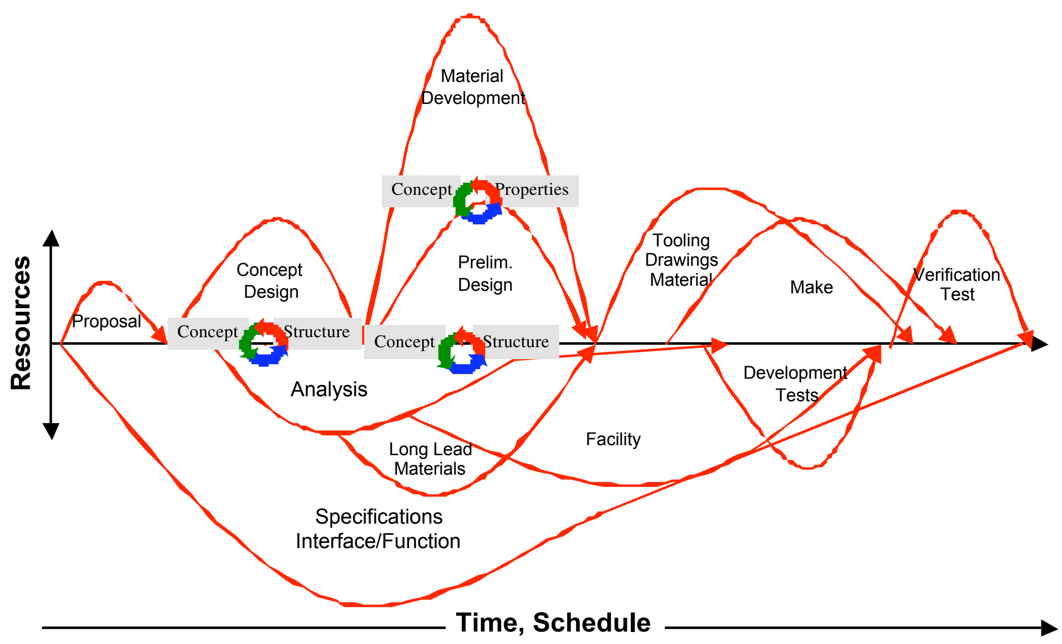
Gantt Chart

The venerable Gantt chart is a traditional method for displaying sequence, schedule, and dependency between tasks. It is widely used in the PD community for display of schedule and milestone information.



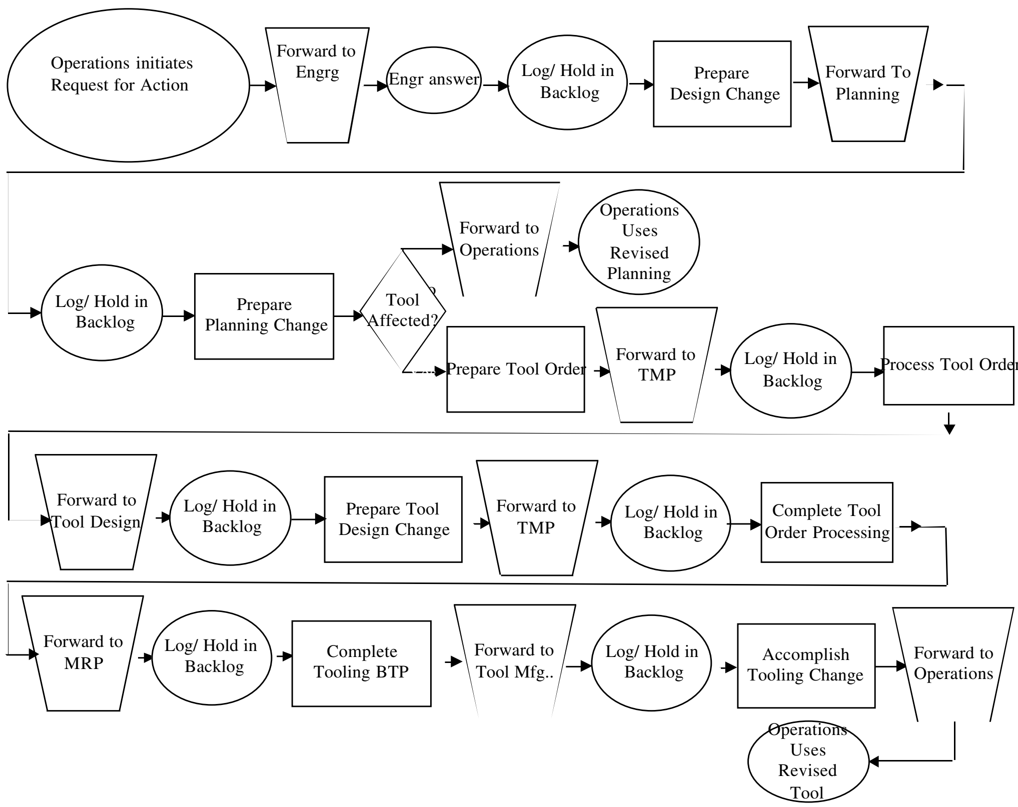
Ward/LEI Map

Alan Ward of Ward Synthesis, Inc, has advanced a map that highlights the concurrent and cyclic nature of PD processes. The map shows time along the horizontal axis, and the magnitude of resources required to perform each task on the vertical axis. Overlapping curves are used to illustrate the time and resources required for each activity.



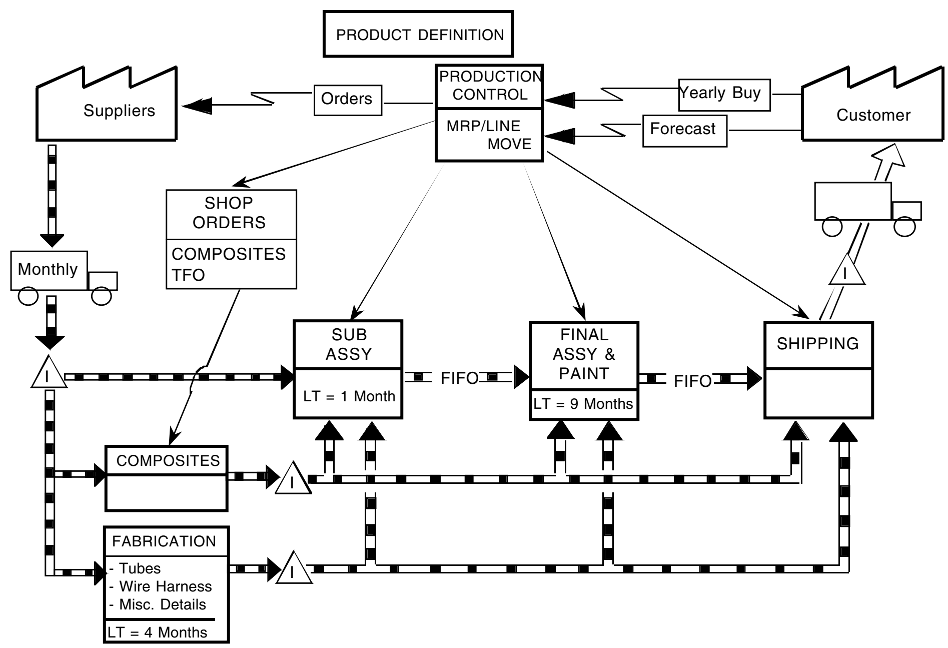
Process Flow Map

The application of lean ideas using traditional process mapping tools has also been developed through works such as Trischler's *Understanding and Applying Value-Added Assessment*.[9] Standard symbols are connected by arrows to describe flow, and color can be utilized to denote value-added versus non-value-added assessment.



