Development of a Process for Continuous Creation of Lean Value in Product Development Organizations

by

Jin Kato

B. Eng., Aeronautics and Astronautics (1996)
M. Eng., Aeronautics and Astronautics (1998)
University of Tokyo

Submitted to the Department of Mechanical Engineering
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Mechanical Engineering

at the
Massachusetts Institute of Technology

June 2005

Signature of Author..
Department of Mechanical Engineering
May 6, 2005

Certified by ....
Warren Seering
Weber-Shaughness Professor of Mechanical Engineering and Engineering Systems
Thesis Supervisor

Accepted by ....
Lallit Anand
Chairman, Department Committee on Graduate Students

Development of a Process for Continuous Creation of Lean Value in Product Development Organizations

by

Jin Kato
jkato@alum.mit.edu

Submitted to the Department of Mechanical Engineering
on June 6, 2005 in Partial Fulfillment of the Requirements for
the Degree of Master of Science in Mechanical Engineering

ABSTRACT

Ideas and methodologies of lean product development were developed into tools and processes that help product development organizations improve their performances. The definition of waste in product development processes was re-examined and developed into a frugal set to cover all types of waste in product development processes through preliminary case studies. Value stream mapping (VSM) was optimized for measuring the waste indicators in product development processes. Typical causes for low product development project performances were organized into a root-cause analysis diagram.

Three case studies in product development companies were performed. The tools were tested and improved through intensive interviews with both project managers and engineers. VSM was effective for identifying and measuring waste indicators. The root-cause analysis diagram was effective for quickly identifying root causes for low product development project performances. Synchronized uses of these tools made it possible to measure each root cause's impact on project performances. The result of measurements revealed both problems shared by all the projects and the ones specific to the projects, indicating that the tools and processes developed in this research can provide suggestions for continuous improvement of product development processes.

Some waste indicators were more prevalent than the others, implying that the number of waste indicators to be considered can be reduced. Inventory of information was prevalent in all the projects, and the analyses of it implied that Today's product development processes are as premature as those of manufacturing several decades ago. Wastefulness of information inventory was proved quantitatively. Time spent on one occurrence of rework was proved to take longer near the end of a project than at the beginning of it.

Thesis Supervisor: Warren Seering
Title: Weber-Shaughness Professor of Mechanical Engineering and Engineering Systems

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my thesis advisor, Professor Warren Seering. He has always provided for me invaluable guidance in a manner I have never experienced. I have learned from him not only knowledge in product development, but also ways to keep people motivated all the time.

I am also thankful to Professor Seering's students. Victor Tang kindly helped me to find research investigation opportunities, based on his outstanding carrier in managing big projects. Christoph Bauch, Martin Graebsch, Josef Oehmen, and Ryan Whitaker made it possible to formulate my ideas through thoughtful suggestions.

I would like to thank East Japan Railway Company, my employer, for giving me a special opportunity to learn in the best environment.

I am indebted to the two excellent companies that kindly accepted my proposal to conduct my research investigation.

Lean Aerospace Initiative (LAI) made it possible to investigate these two companies in Japan.

I could not have completed my study at MIT without the full support of my wife, Chiaki, and my daughter, Niina.

TABLE OF CONTENTS

CHAPTER 7: HOW TO MEASURE WASTED TIME USING VALUE STREAM MAPPING

CHAPTER 8: INVENTORY OF INFORMATION AND HOW TO MEASURE IT USING VALUE STREAM MAPPING

CHAPTER 9: OVERVIEW OF THE RESEARCH INVESTIGATIONS

CHAPTER 10: RESULTS OF RESEARCH INVESTIGATIONS 1: ANALYSES ON NINE WASTE INDICATORS

CHAPTER 11: RESULTS OF THE RESEARCH INVESTIGATIONS 2: ANALYSES ON INVENTORY OF INFORMATION

LIST OF FIGURES AND TABLES

FIGURES

TABLES

CHAPTER 1 INTRODUCTION AND OVERVIEW

1.1 MOTIVATION

Although different types of waste in product development have been suggested, there have been no research investigations in determining what types of waste are more prevalent than others in terms of wasted engineering time. Although there have been substantial amount of literature on how to successfully manage product development projects, there is no practical tool with which project managers can quantitatively analyze their unsuccessful projects. For these reasons, there is no effective project management tool that enables product development project managers to know how much each factor quantitatively affects their product development project performances. For example, they may attribute their product failure to late specifications/ requirements changes, but they cannot estimate how much those changes affected the overall project performances. Therefore, it is difficult for them to know what the right thing is to improve their product development processes.

1.2 GOALS AND OBJECTIVES

The goal of this research is to develop a process for continuous creation of lean value in product development organizations. Creation of lean value here means realizing value with minimum wasteful process.

To achieve this goal, the objective of this research was determined to develop ideas and methodologies of lean product development into tools and processes that can help product development organizations (1) identify and measure the waste in their teams' processes; (2) identify causes and measure their impacts on PD processes; and (3) finally learn the best strategies to pursue to improve their PD processes.

1.3 OVERVIEW OF THE REMAINING CHAPTERS

Different types of waste were reexamined and nine plus one waste indicators were selected. Value stream mapping was optimized for quantitative measurement of wasted time. Exploration of causal relationships among these waste indicators and various types of causes for waste lead to a comprehensive diagram that can be used for identifying root causes for waste. To test the supposition that these can be applicable for quantitative analysis of product developments projects, three case studies were performed.

Chapter 2 takes a look at lean manufacturing, and how it has been developed into lean product development.

Three preliminary case studies were performed in chapter three, in which value stream mapping's applicability for quantitative measurements of wastes suggested in literature on lean product development was evaluated. Several problems were identified. These problems were addressed in chapter 4.

Different types of waste that have been suggested in studies in lean product development are compared in chapter 5 by exploring causal relationships among each waste. This study was finally developed into the root-cause analysis diagram by adding more types of waste identified in papers and books, and the preliminary case studies.

Chapter 6 is a how-to manual for drawing value stream maps for quantitative measurement of waste. Many features of value stream mapping that were unnecessary for the purpose were eliminated.

A methodology for measuring waste using nine waste indicators is described in chapter 7. Inventory of information can be measured by the methodology described in chapter 8. These methodologies were applied in the case studies covered in the following three chapters.

Chapter 9 introduces the case studies in three industrial product development projects. Chapter 10 discusses the quantitative analysis results obtained using nine waste indicators. Three waste indicators detected significantly more waste than the other six. The results also revealed rework takes more time at the end of projects than at the beginning of them. Chapter 11 discusses results obtained by identifying inventory of information. In a project in which market and technical risks are high, inventory of information became bad at the rate of six percent per month. The analysis of the results described in chapters 10 and 11 suggests that the tools and methodologies developed in this research can show how engineers' time is wasted in each specific project, implying these can be used for improving each project's processes.

Chapter 12 concludes this research.

1.4 NOTE

This thesis contains some figures in color, although the author tried to convey all the information in black and white whenever possible. The full-color version is available online at Lean Aerospace Initiative (LAI)'s website: lean.mit.edu.

CHAPTER 2 LITERATURE REVIEW

The concept of "lean" was first applied to product development by introducing ideas and tools of lean manufacturing. Wormack and Jones (1996) defined five lean principles, "specifying value," "identify the value stream," "flow," "pull," and "striving for perfection." Two research topics, definition of waste and practical way of value stream mapping have been focused on by many scholars and product development practitioners, based on the idea that addressing these topics lead to realization of the five lean principles. From the perspective of waste, Wormack and Jones introduced nine categories of waste by adding two new categories to Toyota's seven categories of waste in manufacturing. Slack (1999) tried to prioritize the nine types of waste by conducting surveys of product development organizations, questioning each category's frequencies. He also analyzed each category's effect on value.

The definition of categories of waste has continuously been discussed by exploring the differences between manufacturing and product development environment. Morgan (2002) dramatically changed the definition of waste from the perspective of systems engineering. Based on the idea that unsynchronization leads to low performance in product development processes, he introduced eleven categories of waste, replacing all but one: waiting. Recognizing interdependency among the categories of waste defined by forerunners, Bauch (2004) re-defined ten categories of waste by analyzing interactions among the categories.

Value stream mapping has also been tried to apply to product development processes as product development value stream mapping (PDVSM). Early versions of PDVSM, inherited many features in value stream mapping for manufacturing, were not capable of displaying activities specific to the product development environment such as iteration and multiple tasking. Morgan (2002) improved PDVSM by making each process box' length proportional to the time spent on it. Although this suggestion clearly differentiated PDVSM from ordinal process maps in that unsynchronization became visible, how to display iteration and multiple tasking remained to be solved.

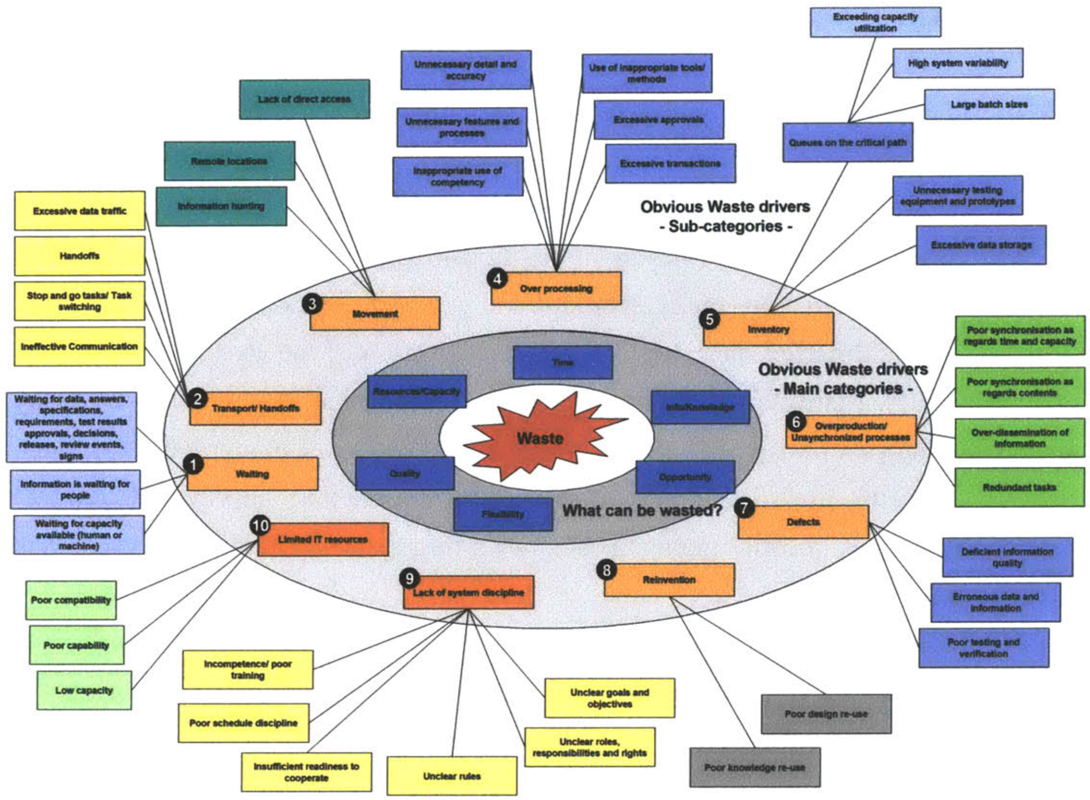


Figure 2.1 Ten Categories of Waste in Product Development (Bauch, 2004)

The definitions of waste have not been explicitly utilized for displaying waste in value stream maps until Graebsch (2005) applied Bauch's definition to his microscopic case studies of ongoing MIT student projects. He successfully measured occurrences of waste by displaying it in value stream maps, making it possible to measure a frequency of each category of waste on a value stream map. This achievement raised the following research research questions:

1. Can value stream mapping be used for measuring wasted time?
2. Can value stream mapping be applied to analyses of industrial product development processes?

CHAPTER 3 PRELIMINARY STUDIES

3.1 INTRODUCTION

To evaluate value stream mapping's applicability for measuring wasted time, three preliminary case studies had been performed.

3.2 THE FIRST PRELIMINARY CASE STUDY

3.2.1 OBJECTIVE OF THE FIRST CASE STUDY

The objective of this case study was to evaluate value stream mapping's applicability to measuring wastes defined by McManus (2004) and Morgan (2002).

3.2.2 SELECTED PROJECT AND FOCUS

The first project chosen for this evaluation was a railway vehicle constructor's development process. The railway vehicle had been developed by three design teams: body, power/electronics, and bogie teams. For simplicity, only a mechanical designer's design process had been tracked, based on value stream mapping techniques proposed by Morgan and McManus. The designer's task was to re-design the structure of the current model of railway vehicle by performing finite element analysis. The structure change had affected the other teams design, causing cross-team iteration. The project was finished in 2002.

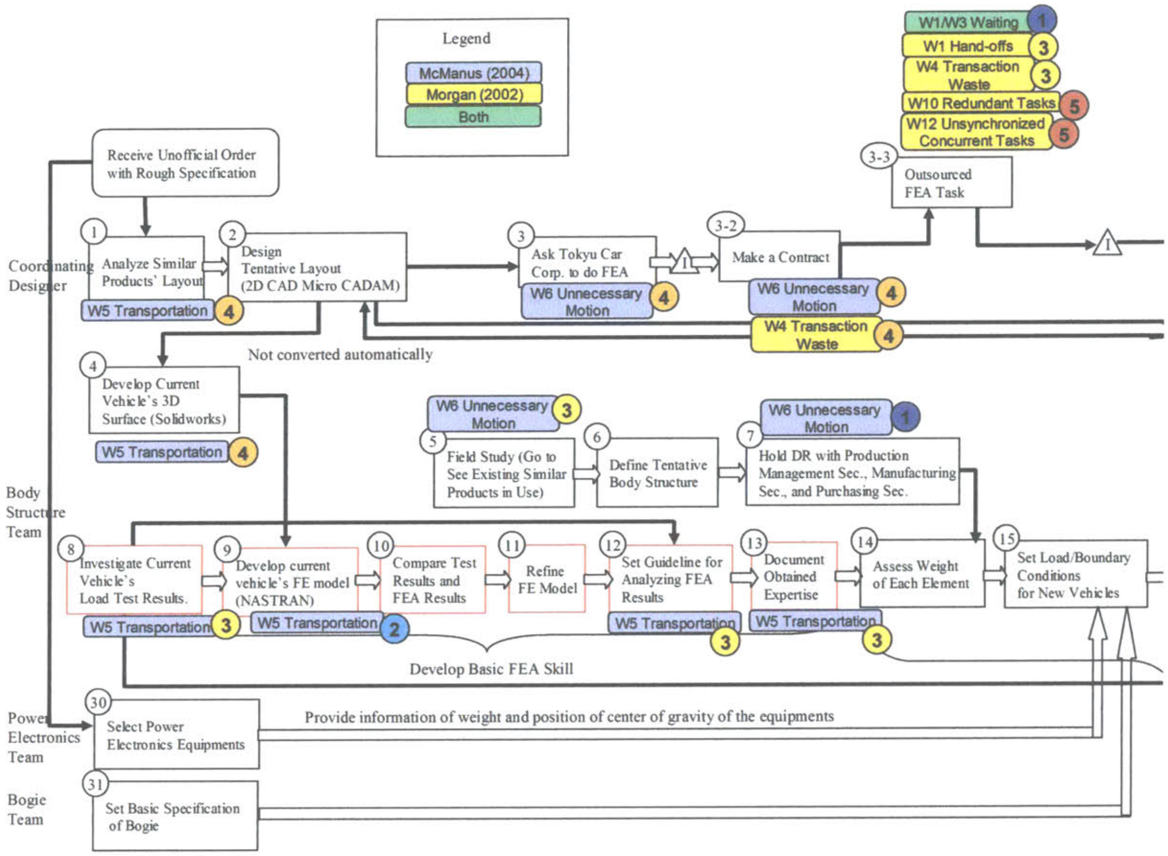


Figure 3.1 Value Stream Map for One Designer's Activities with Waste Defined by McManus (2004) and Morgan (2002) –Continued to Figure 3.2

3.2.3 DRAWING A VALUE STREAM MAP

Value stream maps created in the first preliminary case study are shown in figures 3.1 and 3.2. In this case study, process boxes' lengths were not made proportional to the time spent on them because the remaining data of the development process did not have detailed information about time spent on each task. Based on the designer's memory, which was the only available resource, roughly estimated waste time was put on the value stream map. This map was created through the following steps.

Step. 1. Draw all process boxes and add information flows.

Step. 2. Identify wastes in the lists of Morgan's and McManus'.

Step. 3. Estimate roughly wasted time spent on each waste.

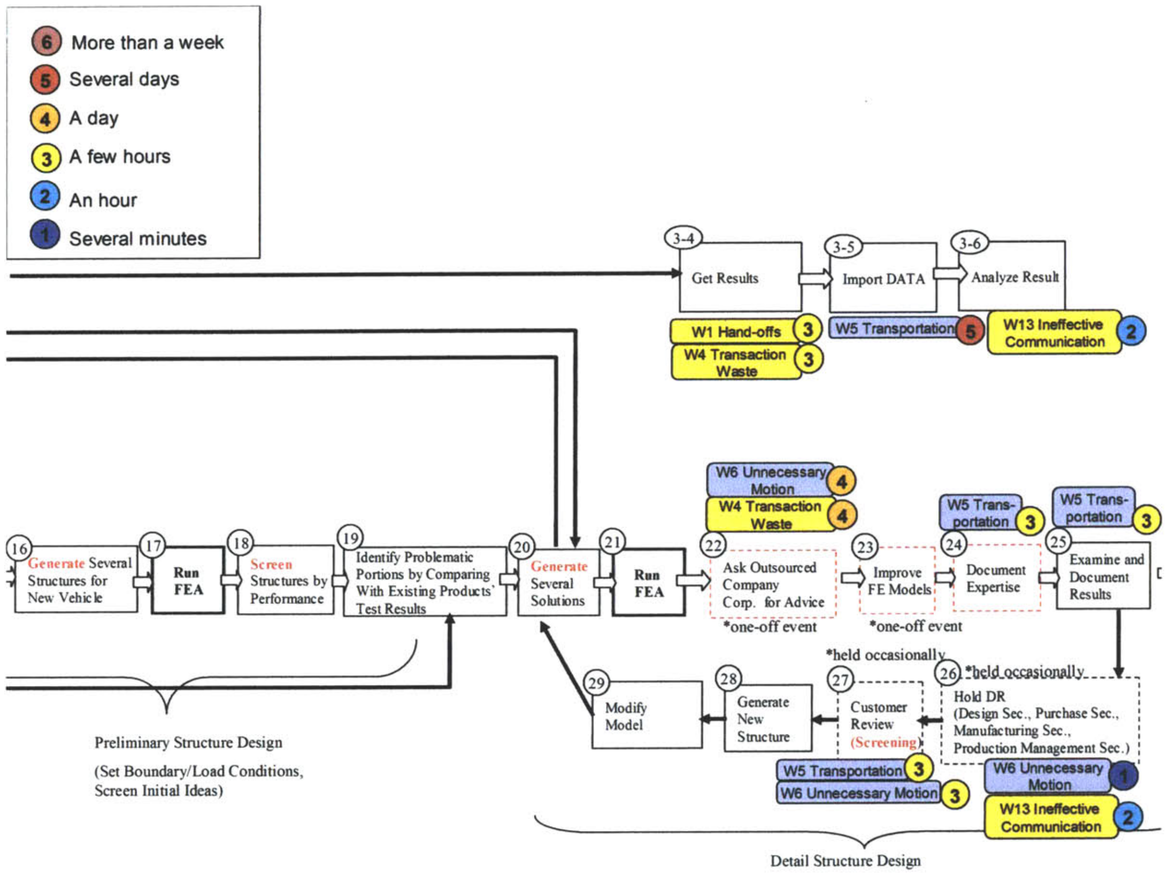


Figure 3.2 Value Stream Map for One Engineer's Activities with Waste Defined by McManus (2004) and Morgan (2002) –Continued from Figure 3.1

3.2.4 EVALUATION

In this case study, rework was shown by using an arrow that went back to the reworked task. For example, in figures 3.1 and 3.2, tasks between (2) and (20) were repeated several times. This way of showing rework made it impossible to satisfy the following rules suggested by Morgan.