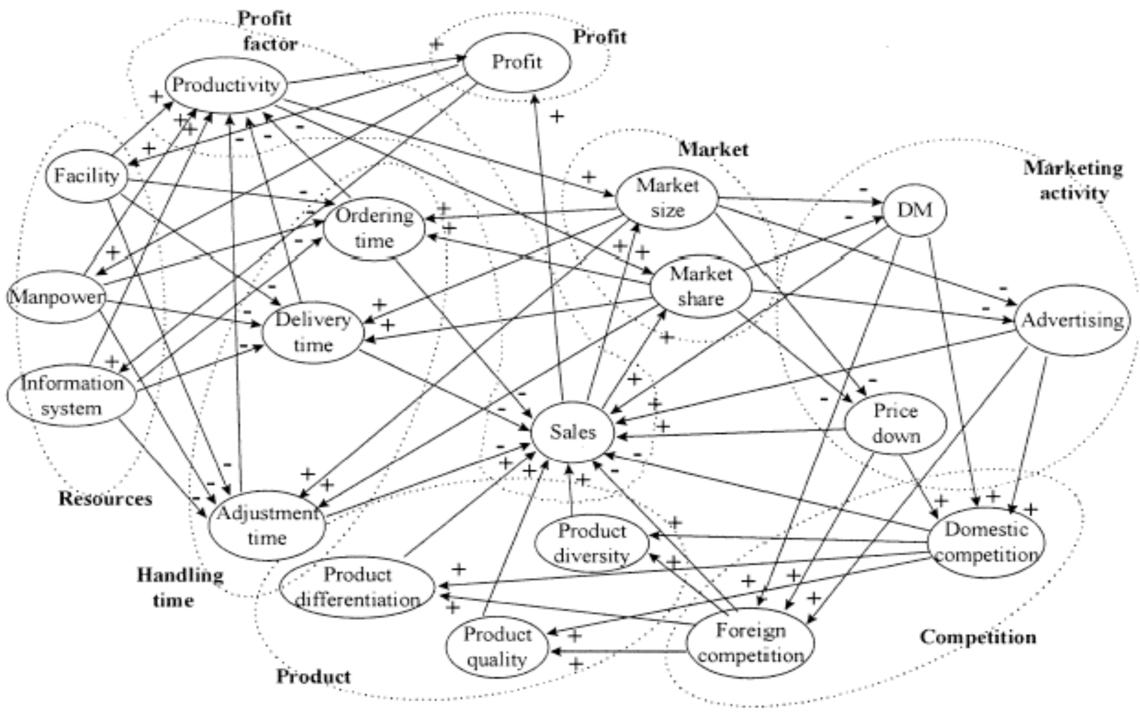
Learning to See

Rother and Shook's *Learning To See* method, based on factory floor mapping, can be adapted to PD processes once an understanding of how lean concepts, like flow and waste, translate to PD activities. *Learning To See* provides the most proven tool for lean-based VSA/M to date.



System Dynamics

Although not a VSM method, *per se*, system dynamics modeling (see, for example, Sterman[91]) can be accomplished in the context of a lean improvement exercise.



Design Structure Matrix (DSM)

The DSM technique is a well-developed method for analyzing the sequences of, and information flows between, the tasks in a process. An "n-squared" matrix is used to depict the information flows from one task to another. The matrix can be numerically optimized to minimize iterations and maximize the potential for concurrent work. A complete description of the method, with tutorials for its use, can be found at the MIT-DSM website.[10] A sophisticated extension of this technique can also be found in Browning.[2]

	A	D	B	J	H	C	K	G	M	E	L	I	F
Customer Requirements A	A												
System Level Parameters D	■	D											
Wheel Torque B		■	B										
Piston—Front Size J		■	■	J	■								
Piston—Rear Size H		■	■	■	H								
Pedal Mechanical Advantage C	■	■		■	■	C			■	■			
Rear Lining Coeff. of Friction K		■		■	■	■	K		■	■			
Front Lining Coeff. of Friction G		■		■	■	■		G	■	■			
Booster Reaction Ratio M		■	■	■	■	■	■	■	M	■			
Rotor Diameter E	■	■	■	■	■	■	■	■	■	E			
Booster—Max. Stroke L									■		L		
Caliper Compliance I		■		■		■			■			I	
ABS Modulator Display F												■	F

Evaluations

The evaluations of the mapping methods used revealed a variety of strengths and weaknesses inherent in each method. The detailed analysis and results are in Reference 1. The Gantt and Ward/LEI methods proved the most useful for representing process characteristics. The process mapping, DSM, and Rother and Shook maps gave the user the most versatility in analyzing and changing the processes mapped.

Table A-1. VSM Tool Characterization Matrix

Process Attribute	**Gantt**	**Process Flow**	**DSM**	**Learning To See**	**System Dynamics**	**Ward/LEI**
concurrency	✓		✓		✓	✓
start/stop times	✓		(✓)			✓
task duration	✓					✓
decision branching		✓				
feedback			✓		✓	
flow:						
“product” info.		✓	✓			
command info.		✓		✓		
material		✓		✓		
inputs/outputs		✓		✓	✓	
iteration			✓		✓	✓
metrics		(✓)		(✓)		(✓)
task precedence	✓	✓	✓		✓	✓
resources:						
generalized						✓
specific	(✓)	(✓)	(✓)	(✓)	(✓)	
tasks	✓	✓	✓	✓		✓
value		(✓)		(✓)		(✓)
geography		(✓)		✓		
grouping/teaming			✓			
milestones	✓					✓
organizations			✓	(✓)		

The capabilities of the tools as used at the various sites were normalized to a rating of relative capability. The ratings went from 0 to 1; where 0 is useless and 1 as good as any tool considered.

The as-used tool capability was compared to the lean context rating and the self-assessed metric of success, with interesting results. Figures A-1 through A-3 show the correlations of tool capability and success, lean context and success, and tool capability and lean context. All factors are correlated. This confounds the effect of tool capability on process improvement success, but strongly suggests that good tools and lean context are both necessary (or at least correlate with) process improvement success.

No single best practice was identified, but the highest as-used capability ratings went to sites using process flow maps, DSM analysis, or combinations of the two.

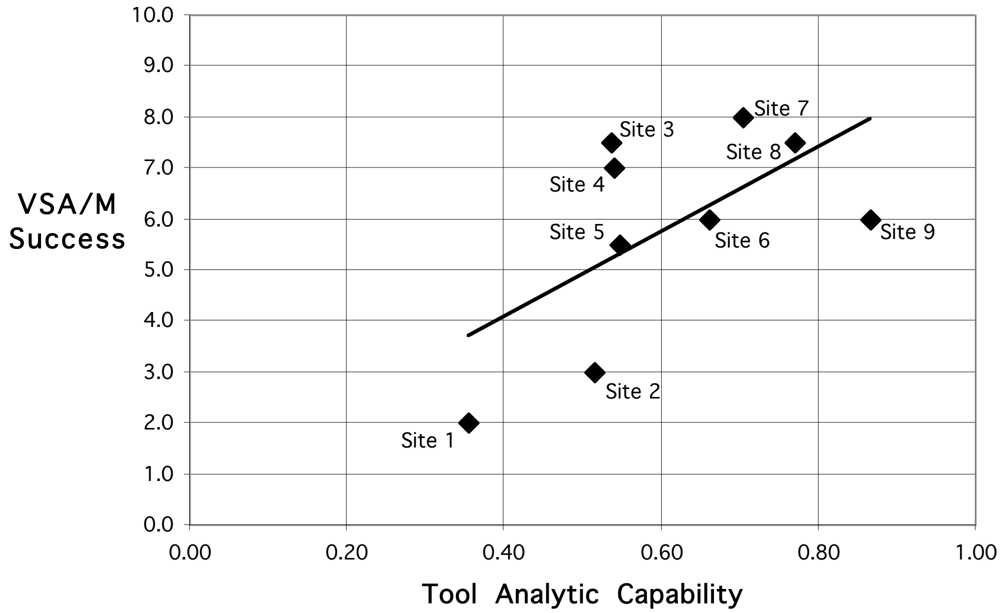


Figure A-1. Success-Capability Correlation

The correlation of context to success showed a stronger relationship. Figure A-2 illustrates the relationship below, which exhibits an r^2 value of 0.572.

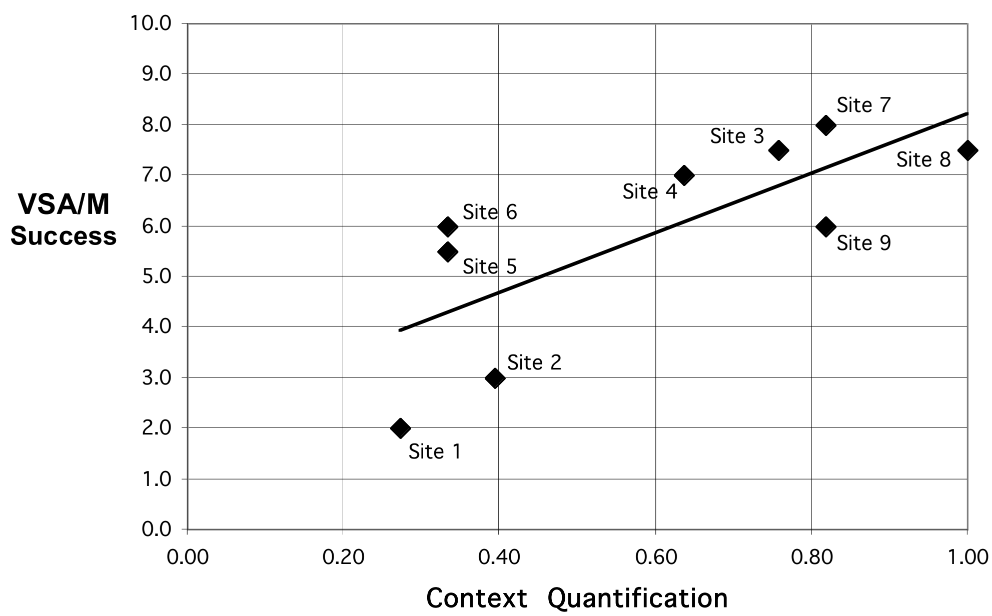


Figure A-2. Success-Context Correlation

Figure A-3 shows the relationship between context and tool capability. This relationship also shows a strong correlation with an r^2 value of 0.495.

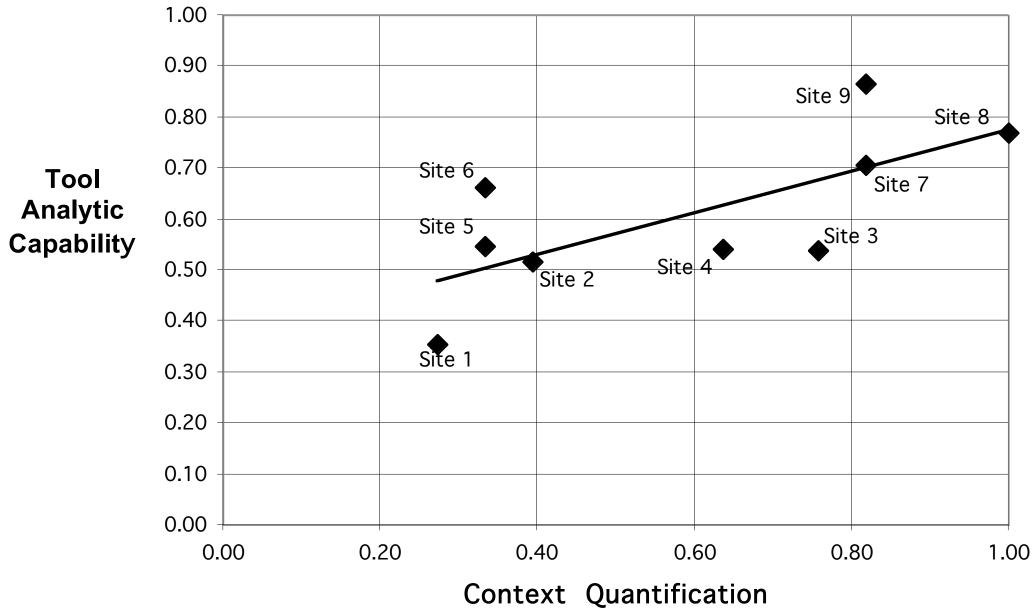


Figure A-3. Capability-Context Correlation

Concluding Remarks

Value stream mapping can be an important tool for product development process improvement. Aerospace industry experience indicates that both a good set of value stream mapping tools and a lean context correlate with process improvement success. No single best practice was found. However, lean terminology reassessed to apply to PD, combined with several complementary value stream mapping methods, provide an effective set of tools for PD value stream mapping.

APPENDIX B: SAMPLE DATA COLLECTION FORM

This form is taken directly from Millard. It is intended to be a single, double-sided sheet that is used by the PDVSM team on the floor. It is not necessarily a recommended template (some problems with collect the sort of data on it are recounted in the example) but it could serve as a starting point for your own intelligent adaptation.

Value Stream Analysis and Mapping Data Collection Sheet

General		Resources		
Activity Name		Elapsed Time		(days)
Location		In-process Time		(hrs)
Pers./Org. Performing		Core Task Work Time		(hrs)
Completion Criteria		Activity Based Cost		
Success Criteria		Special Resources Req.		
Other:		Chance of Rework/Time	%	(hrs)

Input #1		Input #2		Input #3	
Name		Name		Name	
Sender		Sender		Sender	
Transfer		Transfer		Transfer	
Quality	1 2 3 4 5 N/A	Quality	1 2 3 4 5 N/A	Quality	1 2 3 4 5 N/A
Utility	1 2 3 4 5 N/A	Utility	1 2 3 4 5 N/A	Utility	1 2 3 4 5 N/A
Format	1 2 3 4 5 N/A	Format	1 2 3 4 5 N/A	Format	1 2 3 4 5 N/A

Output #1		Output #2		Output #3	
Name		Name		Name	
Receiver		Receiver		Receiver	
Transfer		Transfer		Transfer	
Purpose		Purpose		Purpose	

Critical Drivers

(metrics/attributes)

Context (interaction with other VS)

Value

Non-Value-Added Enabling Value-Added

1- - - - - - - - - 2 - - - - - - - - - 3 - - - - - - - - - 4 - - - - - - - - - 5

Functional Perform.	1 2 3 4 5 N/A	Enabling Activities	1 2 3 4 5 N/A
Defn. of Processes	1 2 3 4 5 N/A	Cost/Schedule Savings	1 2 3 4 5 N/A
Reduction of Risk	1 2 3 4 5 N/A	Other:	1 2 3 4 5 N/A
Form of Output	1 2 3 4 5 N/A	Other:	1 2 3 4 5 N/A

Waste Sources

Waste of Resources	
Waste of Time	
Waste of Quality	
Waste of Opportunity	
Information Waste	
Other:	

Comments/Suggestions

(improvement ideas, problems, stress points)

Data Collection Sheet Legend	
Elapsed Time: days from authorization to proceed, to the completion of the activity **In-process Time**: hours of active work, as measured, for example, by time charged **Core Task Work Time**: time when core task is being worked, excluding setup, data retrieval, etc.)	**Special Resources Required:** any personnel, tools, or information that may distinguish the activity or provide constraint **Chance of Rework/Time:** percent chance of rework being required for (or because of) the activity, and the time associated with that rework
Input Criteria	
Quality 5 - Significantly more information than needed 4 - More information than needed 3 - Quality is just right 2 - Information is missing 1 - Information is inaccurate and/or untrustworthy	**Formatting** 5 - Ideal formatting for immediate use 4 - Fairly good formatting 3 - Acceptable formatting 2 - Some reformatting necessary 1 - Reformatting necessary
Utility 5 - Direct and critical contribution 4 - Important contribution 3 - Beneficial contribution 2 - Indirect contribution 1 - No contribution	**Transfer:** the method of transfer by which the input arrives to the activity **Output Purpose:** the product that the output is contributing to, or the goal of the activity
Critical Drivers: **metrics that reveal the distinguishing nature and critical drivers of the process**	
Context: **interaction with other Value Streams (such as manufacturing and R&D), and any authority/review issues**	
Value Criteria	
Functional Performance (FP) Functional performance of the end product to be delivered to the customer 5 - Direct specification of major FP parameters 4 - Direct specification of FP parameters 3 - Direct specification of minor FP parameters 2 - Indirect specification of FP parameters 1 - Possible specification of FP parameters	**Form of Output** The form of the output of this task (e.g. report, spreadsheet, build-to-package, etc.) 5 - Flows easily into program milestone 4 - Flows into milestone with some changes 3 - Flows easily into downstream task 2 - Flows into next task with some changes 1 - Flows into next task with major changes
Definition of Processes Definition of processes necessary to deliver the end product to the customer 5 - Direct specification of major downstream processes 4 - Direct specification of downstream processes 3 - Direct specification of minor downstream processes 2 - Indirect specification of downstream processes 1 - Possible specification of downstream processes	**Enabling Activities** Enabling other activities to occur (e.g., the other activity is required for completion of program) 5 - Major checkpoint preventing further work 4 - Moderate checkpoint in program 3 - Task necessary for continued work 2 - Necessary, but not especially time-sensitive 1 - Necessary, but not time sensitive
Reduction of Risk Reduction of risks and uncertainties associated with functional, process, or market areas 5 - Major risks greatly reduced or eliminated 4 - Significant reduction of risks 3 - Minor reduction of risks 2 - Indirect reduction of risks 1 - Possible reduction of risks	**Cost/Schedule Savings** Cost and/or schedule savings resulting from task execution (i.e., a core competency) 5 - Recognized as a core competency 4 - Major improvement over historical predecessor 3 - Improvement over historical predecessor 2 - Minor improvement over predecessor 1 - Possible improvement over predecessor
Waste Sources	
Waste of Resources: possible misuse or non-optimization of resources **Waste of Time:** possible cause for delays, waiting, unplanned rework **Waste of Quality:** possible cause for lack of quality, errors, defects **Waste of Opportunity:** possible oversight of personnel, tool, or technology potential **Info Waste:** overproduction, inventory, transportation, unnecessary movement, over-processing, transfers, scatter	

APPENDIX C: SECOND EXAMPLE PDVSM

The following map was created during a Lean Now[92] event at a member company. The actual data and names have been omitted. Pseudo-data that are representative of the kind of data collected during the event are shown on the map. The pseudo-data also illustrate how some of the problems uncovered at the event were discovered through examination of the *data*, not just the value stream.