1. Process boxes' lengths should be proportional to the time spent on them.
2. Process boxes' order should be the same as their actual occurrences.

This problem is obvious when some downstream tasks are affected by an occurrence of rework like in figure 3.3. In this figure, task 3, which was started after task 2's start, appears before task 2, which does not satisfy the second rule above. Thus, using going-back arrows makes it impossible to follow Morgan's two rules.

Another problem is that using a going-back arrow makes it impossible to display how a project's schedule is affected by an occurrence of a wasteful activity, such as making defective information. For example, it is not obvious in figure 3.3 whether task 2 was reworked because of defective information received from task 1, or it was reworked because of defective information made inside of task 2. Thus, using a going-back arrow makes it unclear how other tasks are affected by a wasteful activity. For the same reason, measuring waste is also difficult when a going-back arrow is used.

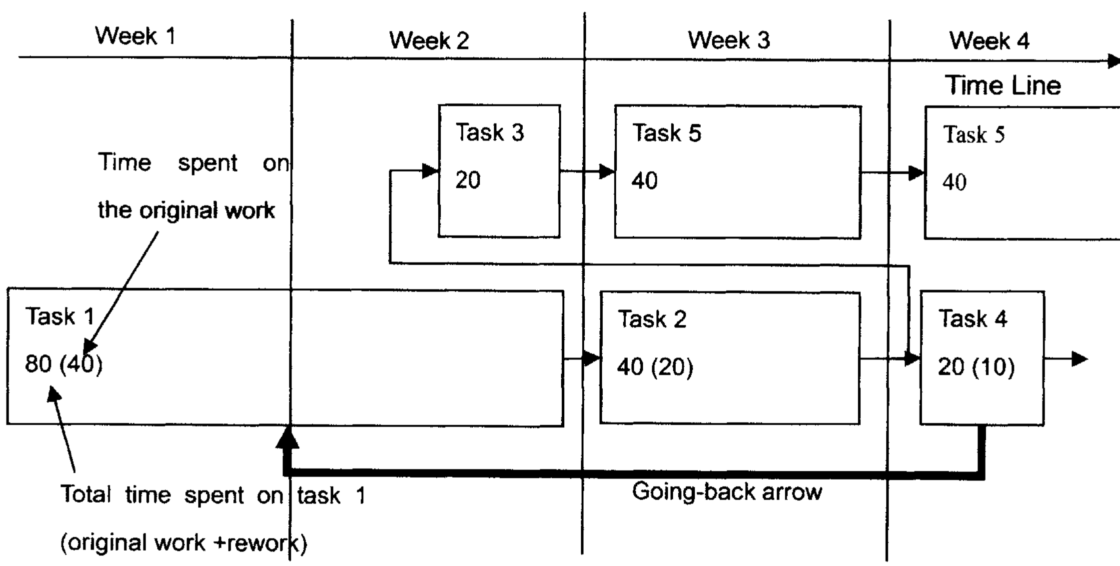


Figure 3.3 Simplified Value Stream Map with Rework with Going-Back Arrows

3.3 THE SECOND PRELIMINARY CASE STUDY

3.3.1 OBJECTIVE OF THE SECOND CASE STUDY

The objective of this case study was to evaluate value stream mapping's applicability to a whole product development project.

3.3.2 SELECTED PROJECT AND FOCUS

The same railway vehicle development project was selected for the second case study. This time, the whole development processes were selected as the scope of the value stream map.

3.3.3 DRAWING A VALUE STREAM MAP

Figure 3.4 shows the value stream map drawn in this preliminary case study. Contrary to the first case study, process boxes' lengths were made proportional to the time spent on them, although the accuracy was limited. After several trials, the swim-lane form was finally chosen.

3.3.4 EVALUATION

Adoption of swim-lane form made it possible to draw a value stream map without making it too complicated.

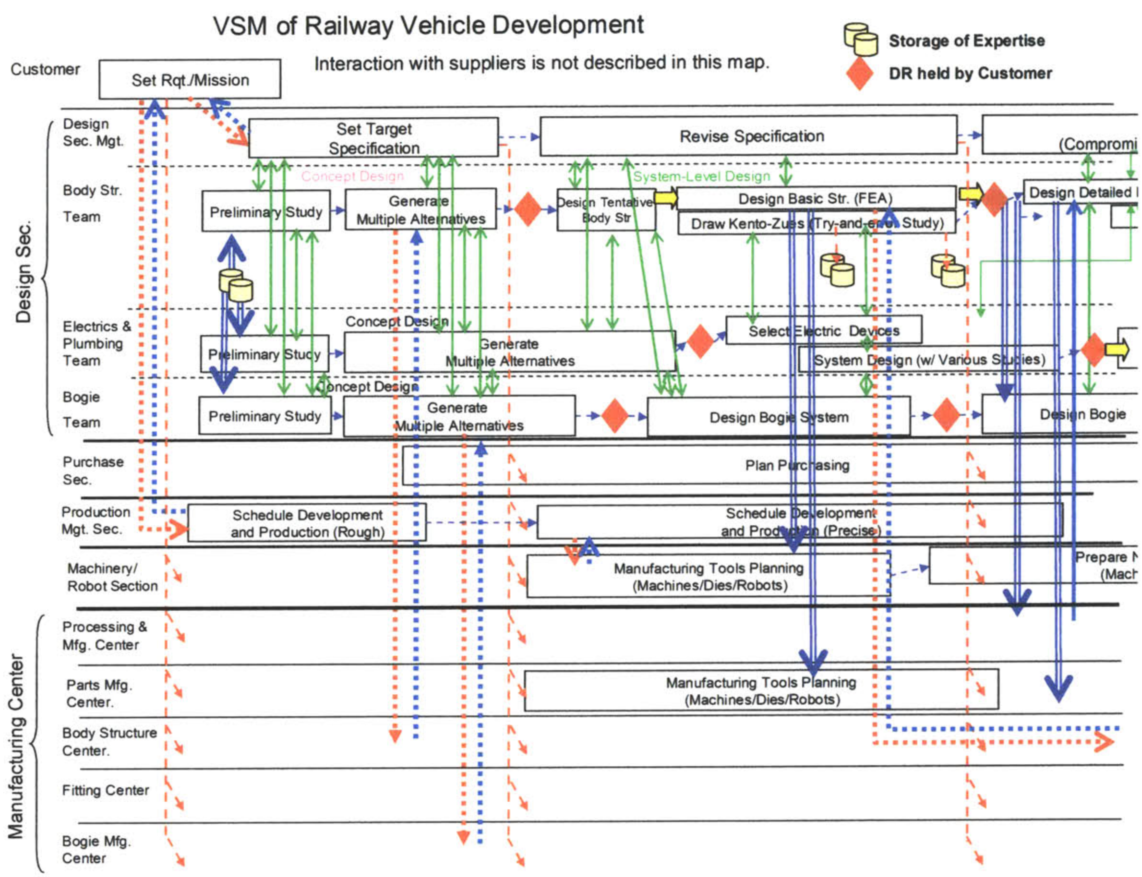


Figure 3.4 Value Stream Map for Whole Railway Vehicle Development Project

3.4 THE THIRD PRELIMINARY CASE STUDY

3.4.1 OBJECTIVE OF THE THIRD CASE STUDY

The objective of this case study was to test the applicability of value stream mapping to measuring waste inside of each task.

3.4.2 SELECTED PROJECT AND FOCUS

A student product development project was chosen as the target of the third case study. The project was a one-semester-long project in which undergraduate students developed a unique product, observing a strictly enforced schedule with several milestones set by faculty. The team consisted of eighteen students. They were divided into two sub teams during the concept development phase, each of them developing different mock-ups. After the "mock-up review" milestone, the more promising one was selected and the sub teams were combined into a big team.

3.4.3 DRAWING A VALUE STREAM MAP

Because the objective of this case study was to measure waste only within process boxes, information flows were not drawn in the value stream map. Instead, both planned and actual processes were drawn to make schedule slips visible. As a result, the value stream map (figures 3.5 and 3.6) became similar to a Gantt chart. The tracked period included four milestone reviews including the mock-up review. Each task had two boxes in the value stream map: the upper one being the original schedule and the lower one being the actual process. Rework processes were distinguished by being hatched. In this case study, process boxes' lengths were precisely made proportional to the period between the start and end dates. Instead of using wastes defined by McManus and Morgan, nine waste indicators introduced in chapter 5 were used.

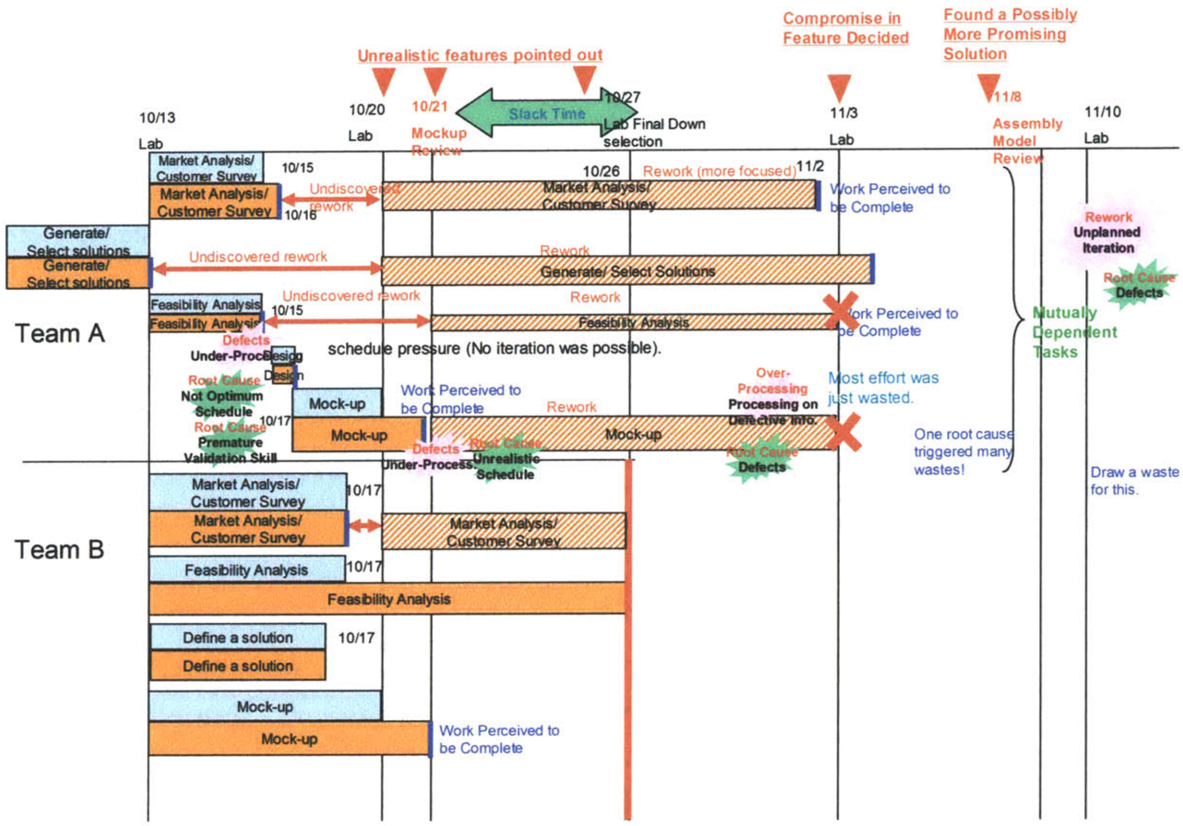


Figure 3.5 Value Stream Map for a MIT Student Product Development Project – Continued to Figure 3.6.

3.4.4 EVALUATION

In this case study, all rework took more time than the original work. One reason for this phenomenon was that scope of the tasks extended as students identified several problems through manufacturing process of a mock-up. However, there was another reason: the students did not work as intensively as they had done in the original work: the tasks had not been worked on all the time. This situation corresponds to today's prevalent product development environment in which the same engineers are shared by some projects. And, generally, the sizes of impacts from outside of the project fluctuate. Possible problems caused by impacts from outside of the project are the following:

1. Project delays due to low availability of engineers.
2. Information loses its value even while it is not worked on because of risks including

market risk (see chapter 8).

Therefore, even when a value stream map's focus is one project, impacts from outside of the focused project should be displayed explicitly somehow.

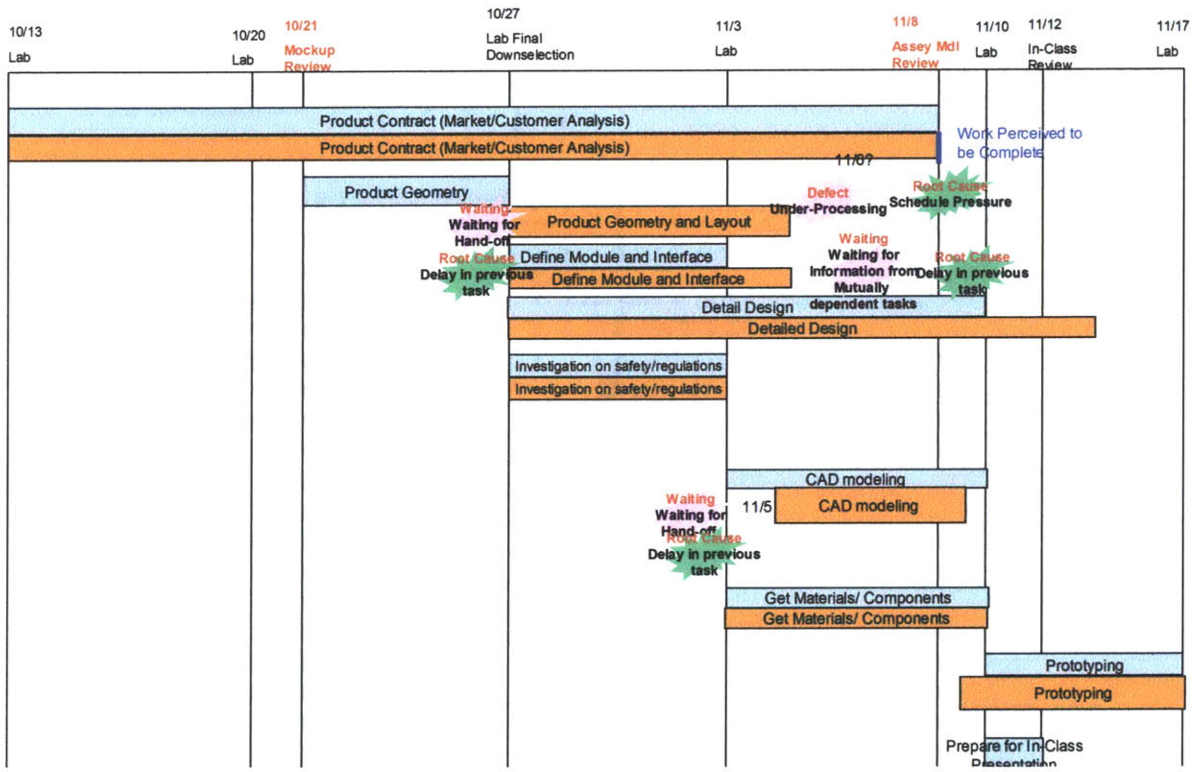


Figure 3.6 Value Stream Map for a MIT Student Product Development Project – Continued from Figure 3.5.

CHAPTER 4 IMPROVEMENT SUGGESTIONS FOR VALUE STREAM MAPPING

4.1 WAY OF DISPLAYING REWORK

The problem about representation of rework (3.2.4) can be addressed by displaying rework with a separate box (figure 4.1). This makes it possible to satisfy the two rules suggested by Morgan (2002) (see 3.2.4). In order to make it easy to identify the original task of the rework, only the process boxes for rework should be along the same line with that of the original task.

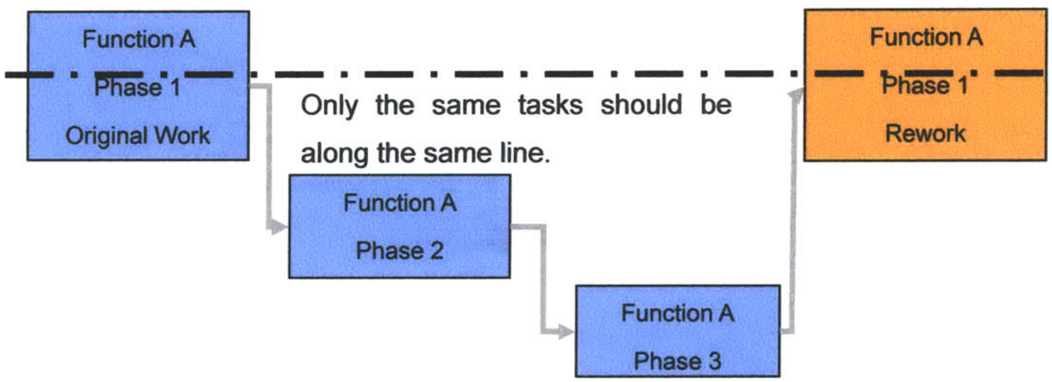


Figure 4.1 Showing Rework in a Separate Process Box

4.2 ADOPTION OF SWIM-LANE FORM

Generally, hand-offs across functional groups take more time and effort than they do within a functional group. In the second preliminary case study, swim-lane value stream mapping could successfully visualize this difference. Adoption of this swim-lane form was also effective for keeping the value stream map organized in spite of high complexity of communications across many functional groups. Additionally, when used in combination with the method suggested in 4.5, Swim-lane value stream mapping made it possible to visualize how unsynchronization happened due to interrupting events.

4.3 SHOWING BOTH PLANNED AND ACTUAL SCHEDLULES

In the third preliminary case study, interviews with students were performed several times. When students were asked if they had found any wasteful activity in their development process with the value stream map displaying only actual processes, it was difficult to get enough information related to waste. After adding the original schedule information to the value stream map, however, it became significantly easier. Thus, showing both planned and actual schedules turned out effective for identifying wasteful activities through interviews.

4.4 DISPLAYING INTERRUPTING EVENTS

As indicated in the third preliminary case study, engineers are sometimes occupied with tasks from outside of the project, causing project delay. Such interruption is one of major sources of project delay as well as waste inside of the project: in order to accurately measure waste derived from activities inside of a project, it is necessary to know whether delay is caused by activities inside of the project or not. Interrupting event can explicitly be shown by using a sign shown in figure 4.2.