The event was held to improve the process of detailed engineering and drawing release for major structural components of an aerospace vehicle. This process was selected for improvement having been identified as a bottleneck during an earlier enterprise value stream mapping exercise. Lean Now provided the context for the event, including a well-defined process, subject matter experts to facilitate and provide training, and management and external customer support and participation. Training was provided as described in the "Training the Team" section of Chapter 3, using the Alpha version of this manual, short lectures, and exercises based on the Lean Enterprise Value simulation.

The key stakeholders were the end customer (government program management) and the prime contractor firm. Their desire to implement rapid spiral development of the vehicle required a roughly 50% improvement in the detailed engineering and drawing release cycle time, as well as cost savings. The immediate customer for the drawings were the manufacturing sub-contractors, who needed most of the drawings on time, and some well ahead of time to allow long-lead time purchases and other work to commence.

Two teams were assembled. They consisted of subject matter experts, representatives of the end customer, supervisor level process participants, process workers, and representatives of quality, check, and other groups that interacted with the process. One of the teams had members from the immediate customer (the manufacturing subcontractor who received the drawings), while the other did not.

The value was determined to be process completion at the right time (defined in this case as considerably faster than current practice), with perfect end product quality (structural failures are not acceptable) and good process output quality determined by the output user (i.e., the drawings had to be suitable for use by the manufacturing subcontractor).

Figures C-1 through C-3 show the map assembled by one of the teams. The figures show the map broken up into three phases; the map is best viewed by pasting the figures together to make one large map. Key to the success of this mapping effort was the participation of a supervising engineer who had a complete, detailed picture of how the process worked. Other team members verified and completed the map. Phone calls or short visits to personnel not participating in the event were used to fill in details. Note the use of "swimming pool lanes" (see key on Figure C-3), which divide the map vertically into functions. This technique was very useful for the initial arrangement of the map, which otherwise may have looked very chaotic. The pool lanes also correspond to important functional boundaries; in particular, the "check" function in the review lane is in a different organization from design and stress, complicating attempts at coordination.

Much of the data shown on the map are estimates based on the knowledge of the team members. As each drawing required a different level of effort, best/worst ranges and an average were collected and displayed in data boxes attached to many tasks. Several team members were from a process quality organization, and were able—with some difficulty—to extract data from their databases. It did not cover individual tasks, but did provide hard data on the total time elapsed during phases of the process. This data is displayed on the timeline across the bottom of the map. Also added to the timelines are “soft” data, consisting of roll-ups of the estimated average task times. The later were adjusted (only slightly, as it turned out) to match the hard data that was available. Also collected, although not shown for clarity, was capacity data, expressed as number of available workers. It helped explain the long wait times in front of the stress tasks—stress had, at least initially, insufficient capacity.

The hard data proved key to understanding the problems of the value stream. It confirmed that the design phase, shown in the first third or so of the map, was running on schedule. It also confirmed that the stress and check phase (the middle of the value stream, where it gets most confusing) was taking much longer than anticipated. It also provided one surprise: the final phase, review and release, was *not* taking excessively long, despite the misgivings of some of the participants.

The bursts on the figures summarize the evaluation of the value stream. During the design phase, although the design itself tended to go smoothly, several other things didn’t. The pre-release to the manufacturing subcontractor, and the review with them, was of unknown value due to the lack of a team member from the subcontractor. This was a failure of the value stream mapping effort, and may have masked an actual problem. The fact that stress very rarely took the opportunity to start long-lead work on finite element method (FEM) models added to the problems in the stress and check phase. That phase was plagued by long waits and slow model building in stress, and a confusing and inefficient back-and-forth to check. To add to these problems, stress often did not have the material and load data it needed, and at least occasionally detected a serious problem with designs late in the process, setting off a huge rework loop. The release phase was not a critical driver of cycle time, but could be better.

A future state map was created which met the needs of the event. It established a 20 unit takt time based on the hard data for the design phases; moved FEM modeling and material data and loads acquisition up to run in parallel with the second half of design, and eliminated waiting, excessive reviews, and excessive local iterations. This, along with adequate staffing, should eliminate the long cycle time for the stress and check phase. The theoretical result was a 40% reduction in total cycle time without removing any tasks, within reach of the 50% reduction goal. Unaddressed was the large rework loop and the relations with the manufacturing subcontractor. These were slated for future work, as was a local event for improvement of the review and release phase.