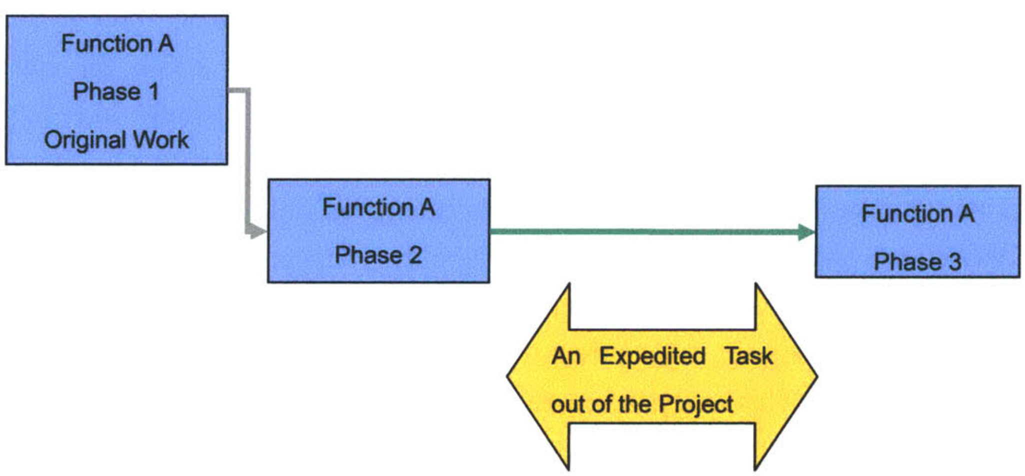


Figure 4.2 Way of Showing an Interrupting Event

CHAPTER 5 ROOT-CAUSE ANALYSIS DIAGRAM

5.1 ANALYSIS OF RELATIONSHIP AMONG DIFFERENT TYPES OF WASTE

Table 5.1 compares several definitions by forerunners. Toyota's seven categories of waste have been revised to address waste in product development processes. Wormack and Jones (1996) modified Toyota's definition by adding two categories, "complexity," and "time lag," and removing over processing. Although these two added categories are not in Toyota's definition, they are the ones that are not peculiar to product development processes. In addition, "time lag" is related to over processing in that time lag causes over processing – time lag means rework discovery time, and long rework discovery time causes over processing (on defects). Morgan dramatically revised the definition by replacing all but "waiting.", based on the systems perspective. Bauch (2004) revised these definitions by analyzing interactions among each category of waste. The total number of all the categories referred to above sums up to twenty three.

5.2 DEVELOPMENT OF ROOT-CAUSE ANALYSIS DIAGRAM WITH NINE PLUS ONE WASTE INDICATORS

Because one of this research's objectives were making it possible to measure waste in product development processes, using all the categories, which were twenty three in total, in Table 5.1 was unrealistic. To deduce a frugal set of categories of waste, the causal relationships among the categories in table 5.1 were analyzed. Figure 5.1 shows an example of this analysis, in which the definitions by Morgan (2002) and McManus (2004) were compared. This figure reveals that most of Morgan's categories were causes for the waste categories defined by McManus, which basically inherited Toyota's seven categories of waste. For example, "lack of system discipline (Morgan)" causes "over production (McManus)," "unsynchronized concurrent tasks (Morgan)," and "ineffective communication (Morgan)." This analysis had been expanded by incorporating other waste definitions and various factors of low performances in product development processes found in papers and books. And, it had been improved through the preliminary case studies discussed in chapter 3. The complete result of this analysis is shown in 5.4: Root-Cause Analysis Diagram.

Table 5.1 Comparison of the Definitions of Waste

	Toyota's seven wastes (Ohno, 1978)	Wormack and Jones (1996)	Morgan (2002)	McManus (2004)	Bauch (2004)
1	Waiting	Wait Time	Waiting	Waiting	Waiting
2	Transportation	Transport	-	Transportation	Transport/ Handoffs
3	Over Processing	-	-	Excessive Processing	Over Processing
4	Inventory	Inventory	-	Inventory	Inventory
5	Defects	Defects	-	Defects	Defects
6	Motion	Movement	-	Unnecessary Motion	Movement
7	Over Production	Overproduction	-	Over Production	Overproduction/ Unsynchronized Processes
8	-	Complexity	-	-	-
9	-	Time Lag	-	-	-
10	-	-	Hand-Offs	-	Transport/ Handoffs
11	-	-	External Quality Enforcement	-	-
12	-	-	Transaction	-	-
13	-	-	Re-invention	-	Re-Invention
14	-	-	Lack of System Discipline	-	Lack of System Discipline
15	-	-	High Process and Arrival Variation	-	-
16	-	-	System Over Utilization	-	-
17	-	-	Expediting	-	-
18	-	-	Large Batch Sizes	-	-
19	-	-	Redundant Tasks	-	-
20	-	-	Stop-and-Go Tasks	-	-
21	-	-	Unsynchronized Concurrent Tasks	-	-
22	-	-	Ineffective Communication	-	-
23	-	-	-	-	Limited IT Resources

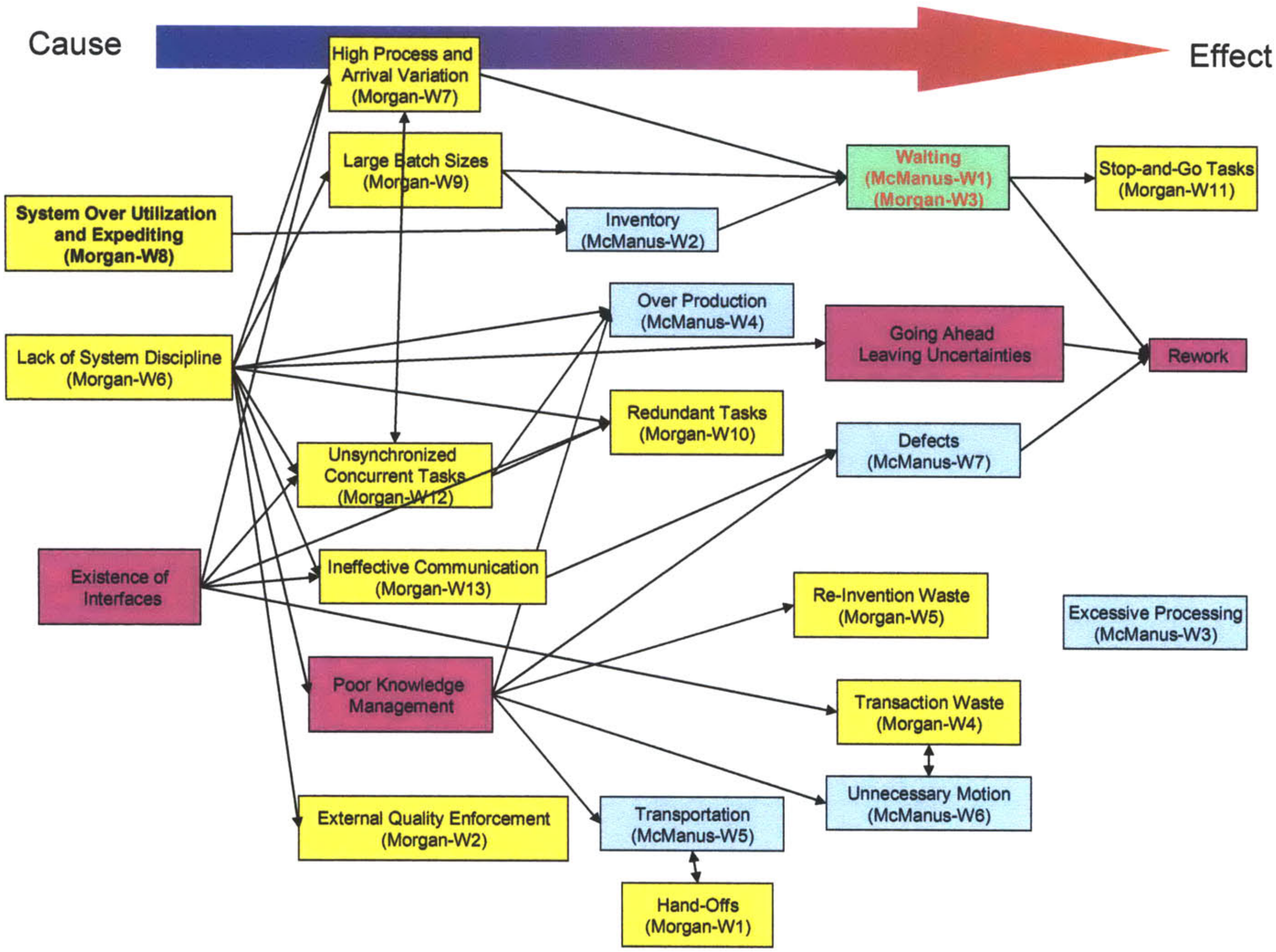


Figure 5.1 Analyses of Relationships among the Categories of Waste Defined by Morgan and McManus – The Categories without the Name of Morgan or McManus were added by the Author.

5.3 DEFINITIONS OF NINE WASTE INDICATORS AND INVENTORY OF INFORMATION

5.3.1 DEDUCING NINE WASTE INDICATORS FROM THE ROOT-CAUSE ANALYSIS DIAGRAM

The root-cause analysis diagram is shown in 5.4. The rightmost categories in the diagram are root causes, and leftmost ones, effects. From the perspective of measurement, desirable categories of waste are the ones that can easily be identified and the wasted time measured. It was found that effects are easier to measure than their causes. For example, wasted time on "waiting" can be measured by measuring an engineer's waiting periods. However, wasted time on "system over utilization" is difficult to measure; "System over utilization is the root cause for waiting in figure 5.1. Therefore, the nine leftmost categories, which are effects, were chosen as metrics of waste in product development processes. They are named waste indicators because they are not causes for waste, but indicate that time is wasted for some reasons.

<u>NINE WASTE INDICATORS</u>

1. OVERPRODUCTION
2. WAITING
3. TRANSPORTATION
4. OVER PROCESSING
5. MOTION
6. REWORK
7. RE-INVENTION
8. HAND-OFF
9. DEFECTIVE INFORMAITON

Rework is the one that had neither been identified by the forerunners in Table. 5.1. It was added by the author because it has frequently been pointed out as indicator of low performances by many scholars and product development practitioners.

As can be understood from 5.4, the entries in the nine waste indicators are not mutually exclusive. For example, defective information causes rework (5.4.5). The reason why the author allowed this redundancy is the existence of strong interdependency among waste in product development: one occurrence of waste can cause many different types of waste, possibly forming a vicious circle, and this interdependency differs on a case-by-case basis. Therefore, reducing one waste indicator may make the set of waste indicators an insufficient one. Thus, the author maintained the nine (plus one described in 5.3.2) waste indicators at this point; the waste indicators are prioritized in the case studies discussed in chapter 10.

Table 5.2 Definitions and Examples of Nine Waste Indicators

Waste Indicators	Description	Typical Examples
1. Overproduction of Information (Duplication)	Different people/groups are unintentionally creating the same information.	-Duplicate creation of information due to unclear division of labor
2. Waiting of People	People are waiting.	-People are forced to wait because of delay of upstream tasks.
3. Transportation of Information (Preparing and forwarding information)	Information is in transportation.	-Paper mail, packages -Tardy approval process with multiple signatures.
4. Over Processing	Engineers create information that won't contribute the value of product.	-Creating information based on defective data. -Trying to design beyond target specifications
5. Motion of People (Information hunting, travel, reviews, documentation, and meetings)	People have to spend time on non value-adding motions.	-Manual data conversion -Business trips
6. Rework	Redoing tasks perceived to be finished for some reason	-Correcting/Revising designs that failed to pass Reviews. -Updating completed information due to requirement changes
7. Re-Invention	Designing similar things without utilizing past experience.	-Design similar thing twice because past designs are not well documented.
8. Hand-Off (Hand-off inside of project)	Information is handed off with its responsibility between two groups/people.	-Hand-off of information to downstream designers.
9. Defective Information (Coupled to Over Processing and Rework)	Erroneous or incomplete information.	-Design not feasible -Information that does not meet the requirements (final, milestone, etc).

5.3.2 INVENTORY OF INFORMATION

Inventory of information is different from the nine waste indicators in that engineers' time is not wasted while information is inventoried. In spite of this, inventory of information was also identified as one waste indicator in this research because inventoried information can lose value, causing rework. Inventory of formation is discussed in detail chapter 8.

5.4 ROOT-CAUSE ANALYSIS DIAGRAM FOR NINE WASTE INDICATORS

The complete root-cause analysis diagram is shown in Appendix II

CHAPTER 6: VALUE STREAM MAPPING FOR QUANTITATIVE ANALYSIS OF PD PROJECTS

6.1 OBJECTIVE

Value stream mapping was originally developed for use in manufacturing environment, and later its scope was expanded to product development environment as PDVSM (Product Development Value Stream Mapping). For this historical reason, PDVSM took over various rules for displaying different types of information flows and activities that had been developed for detailed analyses of manufacturing processes. Some of these rules in PDVSM are unnecessary in this research because the main purpose of using value stream mapping is measuring wasted time. Unnecessary rules are eliminated, and, instead, some minimum set of rules necessary for measurements are introduced in this chapter based on the suggestions discussed in 4.2.

6.2 VALUE STREAM MAPPING (VSM) FOR DETAILED ANALYSIS OF PROJECTS

6.2.1 TIME LINE AND GRIDS

This value stream mapping is similar to a cross-functional process chart in that it has process boxes and arrows that represent flows (figure 6.1). One of the major differences from a cross-functional process chart is that the horizontal axis of the value stream map is a time line with weekly gridlines.

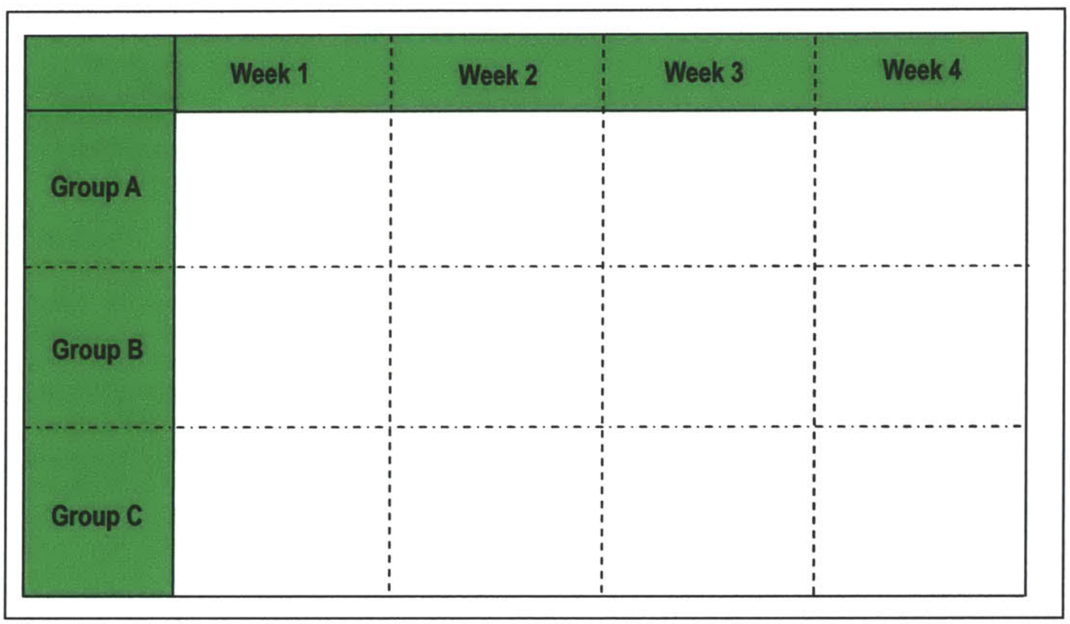


Figure 6.1 Time Line and Grids

6.2.2 PROCESS BOXES – ALLOCATION AND LENGTH

(1) Process Boxes for the Development of the Same Function

If development of a function can be divided into several different phases, each phase should be assigned one process box (figure 6.2). As can be seen in this figure, downstream phase for a function is allocated immediately below its adjacent upstream phase

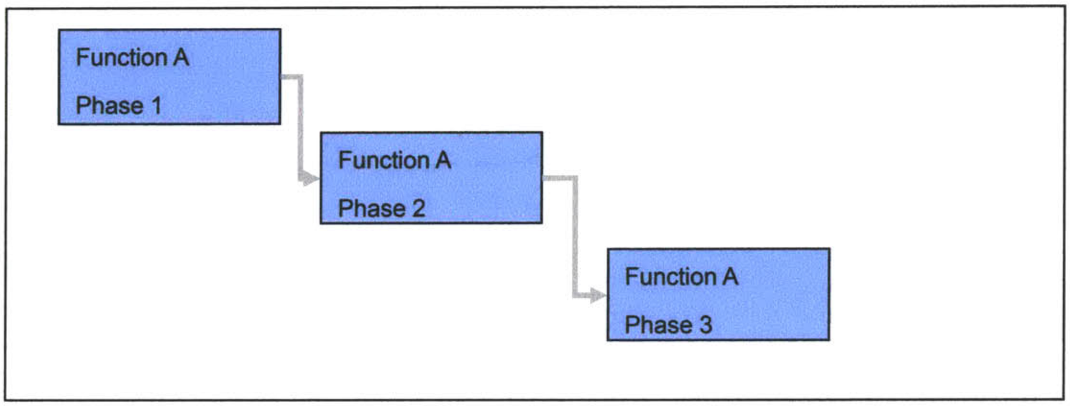


Figure 6.2 Allocations of Adjacent Phases

(2) Process Boxes for Rework

Process boxes for rework should be allocated on the same line of the original tasks (figure 6.3).

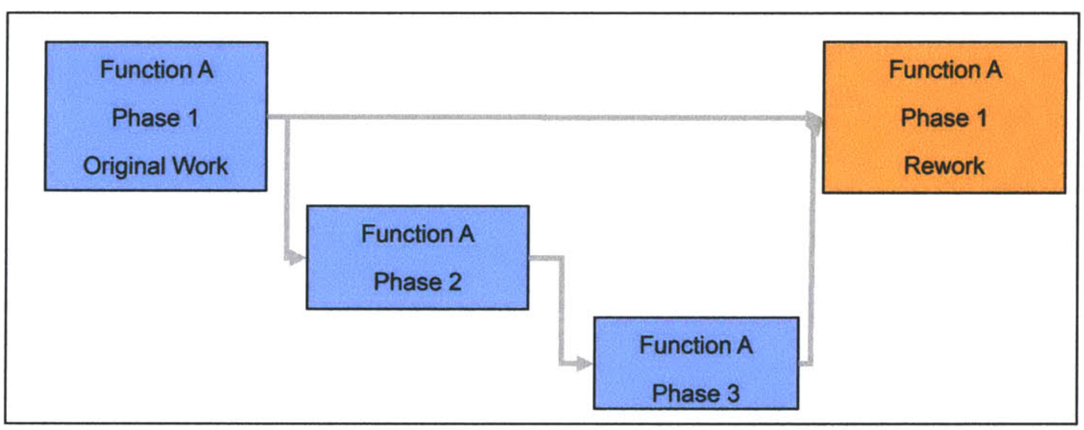


Figure 6.3 Allocation of a Process Box for Rework

This rule does not apply to the process boxes for different functions (figure 6.4). Different tasks can be on the same line if they are not for the same function.