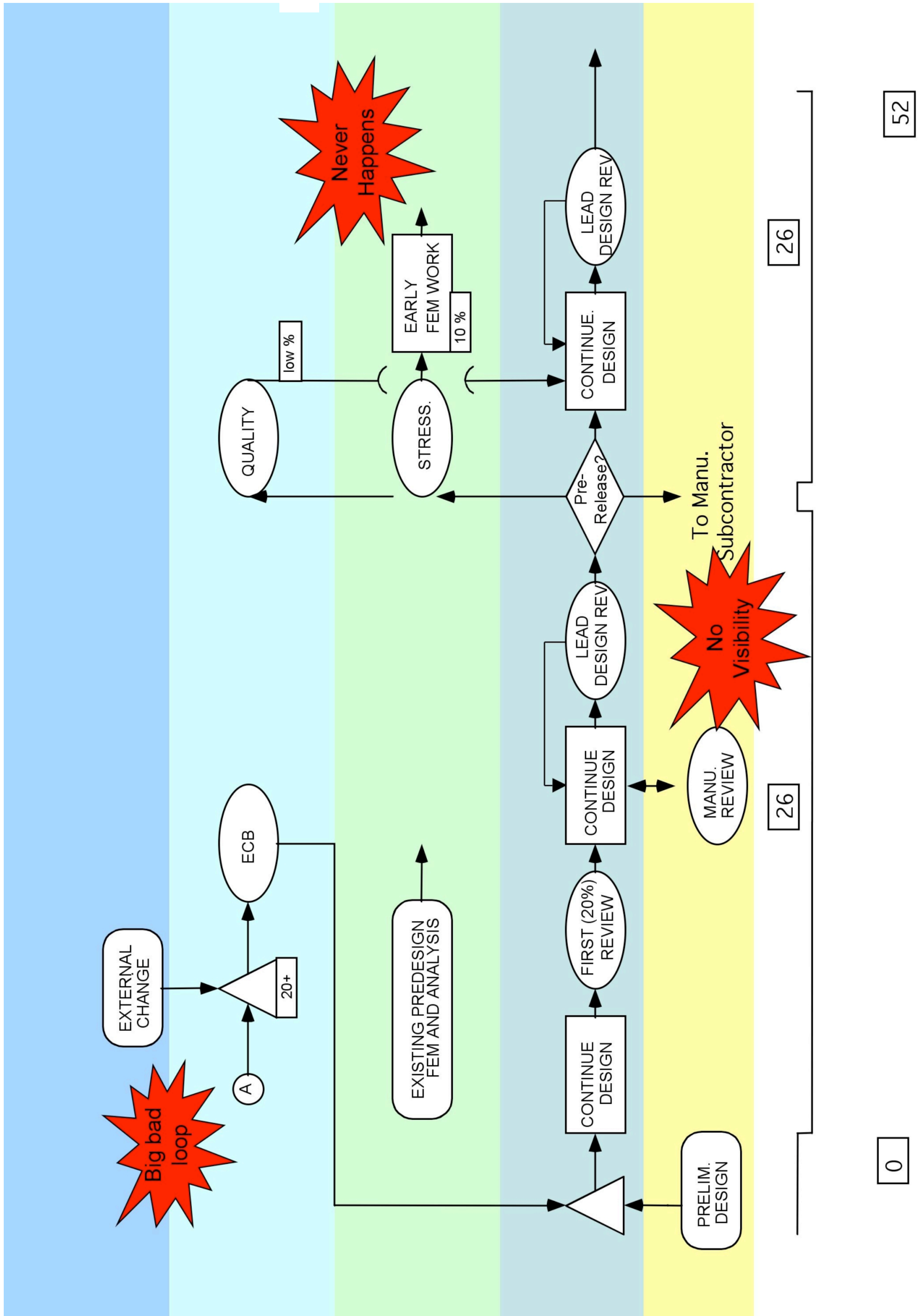


Figure C-1. Design phase

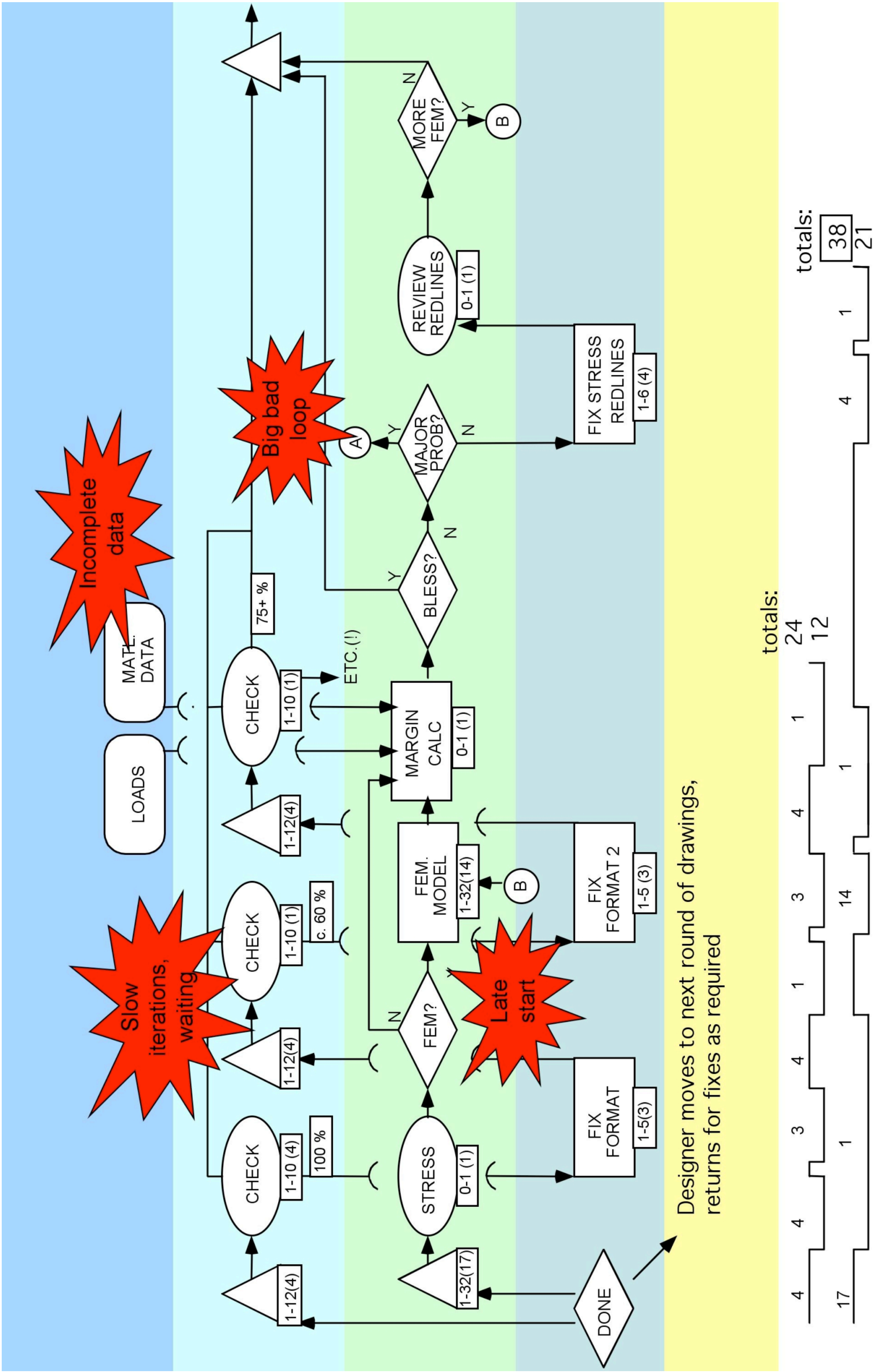


Figure C-2. Stress/Check phase

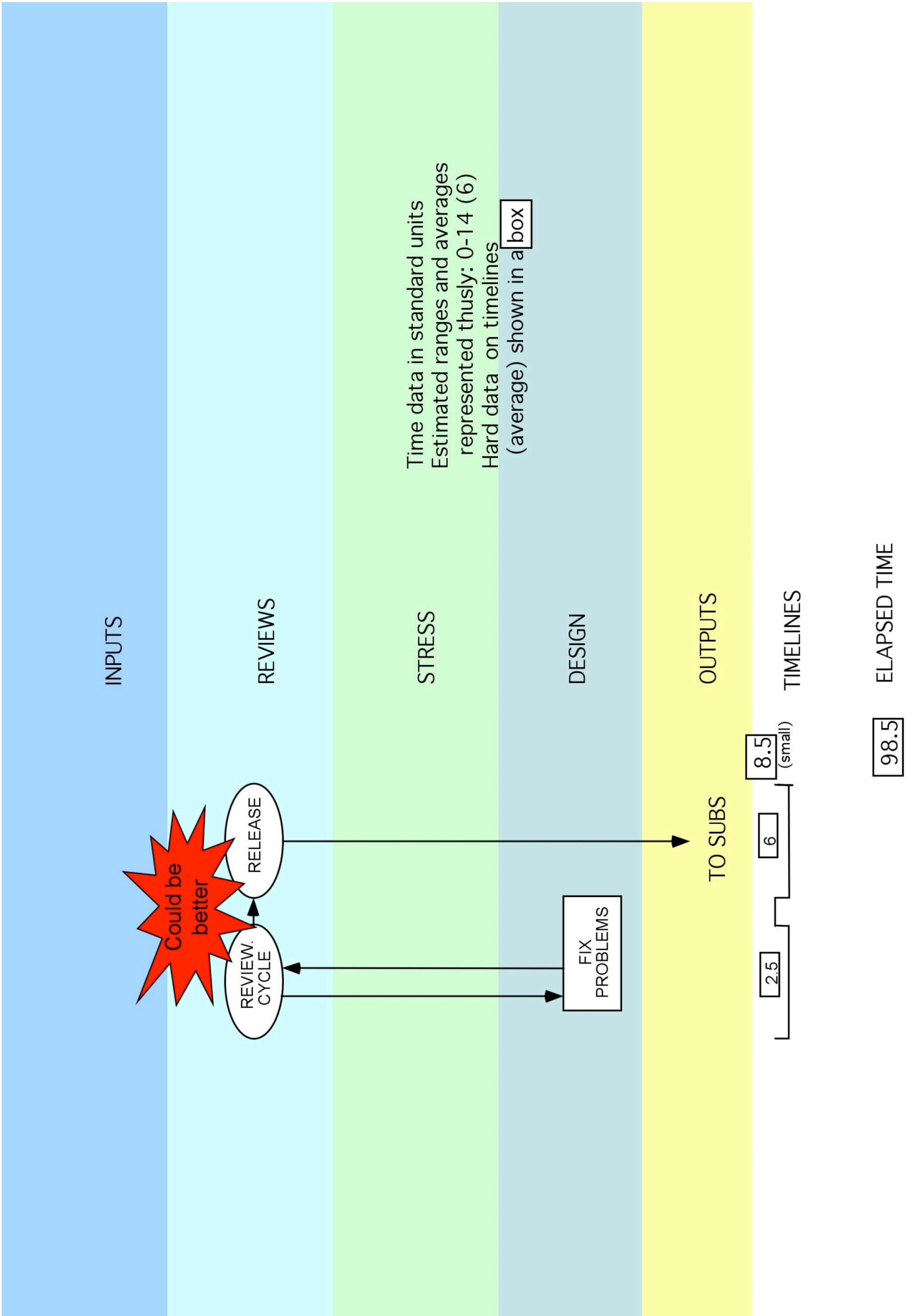


Figure C-3. Review and Release phase

APPENDIX D: ADDITIONAL MAPPING SYMBOLS

Whitaker[47] used a PDVSM method based on the beta version of this work and the work of Morgan[93] to map several real value streams. He found some difficulties in capturing all of the types of interactions using standard symbols. This Appendix illustrates a few of the innovations he used to capture different kinds of PDVSM interactions. All of the examples here are taken from swim-lane VSMs (see Appendix D above).

In Figure D-1, a portion of a high-level swim-lane VSM is shown. Several interesting features are evident. Two of the process boxes, Prelim. Design and Update Reqts, are shown elongated in the style of a Gantt chart to show that they span several months. They proceed in parallel, with a great deal of back-and-forth information flow. This is shown by the small "loop" symbol spanning them. Near the end of the period shown, a "gate" event occurs (PDR) and both it and the preparation for it are shown as vertical bars. This shows that they have some dimension in time, and cut across all functions. There are also two external events shown as labeled vertical line—they have no time dimension, but affect all the functions. EDM Deliverable was a deadline, and Plan B Announced was an external disruption to the program.

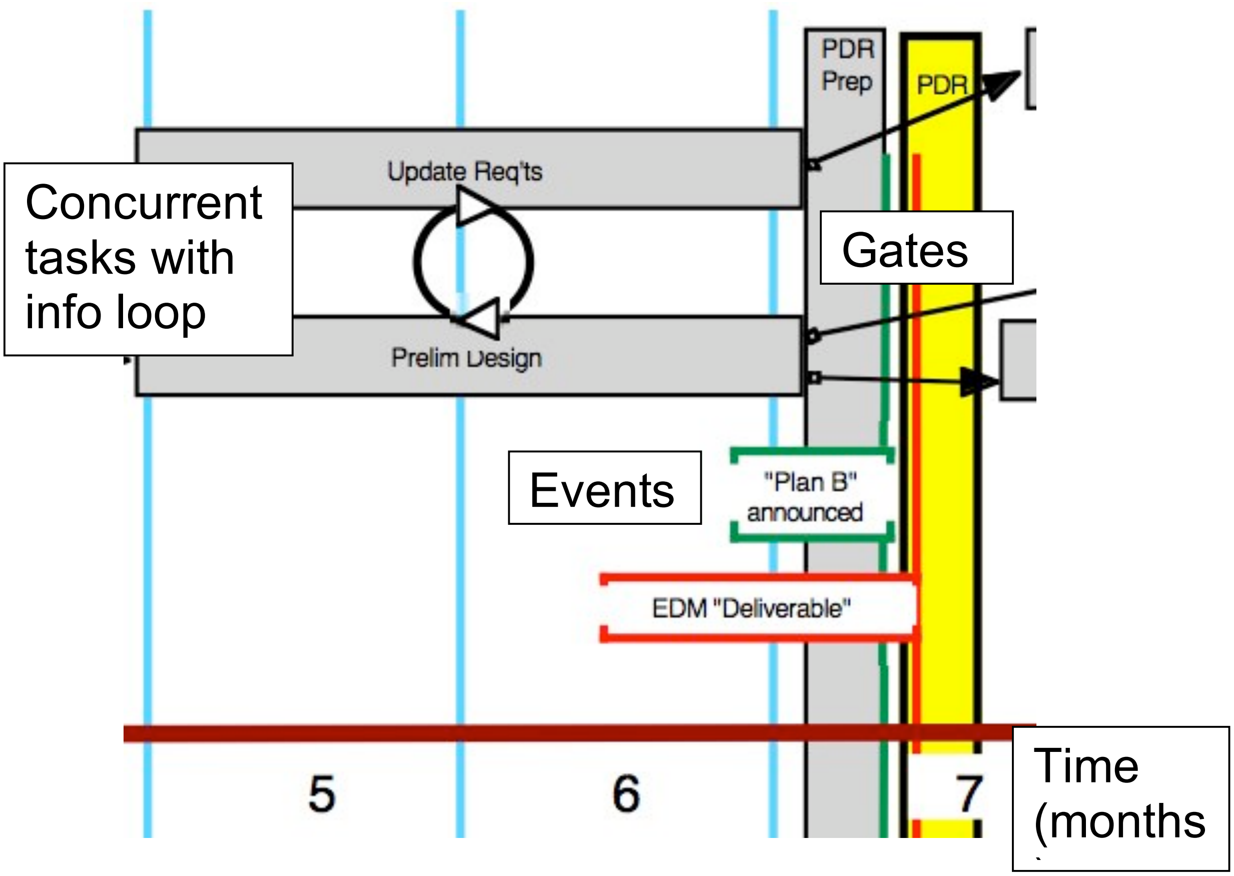


Figure D-1. Portion of a swim-lane value stream map showing concurrent tasks, gates, and events (modified from Whitaker).

Another symbol of note is the “Washing Machine” shown in Figure D-2. Here, many tasks proceed in parallel, but have an complex interaction, including not just information passing but a shared, iterative process. This kind of process is typical of, for example, preliminary design, where information from many functional groups must come together and be iterated upon before a feasible design is decided upon. Note that not all functions are “in the machine,” and several have discrete interactions with parts of the machine cycle.

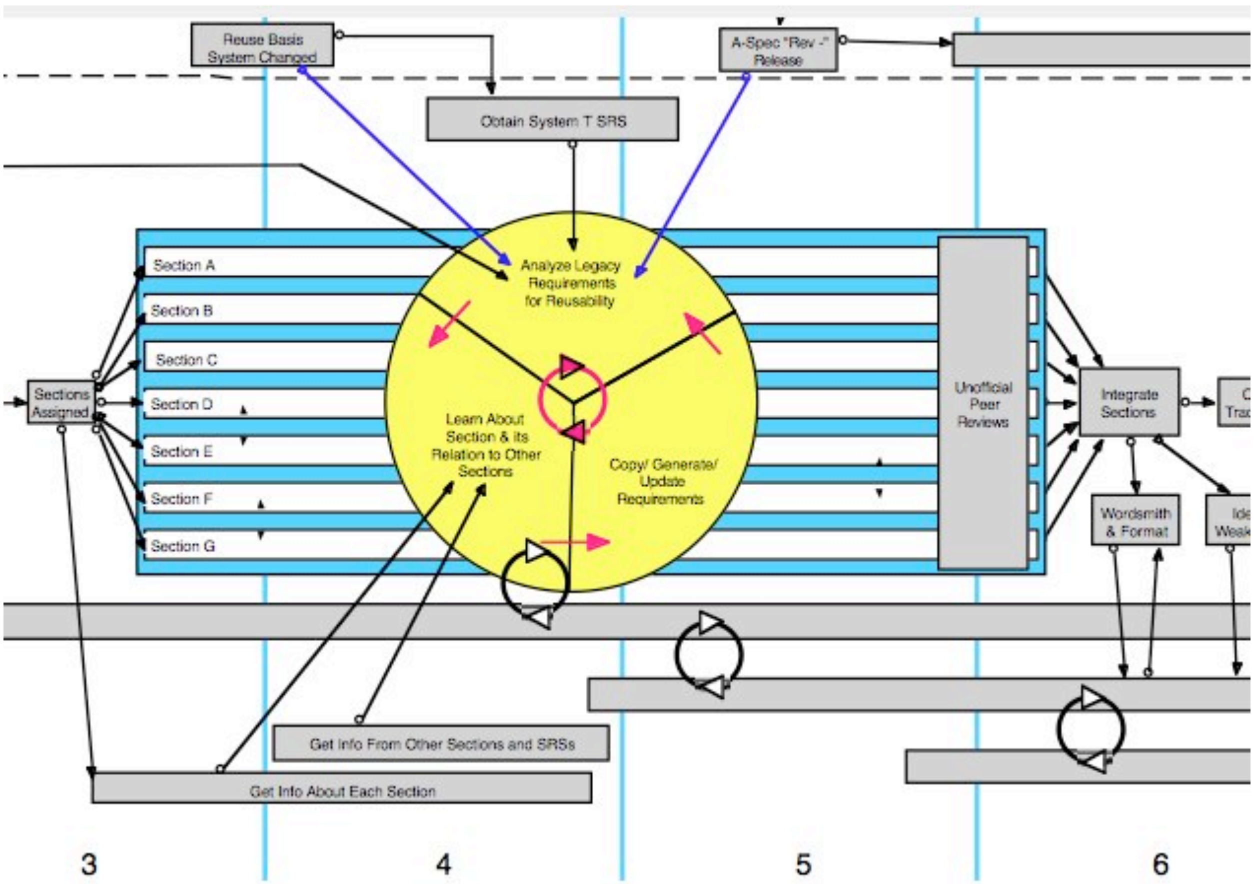


Figure D-2. Portion of a swim-lane value stream map showing many concurrent tasks, in an interactive repeated process a/k/a the washing machine (modified from Whitaker).

Use these and other symbols as necessary to make your VSM work for your purposes.

NOTES AND REFERENCES

Note that most LAI products, thesis, reports, and conference proceedings are available on the web at http://lean.mit.edu. Access to some documents is restricted to LAI members. Contact the LAI staff for LAI documents not found on the web site.

---

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

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

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

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

5 Bernstein, Joshua I., "Multidisciplinary Design Problem Solving on Product Development Teams," Doctoral thesis in Technology, Management and Policy, Massachusetts Institute of Technology, February 2001.