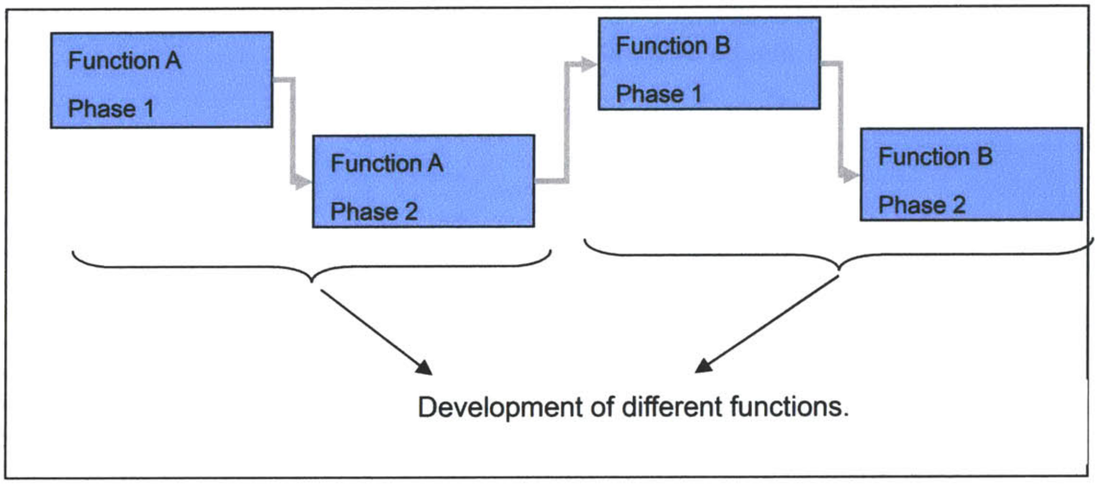


Figure 6.4 Allocations of Process Boxes for Different Functions

(3) Beginnings and Ends of Process Boxes

Beginnings and the ends of the process boxes should be consistent with the time line at the top of the map (Morgan, 2002). As a result, process boxes' lengths become proportional to the time spent on them.

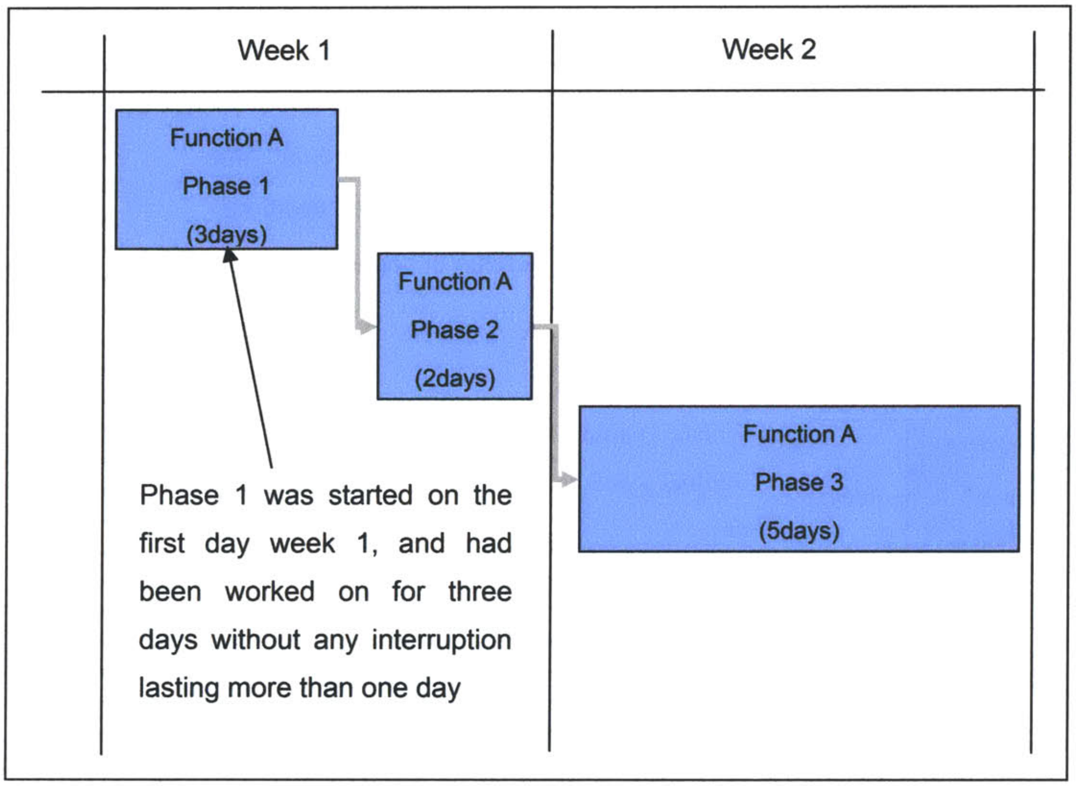


Figure 6.5 Beginnings and Ends Process Boxes– Beginnings and Ends of the Process Boxes Should Match the Time Line at the Top of the Value Stream Map

6.2.3 BOXES – COLOR CODES

(1) Displaying Actual Processes (figure 6.6)

Process boxes for actual processes are painted in blue as long as their length do not exceed the scheduled periods. On the other hand, in cases in which tasks took longer than scheduled, the excess time should be shown by painting in pink.

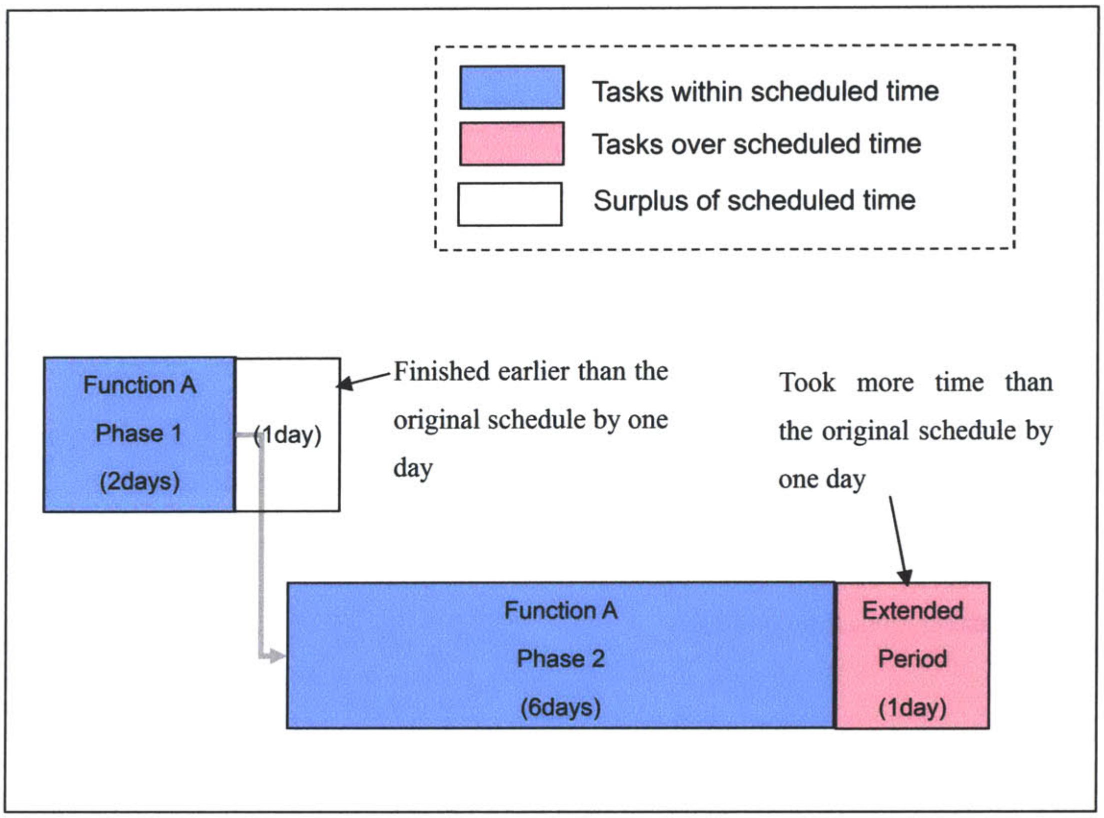


Figure 6.6 Blue, White, and Pink Boxes

(2) Displaying Original Schedules

There are two ways to show original schedules. One way is allocating white boxes that show original schedules (figure 6.7). This method is effective for clearly visualizing schedule slippage from original plans. In this research, this method was applied to the case study in the project B (see chapter 9). However, in cases in which schedule slippage is significant, or processes are interrupted frequently, this method makes value stream maps too complicated.

Another usage of white boxes is using them only when tasks are finished earlier than scheduled; in figure 6.6, a white box is used for showing that phase 1 was finished earlier before its due date by one day. This method is applied to case studies in projects A and C (see chapter 9).

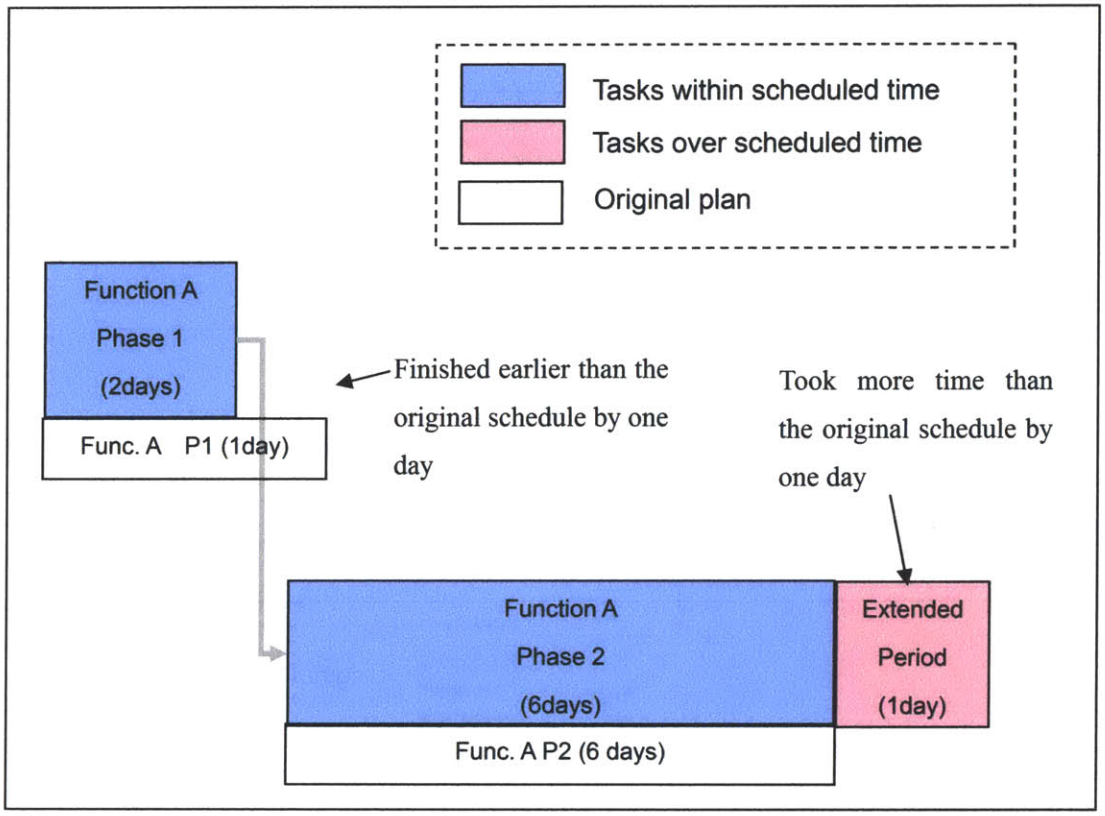


Figure 6.7 Using White Boxes for Showing Original Schedule

(3) Displaying Review and Testing Processes

Review and testing processes should be shown by using diamonds (figure 6.8).

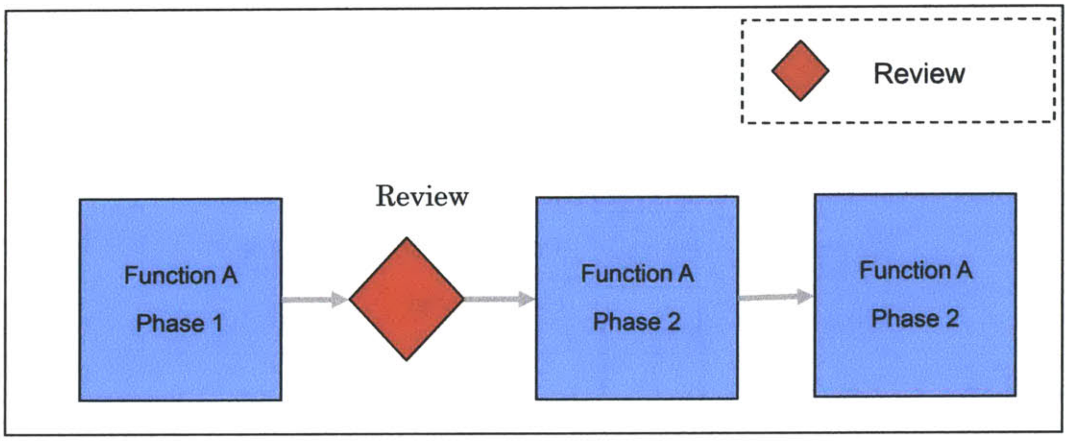


Figure 6.8 Displaying Review and Testing Processes

(4) Displaying Rework

Rework should be painted in orange (figure 6.9).

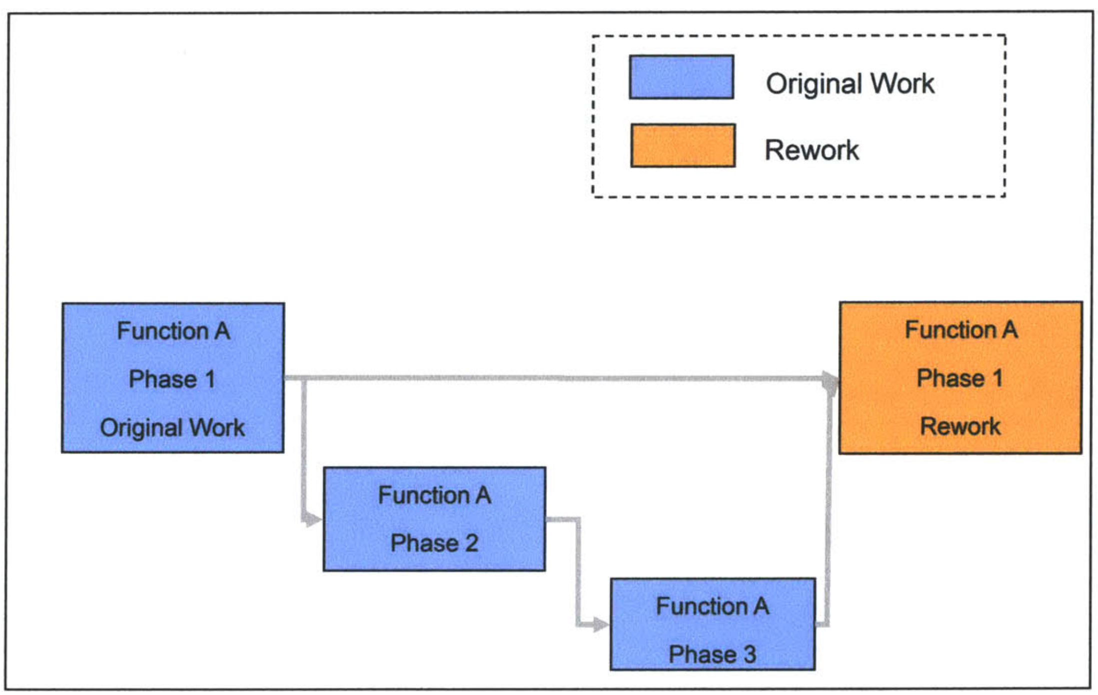


Figure 6.9 Displaying Rework

(5) Displaying Over Processing

Usually over processing is submerged in value adding activities, but when the whole output of a process is considered over processing, its process box should be painted in yellow (figure 6.10)

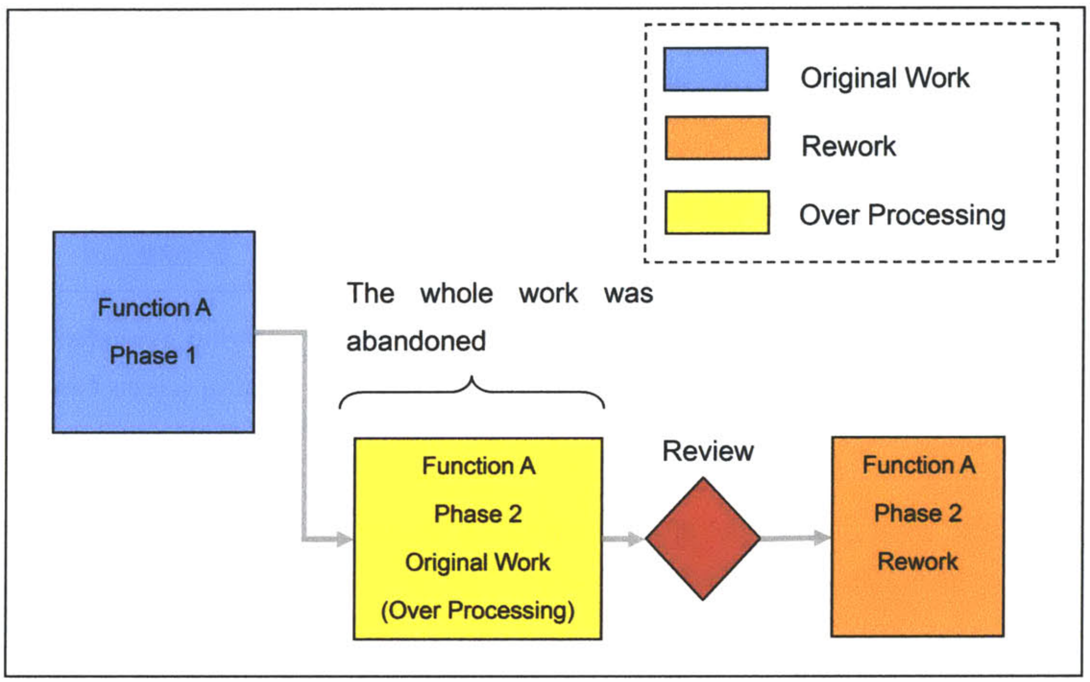


Figure 6.10 Displaying Over Processing

6.2.4 ARROWS – COLOR CODES

(1) Information flow inside the same swim lane

Timeliness of information transfer is one measure of information quality defined by Bauch (2004). Information flows inside the same swim lane should be shown with gray lines if the transferred information is used in a timely manner (typically within a day: this criterion is contingent with various factors such as the project's scheduled period and the market's mobility). If the information is kept untouched for a specific period such as one day or more, the information transfer is considered as inventory, and it should be shown with green lines (figure 6.11).