6 Womack, James P and Daniel T. Jones, *Lean Thinking*, Simon & Schuster, New York, NY, 1996.

7 Rother, Mike and John Shook, *Learning To See: value stream mapping to add value and eliminate muda*, Lean Enterprise Institute, Brookline, MA, 1999.

8 Jones, Dan, and Jim Womack, *Seeing the Whole: mapping the extended value stream*, Lean Enterprise Institute, Brookline MA, 2002, discusses multi-factory value steams, and Rother, Mike and Rick Harris, *Creating Continuous Flow: an action guide for managers, engineers and production associates*, Lean Enterprise Institute, Brookline MA, 2001, which discusses getting a single factory cell to flow. Interestingly, it is the latter book that has the most parallels to PD processes.

9 Trischler, William E., *Understanding and Applying Value-Added Assessment: Eliminating Business Process Waste*, ASQC Quality Press, Milwaukee, Wisconsin, 1996.

10 The MIT Design Structure Matrix - DSM - Home Page, Massachusetts Institute of Technology, http://web.mit.edu/DSM.

11 *Learning to Develop: A Guide to Understanding and Applying Lean Principles to Product Development*, Aerojet-General Corporation, Sacramento, CA, 2002. LAI PD participant Adi Choudri was the team lead on this effort. *Learning to Develop* covers some of the same territory as this manual, but at a much higher level (the entire PD process) and with much

greater breadth. It is an excellent source of “big ideas” while this manual focuses you on the particulars. An overview is available at http://www.learningtodevelop.com/default.htm.

12 Murman, E., Allen, T., Bozdogan, K., Cutcher-Gershenfeld, J., McManus, H., Nightingale, E., Rebentisch, E., Shields, T., Stahl, F., Walton, M., Warmkessel, J., Weiss, S., and Widnall, S., *Lean Enterprise Value*, Palgrave, London, 2002.

13 *Transitioning to a Lean Enterprise: a Guide for Leaders*, Lean Aerospace Initiative, Massachusetts Institute of Technology, 2000.

14 *Enterprise Value Stream Mapping and Analysis*, Lean Aerospace Initiative, Massachusetts Institute of Technology, 2003.

15 *Product Development Transition to Lean Module*, Lean Aerospace Initiative, Massachusetts Institute of Technology, 2002.

16 McManus, Hugh L., and Rebentisch, Eric, “Lean Enterprise Value Training Simulation,” Lean Aerospace Initiative, Massachusetts Institute of Technology, 2004, or see the LAI website under products, or see http://www.metisdesign.com/game_example.htm. Additional information on the use of the simulation in various contexts can be found by searching the LAI website for the keyword “simulation.” Periodically, short courses using the simulation are given; see the LAI website under “Event Calendar.”

17 McManus, H., Hatterty, A. and Murman, E., “Lean Engineering: Doing the Right Thing Right,” 1st International Conference on Innovation and Integration in Aerospace Sciences, Queen’s University Belfast, Northern Ireland, UK, 4-5 August 2005.

18 Ulrich, Karl T., and Steven D. Eppinger, *Product Design and Development*, McGraw-Hill, Boston, 1995, pp. 14-18. This book, and the updated 2nd edition, 1999, are excellent introductions to thinking about product development as a process.

19 Wirthlin, Robert J., “Best Practices in User Needs/Requirements Generation,” Masters thesis Engineering and Management, Massachusetts Institute of Technology, January 2001.

20 See Womack and Jones, p. 308, and the greatly expanded definition in Murman *et al.*, Chapter 3 in general and in particular p. 72.

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

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

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

24 Kato, Jin, "Development of a Process for Continuous Creation of Lean Value in Product Development Organizations," Master’s thesis in Mechanical Engineering, Massachusetts Institute of Technology, June, 2005.

25 It is important to note that Figure 2-4 is notional, not a representation of any one studied process. The data for Figures 2-2 and 2-3 were collected from separate studies, and combining them in this way is not strictly valid. However, it we accept that Figures 2-2 and 2-3 present accurate pictures of a *typical* process, then Figure 2-4 is a powerful depiction of the *typical* levels of waste present in PD processes.

26 For example, see Goodman, Gary, "F-16 Lean Build-To-Package Support Center Process," presented at the LAI Product Development Winter 2000 Workshop, Folsom CA, January 26-28, 2000, (or Murman *et al*., pp. 125-127, which abbreviates the same story), Albecht, Karen, "Lean PD Efforts," presented at the LAI Product Development Winter 2000 Workshop, Folsom CA, January 26-28, 2000, and "Global hawk Lean Program—After *Lean Now*," LAI website presentation, May, 2005. Newer (though less complete) examples include Thompson, Steven and Bergman, Gary "That doesn't apply here…or does it? Lean Deployment for Product Development Teams," LAI Plenary meeting, Dana Point CA, 2005, and Secor, Deb and Bliss, Dean, "Transforming the Enterprise Product Development Function," LAI Plenary meeting, Dana Point CA, 2005.

27 Adapted from Albecht.

28 Dramatic savings in engineering *effort* are possible with large-scale adoption of new processes and technical tools such as 3-D solid modeling. Although we will not discuss specific technical tools here, effective application of these tools has been found to critically depend on defining *processes* to fit tool capabilities. Value stream mapping can be a key enabler of this process redefinition.

29 Womack, James P., Jones, Daniel T., and Roos, Daniel, *The Machine that Changed the World: The Story of Lean Production*, HarperCollins, New York, 1990, pp. 92-93.

30 For a dramatic example of this, see Murman *et al.*, pp.118-120.

31 The customer and end user may be the same, particularly for a commercial product. In government aerospace, the customer handling the acquisition process and the end users (often the warfighters) are *not* the same, and they may have some differences of interest.

32 This section is an adaptation of material from Millard, pp. 130-131.

33 This situation is steadily improving however; many LAI members are developing this expertise internally, and LAI has helped to develop both the *Lean Now!* Subject matter experts and LEV-simulation based training for Lean Engineering.

34 LAI administers an Educational Network for deploying basic lean education in cooperation with participating university partners. See the LAI website under "EdNet."

35 "LAI's Lean Enterprise Value Business Simulation Aids in Mapping Enterprise Value Stream of Textron's SFW program," LAI News Sept. 2004, and "LAI Lean Now Team Engages EC-130 Compass Call Program," LAI News 2005.

36 This discussion based on Millard, p 65.

37 Slack, above, or Slack, Robert R., "The lean value principle in military aerospace product development," Lean Aerospace Initiative Report RP99-01, Massachusetts Institute of Technology, 1999, which contains the relevant material.

38 Chase, above.

39 Browning, above, and also Tyson R. Browning, John J. Deyst, Jr., Steven D. Eppinger, and Daniel E. Whitney, "Complex System Product Development: Adding Value by Creating Information and Reducing Risk," The Lean Aerospace Initiative Report Series WP99-03, December, 1999, modified and published as Browning, Tyson R., John J. Deyst, Steven D. Eppinger and Daniel E. Whitney (2002) "Adding Value in Product Development by Creating Information and Reducing Risk," IEEE Transactions on Engineering Management, 49(4): 443-458.

40 If your situation does demand a deeper look at what vale means in an aerospace product development context, See Slack's decomposition of customer value, Browning *et al.* on risk reduction, and Chase on value creation.

41 Note this exactly the sort of situation Rother and Shook are looking at in *Learning to See.* They analyze the production of steering bracket, a part invisible to the final customer. But the car the customer wants cannot be built without it.

42 An unreleased member study has found that an alarming percentage of PD process outputs are not needed by downstream processes, for program knowledge capture, for meeting regulations, contractual requirements, or quality standards, or for any other purpose. They are waste. Conversely, failure to archive completed PD artifacts such as FEM models in a usable and accessible way can cause substantial wastes due to information hunts or rework later.

43 Quote take verbatim from Browning *et al*., p 1.

44 Reinertsen, D. "Testing: Annoyance or Opportunity," *Electronic Design,* Vol. 46, No. 26, 1998, p. 64.

45 See Chase, pp. 52-53.

46 See Chase, pp. 97-99.

47 Whitaker, Ryan B., "Value Stream Mapping and Earned Value Management: Two Perspectives on Value in Product Development," Master's thesis in Engineering Systems, Massachusetts Institute of Technology, Sept. 2005.

48 Adapted from Chase, p. 59.

49 A JPL study estimated that critical numbers associated with a design are *typed* into computers an average of *15 times* over the course of the design process.