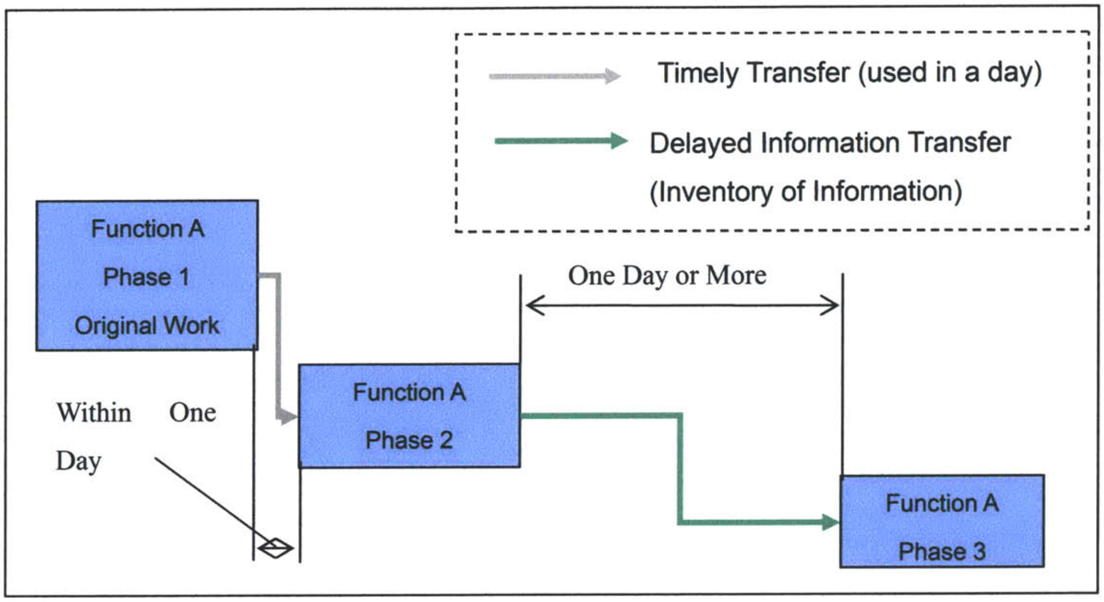


Figure 6.11 Timely Information Transfer and Delayed Information Transfer – Information Stored for a Specific Period is Distinguished by Using Green Lines

This rule is also applicable for the information stored because the group or the engineer is interrupted in the middle of a task (figure. 6.12).

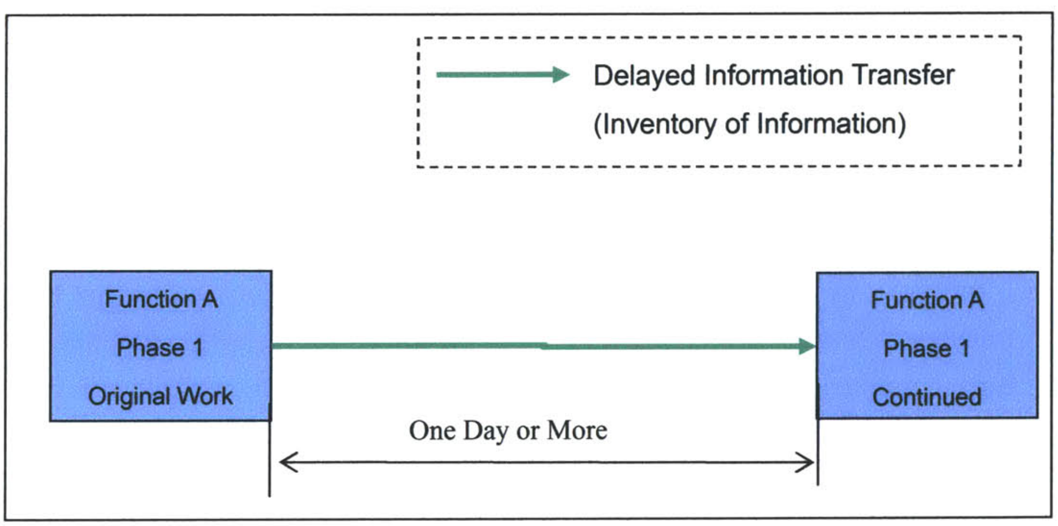


Figure 6.12 Displaying an Interruption

(2) Information transfer with Hand-Off (Information Flow across Swim Lanes)

Hand-Offs (information transfer among engineers/ groups) are marked in blue if the information is handed off immediately (figure 6.13). Typically, transferred information used in one day satisfies this condition. However, this criterion is contingent on various factors including as the market's mobility and the scheduled development period. Hand-Off in which information is kept untouched for one day or more is wasteful and is covered by two waste indicators, hand-off and information inventory. Hand offs with inventory periods should be distinguished from the other hand-offs by using red lines (figure 6.13).

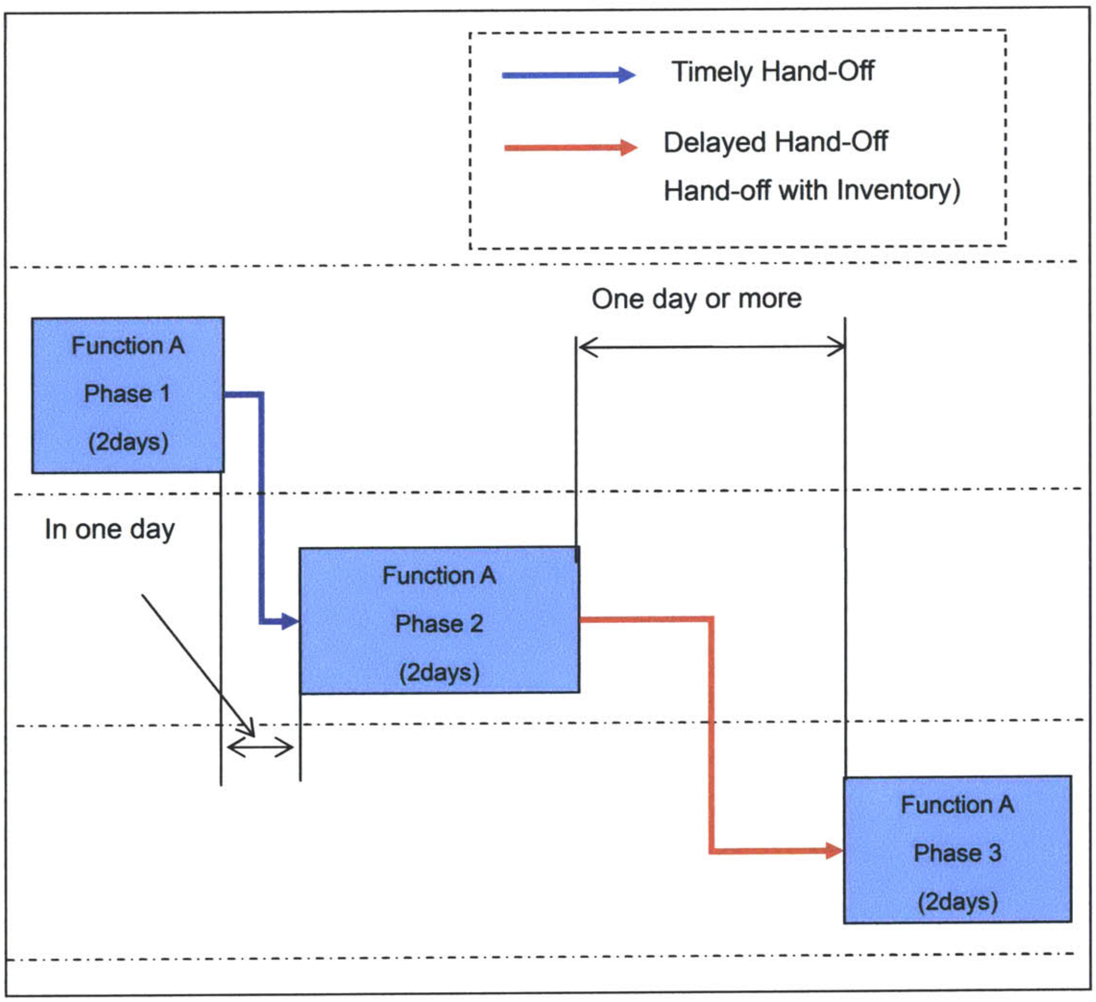


Figure 6.13 Different Types of Hand-Offs

6.2.5 INTERACTIONS

Bi-directional information exchanges are shown by using bi-directional arrows (figure 6.14)

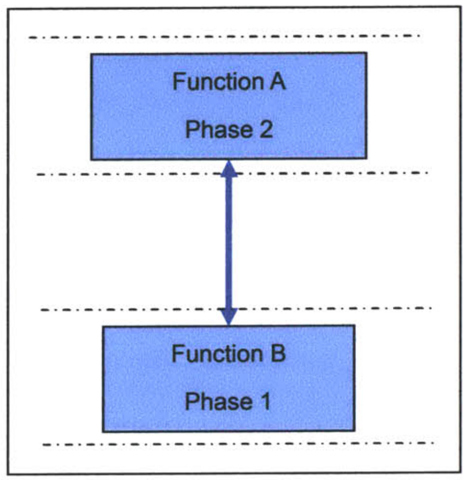


Figure 6.14 Displaying Bi-directional Information Exchanges

6.2.6 CROSS-FUNCTIONAL TASKS

Cross-functional tasks should be shown in the way described in figure 6.15.

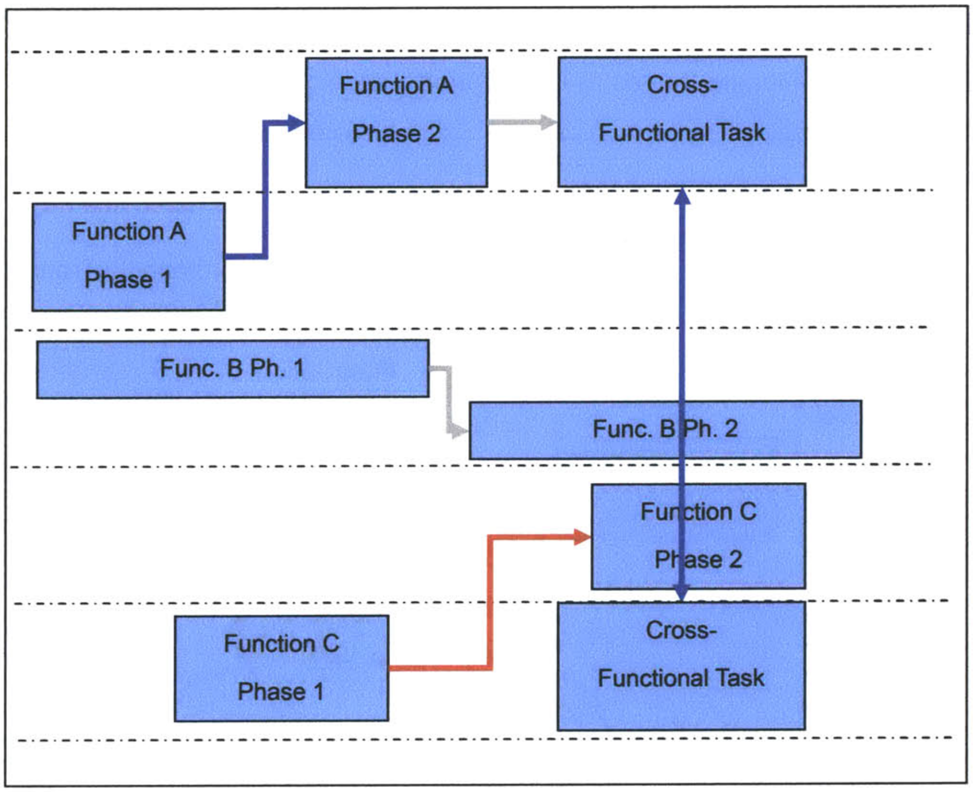


Figure 6.15 Displaying Cross-Functional Tasks

6.2.7 INTERRUPTIONS

In most organizations, engineers are required to work on multiple tasks. Many scholars and product development practitioners have pointed out that this multiple tasking significantly affects product development projects. In this value stream mapping, all tasks that are not part of the focused project are treated as interruptions, and shown as in figure 6.16.

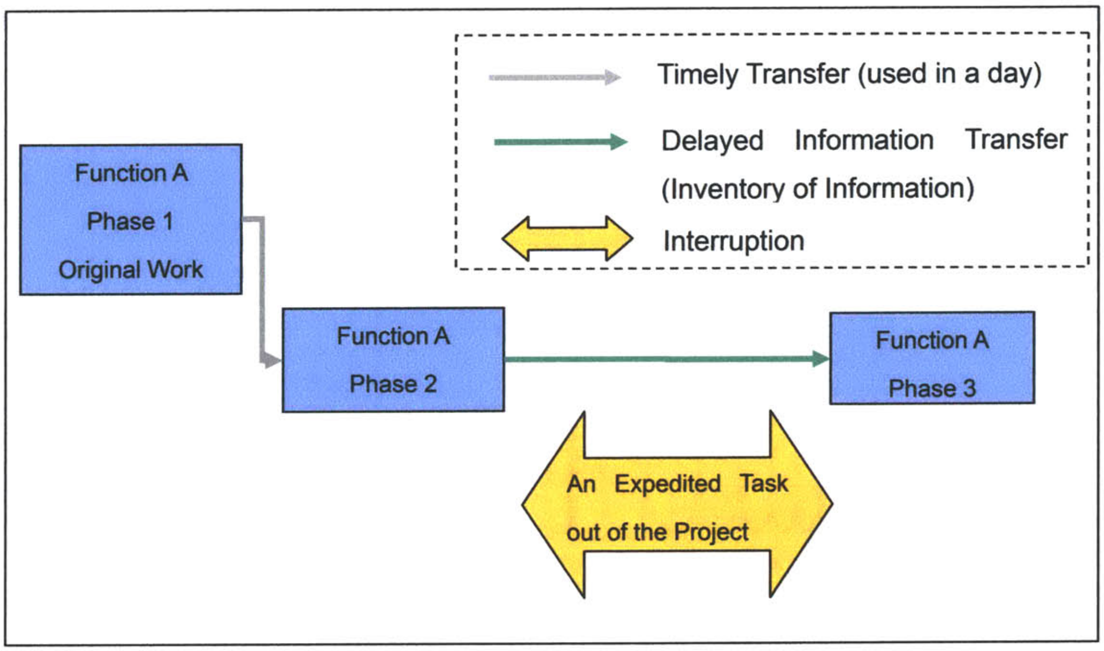


Figure 6.16 Displaying Interruptions

6.2.8 TIME RECORDING

Time spent on every task should be put on a value stream map (figure 6.17). The total time of weekly hours of labor of each functional group should also be put on a value stream map. These numbers will be used for measuring wasted time (chapters 7 and 8).

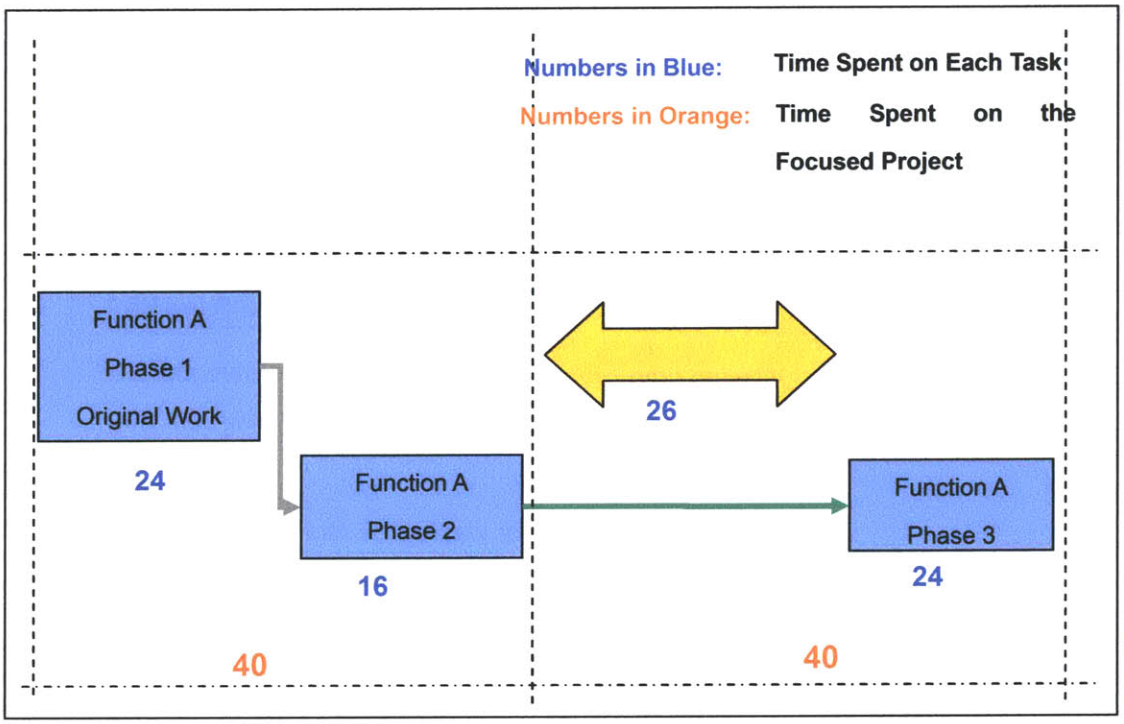


Figure 6.17 Displaying Spent Time

6.2.9 WASTE INDICATORS

Measured wasted time should be put on value stream maps in the way described in figure 6.18.

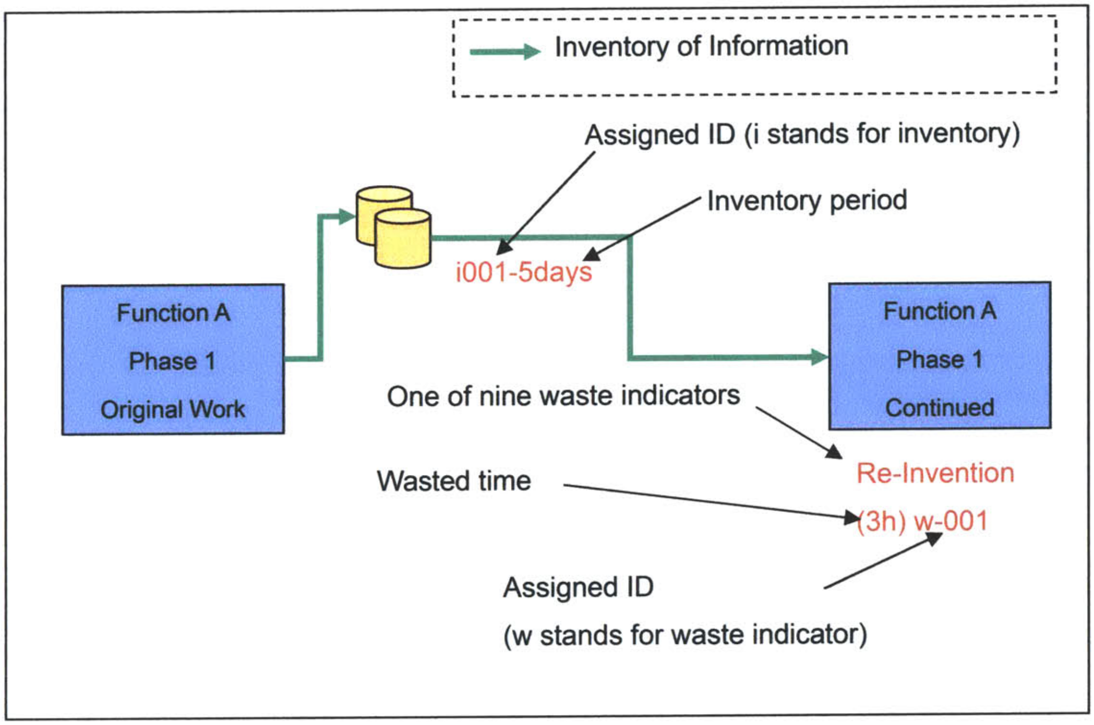


Figure 6.18 Displaying Wasted Time

CHAPTER 7: HOW TO MEASURE WASTED TIME IDENTIFIED BY NINE WASTE INIDICATORS USING VALUE STREAM MAPPING

7.1 INTRODUCTION OF THIS CHAPTER

This chapter describes ways for measuring wasted time that is detected by nine waste indicators defined in 5.3. All the wasted time on waste indicators are measured in units of engineering time.