50 Womack and Jones, pp 264.

51 Millard, Chapters 4 and 5, pp. 39-60.

52 McManus, Hugh L., and Richard L. Millard, "Value Steam Analysis and Mapping for Product Development," Proceedings of the International Council of the Aeronautical Sciences, 23rd ICAS Congress, Toronto Canada, September 8-13, 2002.

53 Womack and Jones, pp. 37-43, has an excellent example of a value stream map created by following a cola can from ore in the ground to disposal.

54 Dong, Qi , "A Proposed Approach for Building Credible DSMs ," http://web.mit.edu/DSM/Tutorial/build_Qi.htm, which is an extract from Dong, Qi, "Representing Information Flow and Knowledge Management in Product Design Using the Design Structure Matrix," Masters thesis in Mechanical Engineering, Massachusetts Institute of Technology, January, 1999.

55 Charts from Millard, p. 68.

56 If the disconnect between where people think their outputs go and where people think they are getting their information is pathological, the handshake matrix (see *Learning to Develop*, pp. 132-133) can be used as a tool to understand the problem.

57 A very interesting and somewhat disturbing high-level look at this problem (which documents a major disconnect between the perceptions of information creators and their internal customers) is Braha, Dan and Bar-Yam, Yaneer, "Topology of large-scale engineering problem-solving networks," *Physical Review E*, Vol. 69, 016113, 2004.

58 Trischler, pp. 75-76. The discussion of data and what to do with it in all of Chapter 8 is worth reading, although it has a different flavor from a lean value stream analysis.

59 Weigel, Annalisa L., "Spacecraft System-Level Integration and Test Discrepancies: Characterizing Distributions and Costs," Masters thesis in Aeronautics and Astronautics, Massachusetts Institute of Technology, June 2000. Also, Weigel, Annalisa L. and Joyce M. Warmkessel, "Cross-Industry Characterization of Spacecraft Integration and Test Discrepancies: Transforming Discrepancies into Product Development Improvements," American Institute of Aeronautics and Astronautics paper AIAA-2000-5084, presented at the AIAA Space 2000 Conference and Exposition, Long Beach CA, September 19-21, 2000.

60 From Browning, Tyson R. (2003) "On Customer Value and Improvement in Product Development Processes," Systems Engineering, 6(1): 49-61: "Whenever a group attempts to classify PD activities as one of Womack and Jones' three types, it typically experiences some passionate debate. PD activities can be difficult to classify, and no one wants to see their activity branded as "waste," necessary or not. Actually, most of the non-value-adding elements are buried deep inside value-adding activities. In the largest sense, the overall PD process adds value (i.e., is a type 1 activity). Yet, decompose it and type 2 and type 3 activities appear within. Continue to decompose the type 1 activities, and activities of the other two types continue to appear. Decompose *ad infinitum*, and the only thing left adding value (by the "three types" definition) is the final output materializing out of thin air! Thus, debating whether an activity is type 1 or type 2 is not very helpful in practice, since the former contains much of the latter anyway. Just think of the entire process in economic terms: remove type 3 activities and make everything else as productive and efficient as possible. The concept of "necessary waste" can be an unnecessary distraction."

61 The LAI website, in particular the proceedings of the product development workshops and research group meetings from 1998 through 2002, contains a history of the development of this idea.

62 Bauch, Chrstoph, "Lean Product Development: Making waste transparent," Diploma Thesis in Product Development, Technical University of Munich (work performed at the Massachusetts Institute of Technology), August 2004.

63 Table 5-1 is a modification of the information waste table in the LAI Enterprise Value Stream Mapping and Analysis (EVSMA) Beta version manual. This manual also has tables of factory and enterprise wastes that make for interesting contrasts.

64 Taken from Slack 1998, p. 34.

65 If only a single job is required to move through the process, the desired cycle time can usually be met simply by removing external causes of waiting and giving the job high-priority access to all needed resources. This happens; urgent jobs are often "walked through" processes, disrupting ongoing work. This approach fails if even two jobs are expedited; they will pile up in front of the slowest task.

66 For example, the Aerojet VIPS system; see http://www.learningtodevelop.com

67 During the Textron SFW improvement event a team was tasked to document and reduce excessive meetings in the program. They found that the program *was not* generating excessive meetings; the perception of excessive meetings was due to the fact that many program employees were working on multiple programs, all of whose meetings they were expected to attend.

68 Womack and Jones, p. 308.

69 Murman *et al.*, p. 72.

70 http://web.mit.edu/lean, click on Products, then Lean Enterprise Model. The "Architecture" shows the basic LEM tool, which is publicly available. Greater depth and the source data are available to LAI members; LAI member password is required as of this writing.

71 Millard, pp. 84-93.

72 Bicheno, John, "Automotive Value Stream Mapping," Lean Enterprise Research Centre, Cardiff Business School, 1998.

73 Allen, Thomas J., *Managing the Flow of Technology: Technology Transfer and the Dissemination of Technological Information within the R&D Organization*, MIT Press, Cambridge, MA, 1988.

74 The C-TOS ionospheric mapping satellite cluster system designed by MIT, Stanford, and CalTech. Information on this project is available at http://web.mit.edu/SSPARC; follow the links through "About SSPARC" to "SSPARC Architecture and Design Projects."

75 This discussion excerpted from Bernstein, pp. 94-95.

76 Henderson, Kathryn,"Flexible Sketches and Inflexible Data Bases: Visual Communication, Conscription Devices, and Boundary Objects in Design Engineering," *Science, Technology, and Human Values*, Vol. 16, No. 4, Autumn 1991, pp. 448-473.

77 Carlile, Paul R., "Understanding Knowledge Transformation in Product Development: Making Knowledge Manifest through Boundary Objects," Doctoral Dissertation, University of Michigan, 1997.

78 Smith, Preston G. and Donald G. Reinertsen, *Developing Products in Half the Time: New Rules, New Tools*, 2nd Edition, John Wiley and Sons, 1997.

79 Reinertsen, Donald G., *Managing the Design Factory: The Product Developer's Toolkit*, Free Press, 1997.

80 Mascitelli, Ronald, *Building a Project-Driven Enterprise*, Technology Perspectives, 2002.

81 George, Michael L., *Lean Six Sigma: Combining Six Sigma Quality with Lean Production Speed*, McGraw-Hill Trade, 2002.

82 Goldratt, Eliyahu M., *Critical Chain*, North River Press, 1997.

83 See http://lean.org under Community / Thinkers Corner.

84 Oppenheim, Bohdan W., "Lean Product Development Flow," *Systems Engineering*, Vol. 7 No. 4, Oct. 2004. See also presentation at the Spring 2004 LAI Product Lifecycle group meeting, Laguna Beach, CA, March 25, 2004.

85 Parkin, K., Sercel, J., Liu, M., and Thunnissen, D., "ICEMaker: An Excel-Based Environment for Collaborative Design," 2003 IEEE Aerospace Conference Proceedings, March 2003.

86 See, for example, http://web.mit.edu/SSPARC (click down to About SSPARC), http://www.lsmd.caltech.edu/, http://NewDesignParadigms.jpl.nasa.gov/ and http://nsd2001.jpl.nasa.gov/

87 Stagney, David, "Applying MATECON principles in satellite design," presented at the LAI Product Development workshop, Dayton OH, March 2003, and Stagney, David, "The Integrated Concurrent Enterprise," Master's thesis in Aeronautics and Astronautics and Management, Massachusetts Institute of Technology, September, 2003.

88 Ross, A. M., Diller, N. P., Hastings, D. E., and Warmkessel, J. M., "Multi-Attribute Tradespace Exploration with Concurrent Design as a Front-End for Effective Space System Design;" the MATE-CON method and many uses and expansions are also reviewed in McManus, H. L., Hastings, D. E., and Warmkessel, J. M., "New Methods for Rapid Architecture Selection and Preliminary Design." Both are published in *Journal of Spacecraft and Rockets*, Vol. 41, No. 1, Jan/Feb 2004, in a special section along with several other papers on the MATE-CON method.

89 McManus, Hugh L., and Schuman, T. E., "Understanding the Orbital Transfer Vehicle Trade Space," AIAA paper 2003-6370, and Galabonva, K., Bounova, G., de Weck, O. and Hastings, D., "Architecting a Family of Space Tugs Based on Orbital transfer Mission

Scenarios," AIAA paper 2003-6368, both presented at AIAA Space 2003, Long Beach CA, Sept. 2003.

90 Millard, p. 97.

91 Sterman J., *Business Dynamics: Systems Thinking and Modeling for a Complex World*, Irwin/McGraw-Hill, 2000.

92 See the "Lean Now!" tab at the LAI website.

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