7.2 OVERPRODUCTION

When overproduction occurred, engineering time spent on overproduction is regarded as time wasted on overproduction (figure 7.1).

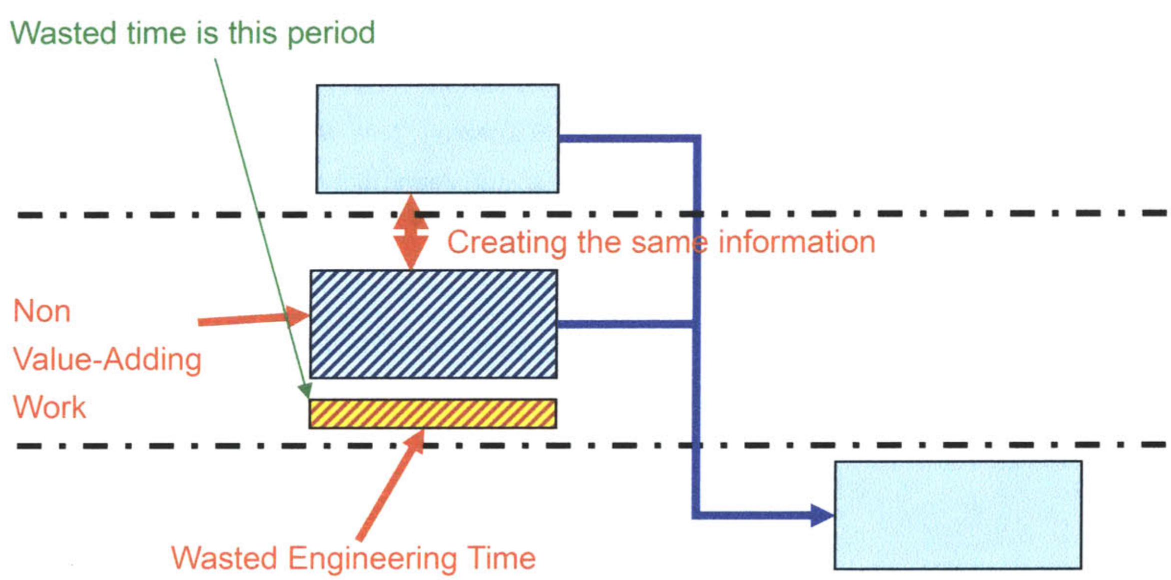


Figure 7.1 Measuring Time Spent on Overproduction – A Hatched Process Box Mean Overproduction

7.3 WAITING

When an engineer was forced to wait doing nothing, the period for which the engineer had waited is regarded as time wasted on waiting (figure 7.2). Waiting is rare in today's product development environment, for engineers usually have several tasks in their cues.

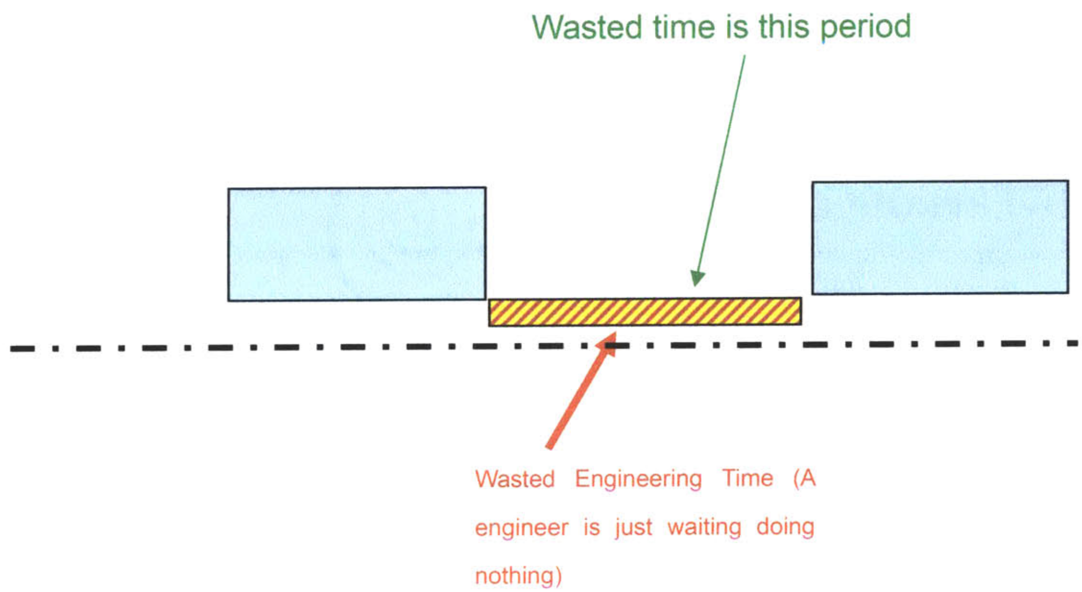


Figure 7.2 Measuring Time Spent on Waiting

7.4 TRANSPORTATION OF INFORAMTION

Sometimes transportation of information takes up a substantial amount of engineers' time. Figure 7.3 is an example case in which an engineer needs to provide his/ her CAD data to a supplier. He/ she may need to spend his/ her time on data conversion processes, which is usually not completely automatic. In this case, time the engineer spent on data conversion is wasted time on transportation.

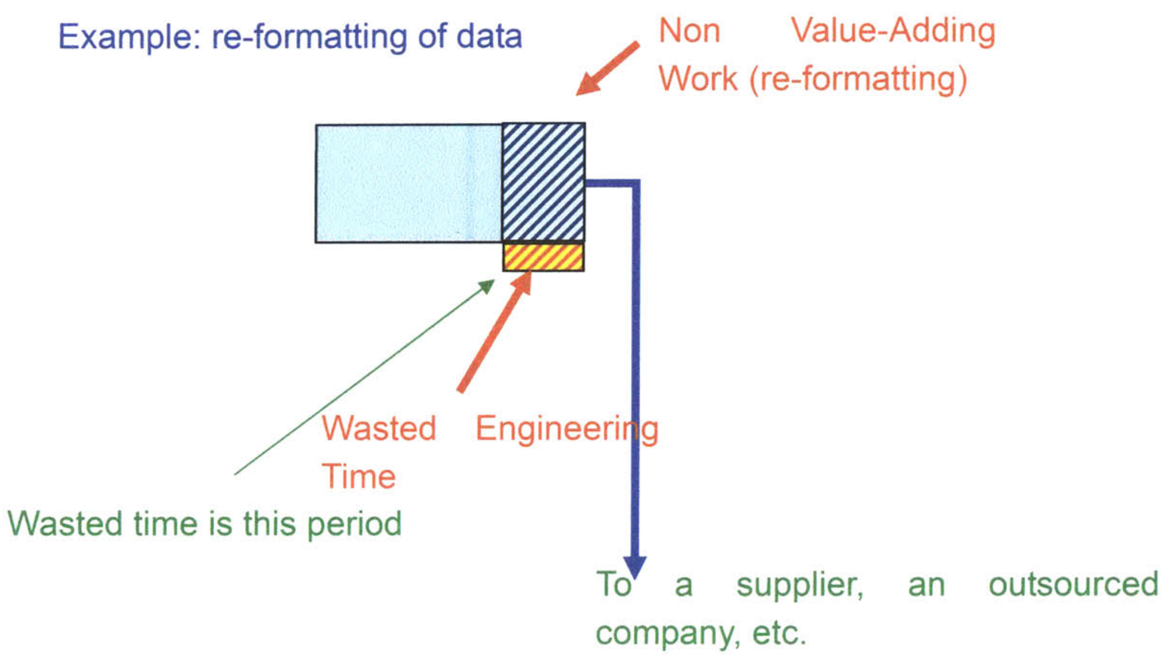


Figure 7.3 Measuring Time Spent on Transportation

7.5 OVER PROCESSING

Wasted time on over processing can be measured in the way shown in figure 7.4. Determination of the actual time spent on over processing usually requires intensive interviews with engineers, for over processing occurs concurrently with other value adding work.

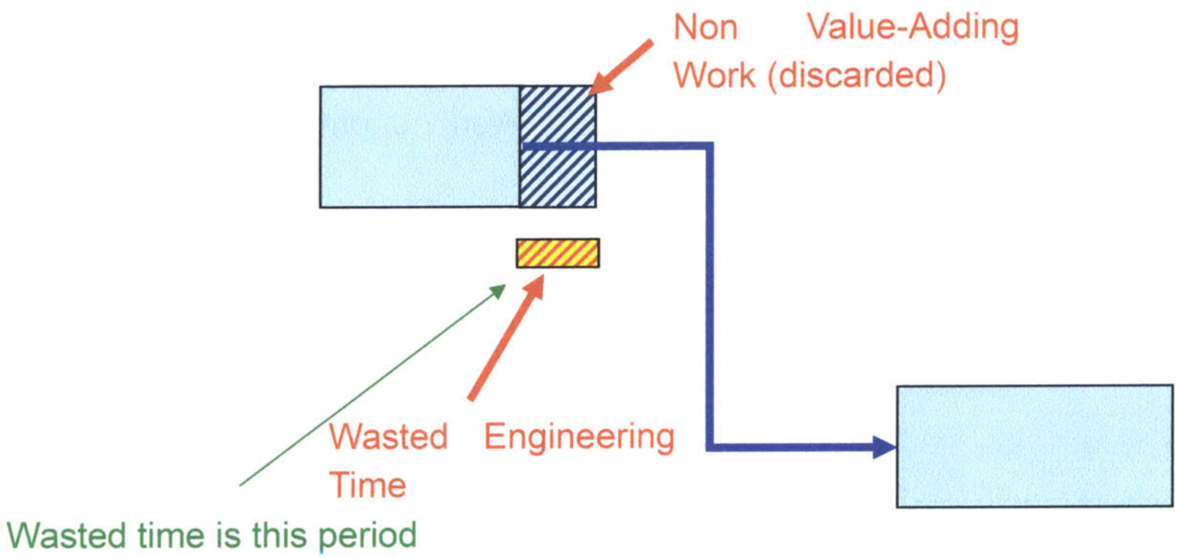


Figure 7.4 Measuring Time Spent on Over Processing

7.6 MOTION

Figure 7.5 is an example of motion. In this example, engineer spent some time on reviewing another engineer's work. Time spent on reviewing is considered to be wasted time.

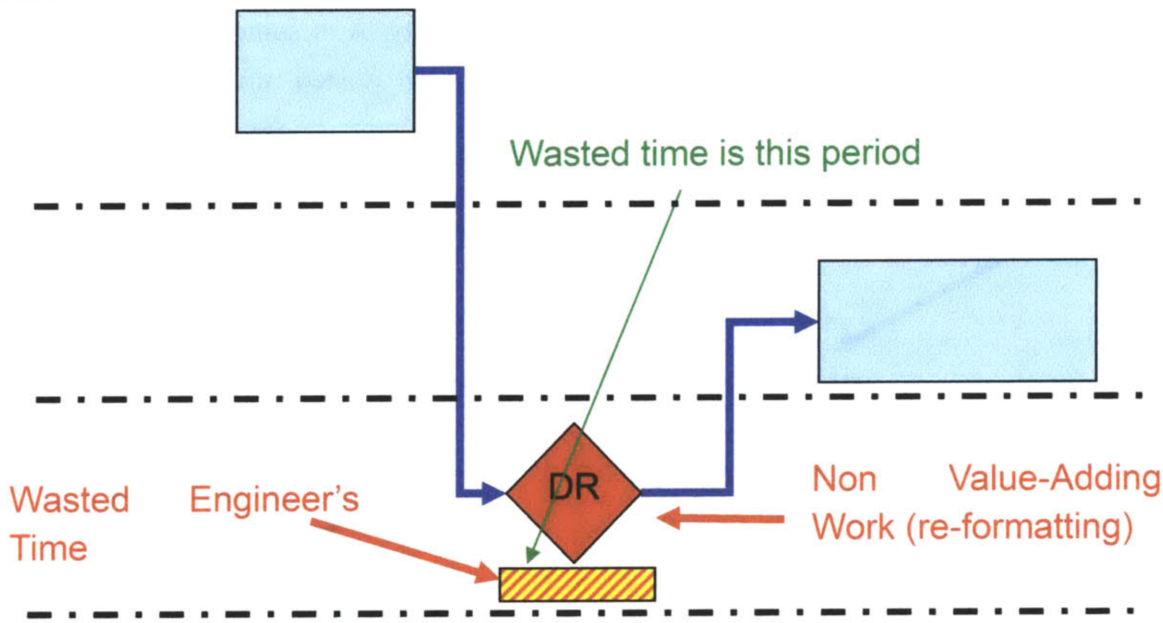


Figure 7.5 Measuring Time Spent on Motion

7.7 REWORK

Figure 7.6 is an example of measuring wasted time on rework. In this example, the original work was partially reworked. In such a case, wasted time on rework should be the total time of A and B (figure 7.6). A is time spent on discarded work, and B, time spent on troubleshooting. C is considered to be value adding activity. A is sometimes difficult to measure, but C can substitute for A when measuring wasted time. Examples of measurements of rework is shown in figure 10.20.

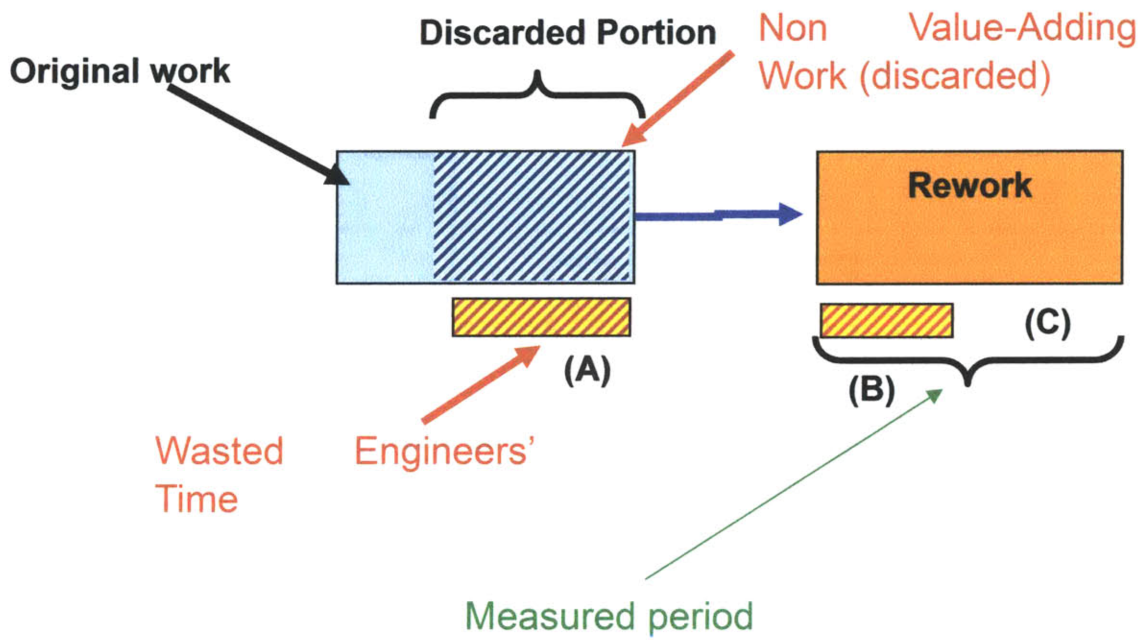


Figure 7.6 Measuring Time Spent on Rework

7.8 RE-INVENTION

Figure 7.7 is an example in which two engineers invented the same information. In this case, time spent on the second invention is regarded as the time wasted on re-invention.

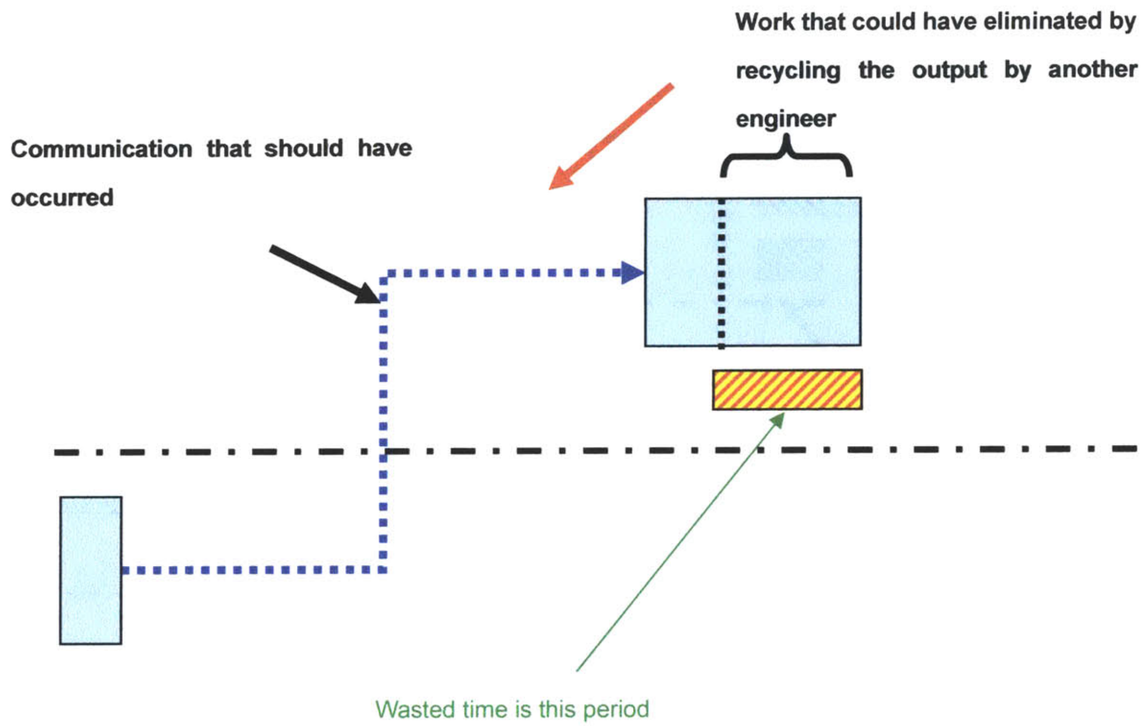


Figure 7.7 Measuring Time Spent on Re-Invention

7.9 HAND-OFF

Sometimes hand-offs takes both the sender's and the receiver's time: the sender may need to spend his/her time on documentation that could be avoided without hand-off, and the receiver usually needs to spend his/her time on understanding the sender's work. Figure 7.8 is an example in which both engineers wasted time on hand-off.

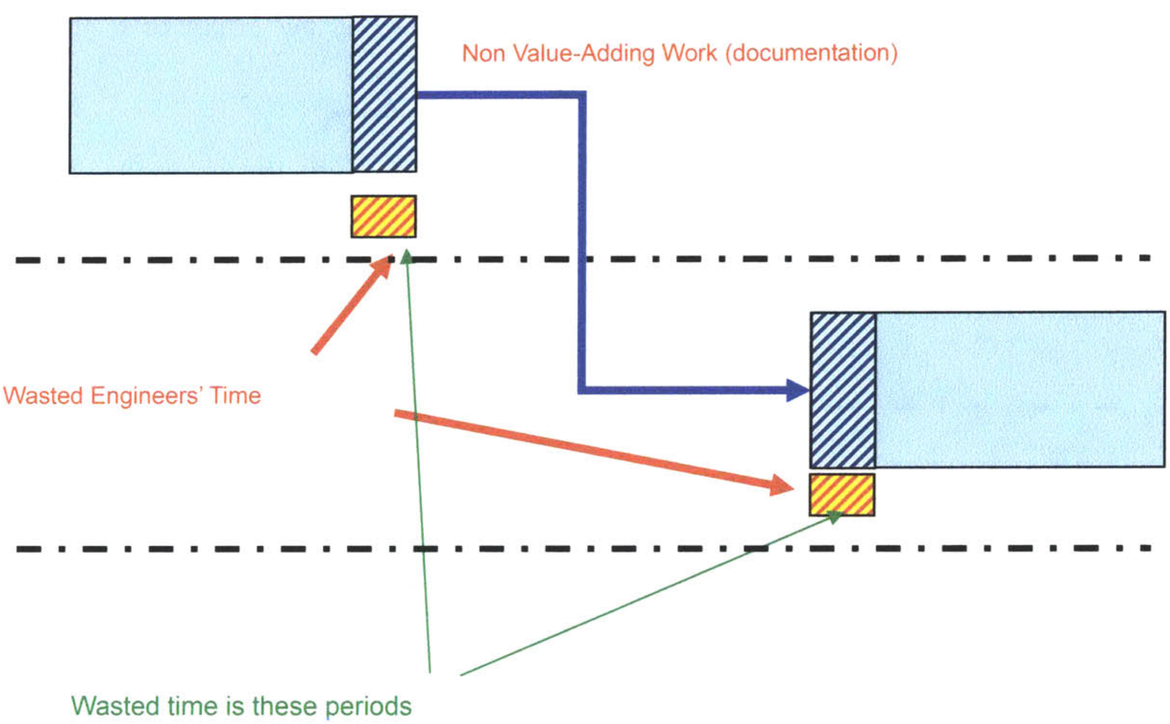


Figure 7.8 Measuring Time Spent on Hand-Off

7.10 DEFECTIVE INFORMATION

Defective information causes waste of time in various forms including rework, time spent on reviews and testing, and customer support work after launching the product. Figure 7.9 is an example in which defective information caused rework. In this case, wasted time is the time spent on creating defective information and fixing it. In many cases, time spent on creating defective information cannot be easily distinguished from other vale-adding activities. In such cases, measuring time on fixing defective information is sufficient.

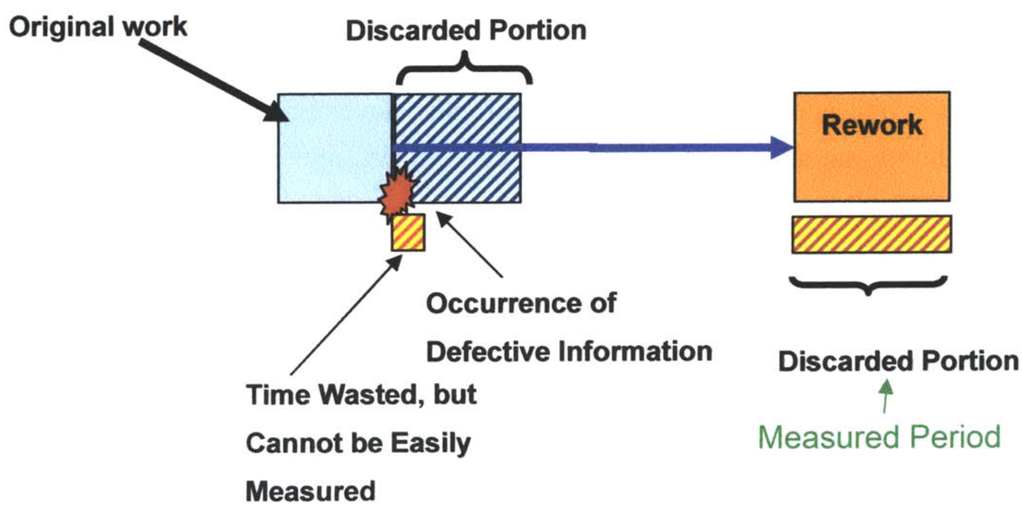


Figure 7.9 Measuring Time Spent on Defective Information

CHAPTER 8: INVENTORY OF INFORMATION AND HOW TO MEASURE IT USING VALUE STREAM MAPPING

8.1 INTRODUCTION

Goldratt (1997) insists in his book "Critical Chain" on not allocating buffer times except at the ends of projects. McManus (2004) puts stress on wastefulness of inventory of information in his "PDVSM Manual," arguing "work in progress" information may become obsolete while it is stored. Both arguments share the common idea that created information should not be kept waiting. However, alike in manufacturing environment, inventory cannot be completely eliminated for two reasons. One is that product development teams usually do not have enough numbers of engineers to keep all information busy all the time. The other is the risks and uncertainties existing in product development projects. This is why even the scheduling methodology suggested by Goldratt requires buffer time allocated on feeding paths.

Thus, there exist two tradeoffs related to inventory of information. One exists between cost of having a big team and cost of having obsolete information caused by having a small team. The other exists between the risk of depleting buffer time, and, again, the risk of having obsolete information. Depletion of buffer time unsynchoronizes the whole project schedule, causing subsequent waste.

Because of these trade-offs, there can be no universal solution for determining the right number of engineers and the right buffer times: product development organizations need to know how much inventory of information costs in their specific contexts. Without quantitative data, they cannot optimize buffer allocations in their schedules. For instance, in an environment in which market is significantly unstable, a huge team that realizes short development cycle times may be desirable because information created goes bad quickly. This research tries to shed light on this topic: the deterioration of information inventory and how to measure it. 8.2 discusses how information goes bad. "Interest rate" of inventory of information is calculated in the case study of Project A; the result is discussed in chapter 11.

8.2 DEFINITIONS OF ROTTEN INFORMATION AND FRESH INFORMATION

Rotten inventory in this thesis is the information inventory that needs to be reworked partially or completely due to changes occurred inside or out of the project. For example, information inventory may need to be reworked because a significant market change is identified. More discussions on causes of rotten information are covered in 8.3.

8.3 HOW INFORMATION GETS ROTTEN

8.3.1 MARKET CHANGE

In some markets, customers' preferences change so quickly that products can be obsolete in one year. This means that the specifications set at the beginning of projects may become obsolete before finishing the projects. Even when customers' preference is consistent, a product loses its value when a competitor releases a product with similar features because most of the leading customers are unlikely to buy the second product.

8.3.2 REQUIREMENTS CHANGE

Shifting requirements are also causes for rotten information. Work-in-progress information may become obsolete by changes in requirements: a fighter's specification is unlikely to be consistent for ten years. Requirements changes may also be caused by internal events such as boss change.

8.3.3 TECHNICAL DIFFICULTIES

Shenhar et al. (2003) argues that overlapping among tasks should be less when technical risk/ uncertainty levels are relatively high. This implies that concurrent engineering's applicability is contingent on the risk/ uncertainty levels. In concurrent engineering, downstream tasks are sometimes started with tentative information from upstream tasks. Working on tentative information may cause rework because the tentative information may turn out to be defective for various reasons like technical difficulties. In such a case, some portion of the original work becomes rotten, causing rework (figure 8.1).

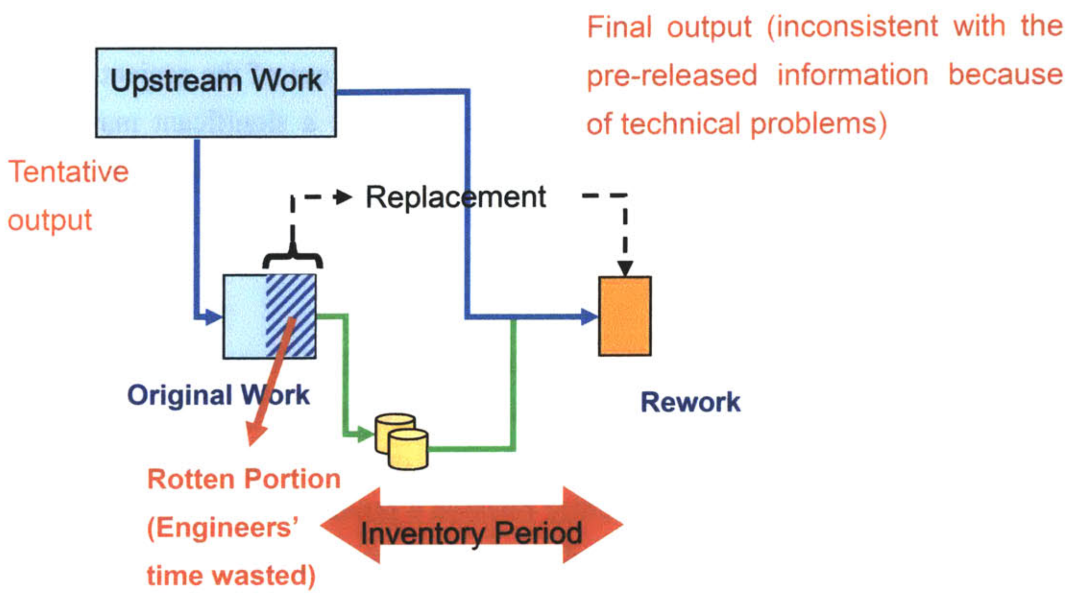


Figure 8.1 Rotten Information Due to Technical Risk

CHAPTER 9: OVERVIEW OF THE RESEARCH INVESTIGATIONS

9.1 RESEARCH QUESTIONS

The lean tools and processes are developed through readings and preliminary case studies. However, it was still uncertain that the tools and processes are applicable for measurements in industrial product development projects. Specifically, the following questions were raised.

(1) NINE WASTE INDICATORS

- **Are they sufficient to address all the waste in product development processes?**
- **Are all the waste indicators equally prevalent, or some are more important than the others?**

(2) INVENTORY OF INFORMATION

- **To what extent inventory of information is prevalent in industrial product development processes?**
- **How quickly inventoried information gets rotten?**
- **How much labor does it take to refresh rotten information?**

(3) THE VALUE STREAM MAPPING FOR QUANTITATIVE MEASUREMENT

- **Can the value stream mapping be applicable for measuring wasted time on every waste indicator defined in this research?**

(4) THE ROOT-CAUSE ANALYSIS DIAGRAM

- **Does the root-cause analysis diagram contain all the causes for wasteful product development processes?**

(5) OVERALL

- **Can the lean tools and processes described above deliver to product development organizations information that leads to continuous improvement of their value-creating processes?**

To answer these questions, the next step is determined to test the tools and processes in several product development projects.

9.2 ABOUT THE COMPANIES AND PROJECTS

9.2.1 OVERVIEW OF THE THREE PROJECTS

Table 9.1 briefly introduces the three projects investigated in this research. The product being developed in Project B is the replacement of the one developed in Project A.

Table 9.1 Description of the Three Investigated Projects

	Project A	Project B	Project C
Company	Company X (Hdqrs: USA)		Company Y (Hdqrs: Japan)
Investigated Development Site	Japan		
Focused Team	6 Engineers + Managers		5 Engineers + Managers
Team's Deliverable	Embedded Software for a High-Tech Machine		
Total Number of Involved Engineers (Partially or Fully)	100+		
Status at the Time of Investigation	Finished	Ongoing	Ongoing
Investigated Period	50 Weeks	17 Weeks	30 Weeks
Investigated Phase(s)	Detailed Design Phase	Investigation Phase (After Setting Basic Specifications)	Investigation (After Setting Basic Specifications) + Detailed Design Phase
Market	Needs: Unstable Market Size: Unstable		Needs: Stable Market Size: Unstable

9.2.2 DIFFICULTIES IN THE PROJECTS

Although the three projects are similar in that they are all embedded software development projects, they all had their own difficulties.

Project A

Design Issues

- Major architecture design change caused by their decision to integrate several components of the existing model to a single component.
- Higher complexity incurred due to this integration.
- In spite of more constraints caused by the integrated design, higher performance was required due to technological development in the market at the time of the beginning of the project.

Project Management Issues

- Offshore outsourcing.
- Resource contention: most engineers are shared by some other projects, causing multiple-tasking. In addition, the interruptions by them were not always predictable, leading to unsynchronization of processes.

Project B

Design Issue

- Maintain compatibility with the previous model.

Project Management Issue

- Scheduled to be completed in half the time spent on Project A.

Project C

Design Issue

- Realize high compatibility with the other manufacturer's machine.
- Basically no communication channel with users – their needs are communicated via the customer.

9.3 INVESTIGATION SCHEDULE

9.3.1 OVERVIEW OF INVESTIGATION SCHEDULE

Companies X and Y were visited three times and twice respectively. Phone calls and emails had been exchanged between and after the visits. Company X's investigation lasted for three months, and Company Y's, two months. Detailed schedule for the investigation of Company X is described in the following sections. The investigation of Company y followed similar processes, although I could make it more efficient by applying what I had learned through the investigation of Company X.

9.3.2 DETAILED SCHEDULE FOR COMPANY X

Preparation

A telephone conference was held with two project managers and one engineer; all of them were active members of projects A and B. Basic information about both projects (project periods, the team's roles, number of involved engineers) were informed, and the scope of the investigation was discussed. Both parties agreed to start the investigation at the end of the conference.

The First Visit

Activities

The first visit to company A lasted five days. On the first day, basic information about the two projects (including organizational structure and its changes, detailed processes, the products, and the market) was explained by the project managers and an engineer. The second and third days were spent on drawing the first version of Project A's value stream map. Last two days were spared for interviews with one of the project managers and all the available engineers engaged in Project(s) A and/or B. Each interview took 30 to 60 minutes. Common interview questions included the following:

- "What do you think was the difference between the two projects?" "Why do you think so?"
- "What kind of difficulties had you encountered through Project A?

These questions worked as effective catalysts.

Although most meetings were held in the company's conference rooms, I was allowed to occupy a desk located close to the development team. This helped me to develop a better understanding about how they work, how information is exchanged, and even each engineer's personality.

Obtained Data Other than from Interviews

Project A:

- MS-Project files with the scheduled and actual processes.
- Each engineer's weekly reports sent to the project managers. The report includes information about time spent on each task, updated information about the problems and difficulties the engineers encountered, and their updated plans for the month.

Project B:

- The MS-Project file that included both the scheduled and actual processes.

What I learned

Value stream maps for the finished portions of projects should be completed if possible, for drawing them takes long time. In order to do to do this, both actual and planned schedules should be obtained well before a visit. For this reason, before I visit Company Y, I obtained as much information as possible, leading to more effective and efficient information exchanges then.

Between the First and the Second Visits

Weekly Updates from the Project Manager

Project B's weekly reports were sent from the project manager to me through the internet every week. The reports were basically intended to report to the other project managers, and they contained information obtained through peer-to-peer interviews with each engineer, the actual time spent on Project B. Updated MS-Project files were attached to the reports. Telephone conferences with the project manager were held for thirty minutes on average almost every week; most of the time was spent on asking questions about the recent weekly reports.

Other Information Exchanges

Additional information was obtained by exchanging emails and phone calls with engineers.

Drawing Value Stream Maps

Project A's value stream map was made based on the schedule information in the MS-Project file. Project B's value stream map was updated after each telephone conference.

The Second Visit

Intensive interviews with the project manager and all the available engineers were performed through this two-day visit. Tentative versions of the value stream maps were used for asking questions. Project A's value stream map at that time had information about schedule slips, and interviews were focused on repeatedly asking the reasons for the slips until root causes were identified as is suggested by Ohno (1978). Project B's value stream map, having more detailed information than Project A's by then, was used for asking reasons for identified wastes and information flows that had not been identified in the value stream map.

Between the Second and the third Visits

Almost same activities and information exchanges were performed as the first interval. Several reports on Project A's design/code reviews were obtained. Project A's value stream map was improved reflecting the recent interview results and the information in each engineer's weekly reports.

The Third Visit

With more detailed value stream maps, intensive interviews with engineers were performed. A presentation showing tentative results was held in front of two project managers and several engineers, followed by extensive discussions.

After The Third Visit

Value stream maps were updated by reflecting the recent interview results. Root-cause analyses using the root-cause analysis diagram were performed based on all the information obtained by then and additional emails and phone calls. Wasted time on each identified waste was measured using the methodology explained in chapter 7.

CHAPTER 10: RESULTS OF RESEARCH INVESTIGATIONS 1: ANALYSES ON NINE WASTE INDICATORS

10.1 OVERVIEW OF THIS CHAPTER

10.2 looks into the wasted time captured by the nine waste indicators. Some waste indicators were more significant than the others. 10.3 analyzes the result by looking into the causes for the wasted time with the root-cause analysis diagram. 10.4 summarizes this chapter.

10.2. OVERALL RESULTS

10.2.1 NUMBER OF OCCURENCES

Figure 10.1 shows the occurrences of waste indicators per 50 engineering weeks in the three projects. Motion was the most frequent in all the projects. Especially, its occurrences were outstanding in Project C. The occurrences of over production, waiting, and rework were fewer than the others. Project B had fewer occurrences of rework and defective information. One of the reasons for this is that most of the tracked period of Project B was on the investigation phase, on which some need for rework may be undiscovered, and the rework not yet done.

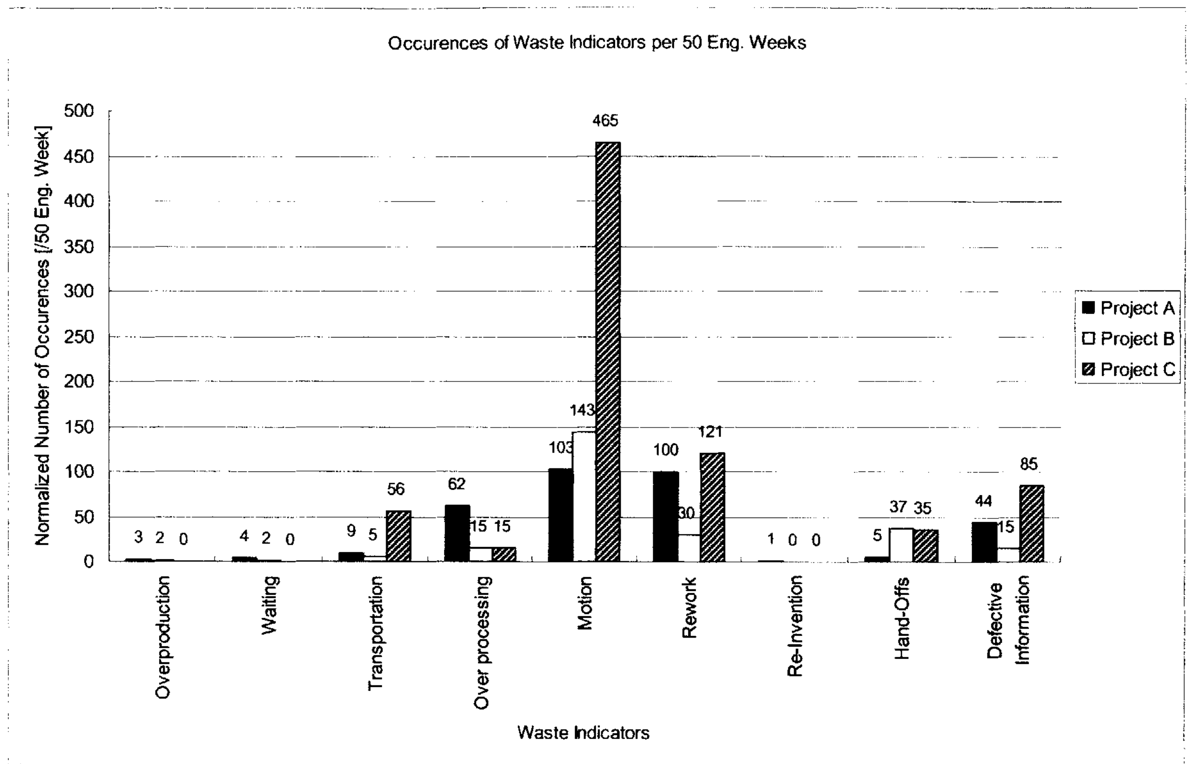


Figure 10.1 Normalized Occurrences of Waste Indicators per 50 Engineering Weeks in Projects A, B, and C

10.2.2 AVERAGE WASTED TIME PER ONE OCCURENCE

Figure 10.2 shows the average wasted time on each waste indicator in the three projects. The overall average wasted time per one occurrence of waste indicator in Projects A, B, and C were 17, 3, and 8 engineering hours respectively. Overproduction took 23 hours in Project A on average. Waiting, along with motion and hand-off, had less average wasted time than the others. Over processing, rework, and defective information took 17 hours or more on average, except in Project B, which was on its investigation phase during my survey period.

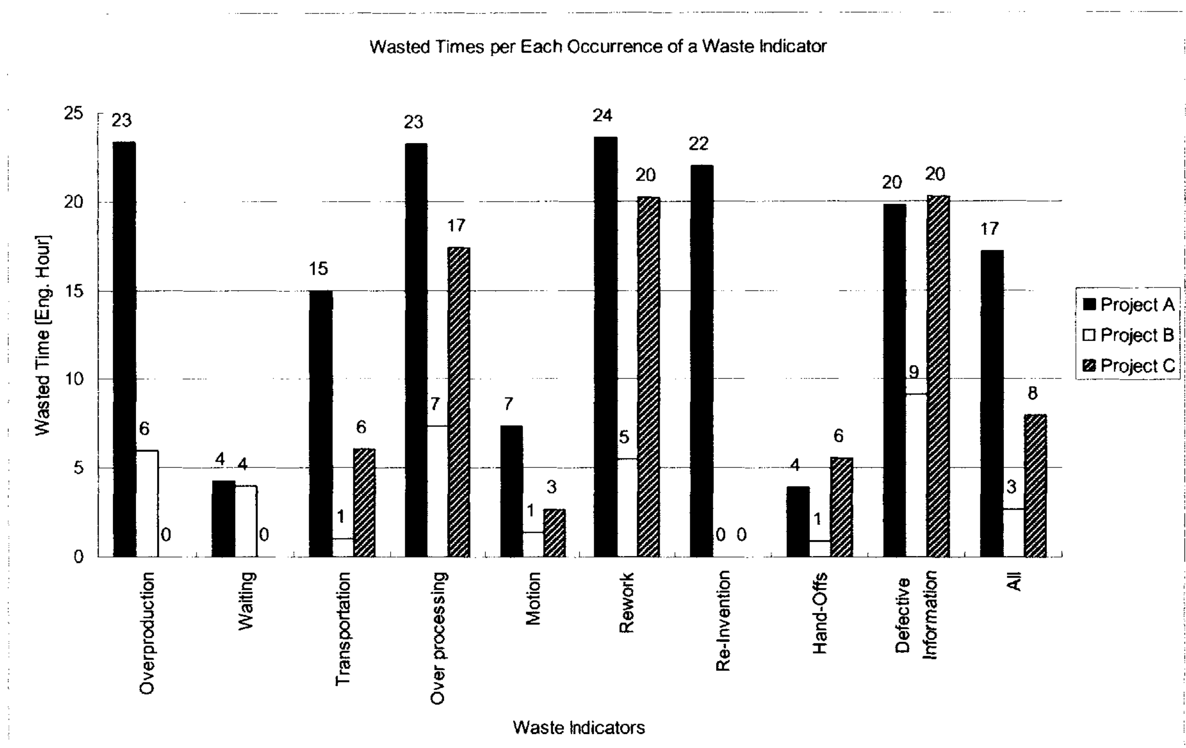


Figure 10.2 Wasted Time per Each Occurrence of a Waste Indicator in Projects, A, B, and C

10.2.3 TOTAL WASTED TIME

Figure 10.3 shows the total wasted time on each waste indicator in the three projects. Over processing, motion, rework, and defective information were the top four waste indicators in almost all the three projects. This implies that the four waste indicators are more important than the others. Although one occurrence of overproduction wasted 23 hours on average in Project A, the total wasted time on it was trivial compared to the top four waste indicators. Waiting was also trivial, implying engineers always have some tasks in their cues.

Project A's wasted time on over production was outstanding among the three projects, indicating that 1,438 engineering hours were wasted for some reasons, including changes and errors. Project A also wasted time on rework more than the other two projects. Project B's wasted time was much less than the others in transportation, motion, rework, and defective information. This result will be analyzed in detail in 10.3.

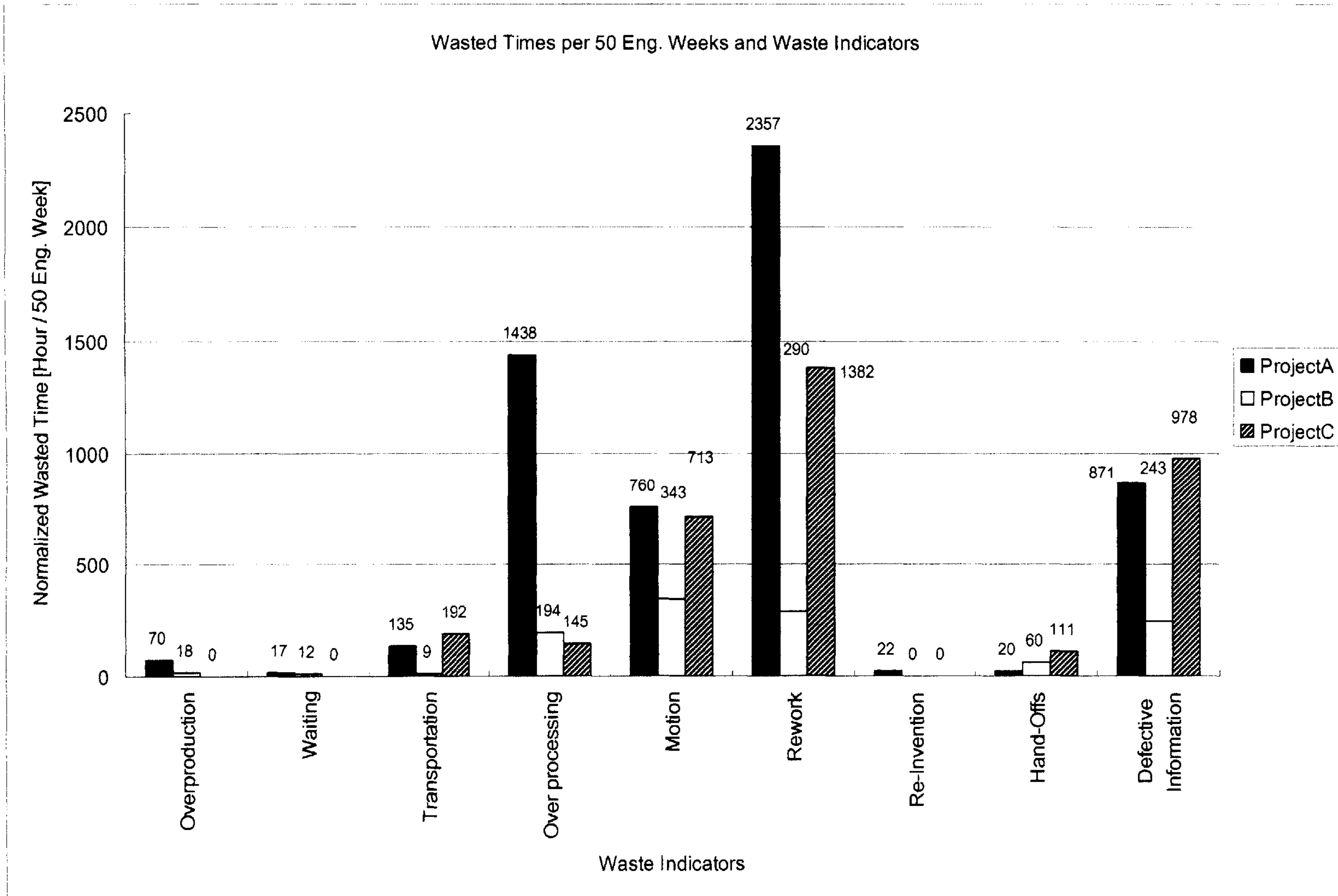


Figure 10.3 Normalized Total Wasted Time per 50 Engineering Weeks and Waste Indicators

10.2.4 TEMPORAL CHANGES IN WASTE INDICATOR DISTRIBUTIONS

Figures 10.4-10.6 show temporal changes in waste indicator distributions in Projects A-C respectively. In Project A, the total wasted time fluctuated over time. This fluctuation implies the software team's activities, which are the downstream tasks of the hardware team's activities, had largely been affected by intermittent hardware releases, for Project A involved major changes in both hardware and software. In contrast, Project C, which involved no major hardware change, had much less fluctuation than Project A. Wasted time on rework increased as time spent on the project increases in all three projects.

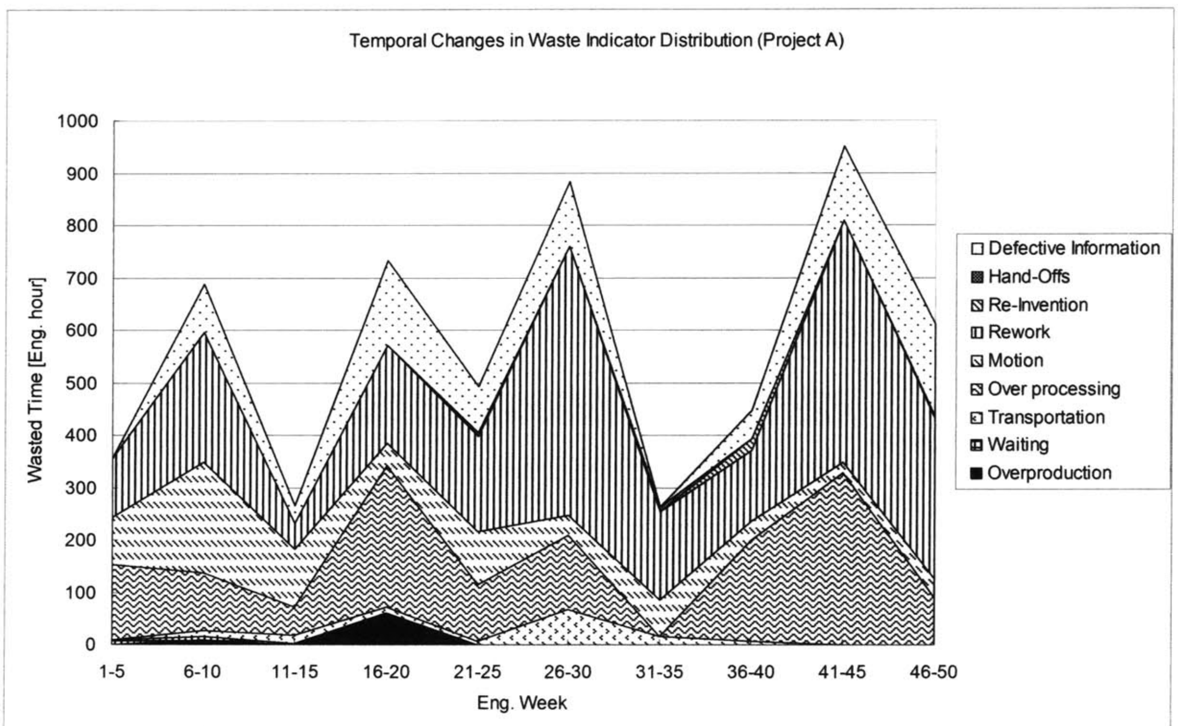


Figure 10.4 Temporal Changes in Waste Indicator Distribution in Project A

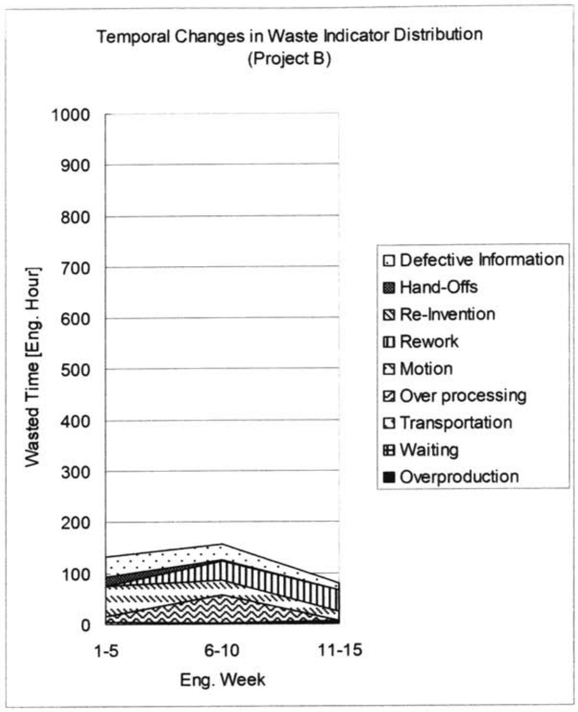


Figure 10.5 Temporal Changes in Waste Indicator Distribution in Project B

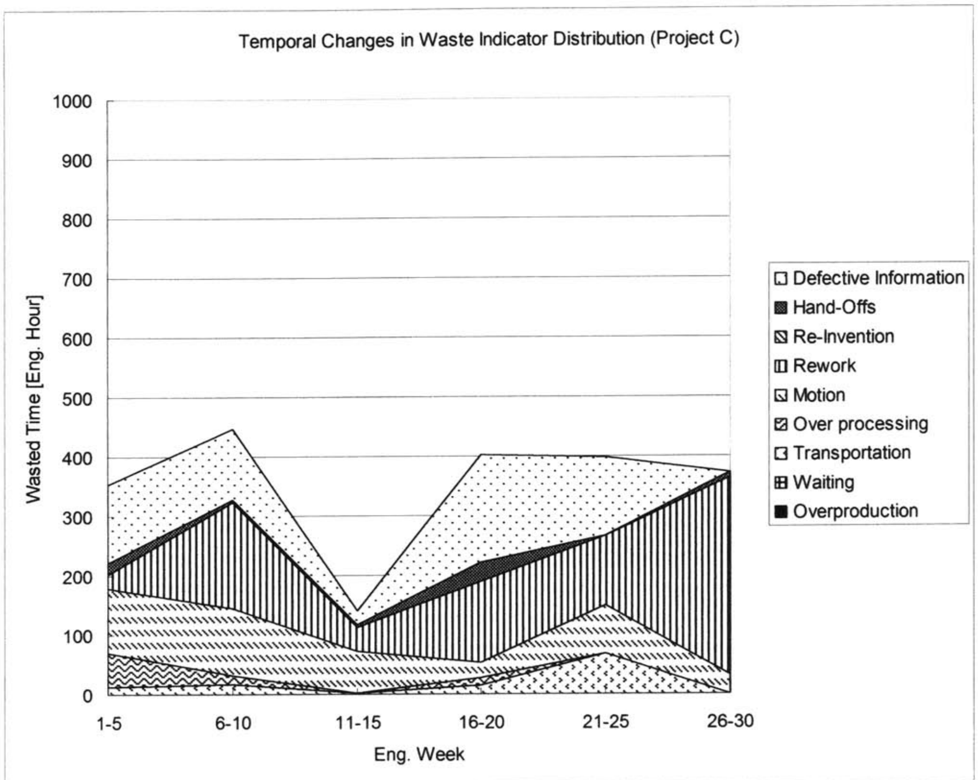


Figure 10.6 Temporal Changes in Waste Indicator Distribution in Project C

10.2.5 DISTRIBUTION OF WASTE INDICATORS AMONG ENGINEERS

Figures 10.7, 10.8, and 10.9 show the distributions of waste indicators of each engineer in the three projects. As can be understood from these figures, the distributions differ significantly among engineers. Tables 10.1, 10.2, and 10.3 briefly introduce the engineers' profiles. These results illustrate the following:

Waste indicator distribution is contingent on each engineer's qualification level, each engineer's role, and how information flows.

An example related to engineer's qualification is engineer T's ratio of rework to defective information. This ratio is significantly bigger than that of most of the others. This implies that it is not often for T that he makes errors by himself. This tendency is consistent with his high experience level and his character as a perfectionist (see table 10.1).

Looking into the distributions of U (in Project A) and NF (in Project C) reveals that the waste indicator distribution is also affected by each engineer's role. U in Project A and Engineer NF in Project C had similar waste indicator distributions: they are the only engineers who had wasted his/her time in the following order.

1. Motion, 2. Rework, 3. Defective information

As can be understood in tables 10.1 and 10.3, they share the same role: they are both responsible for engineering issues of other engineers while working on their own design tasks. In Project B, U's waste distribution changes significantly: he/she wasted his/her time on hand-off most. This is mainly because his/her role in Project B was project manager who provide with the engineers tasks and necessary information including specifications.

Engineer H's waste indicator distribution in Project A is a distinct proof that the distribution is affected by how information flows. H wasted his time on over processing most. This was due to his working on tentative information, which was caused by late information releases from the hardware team.

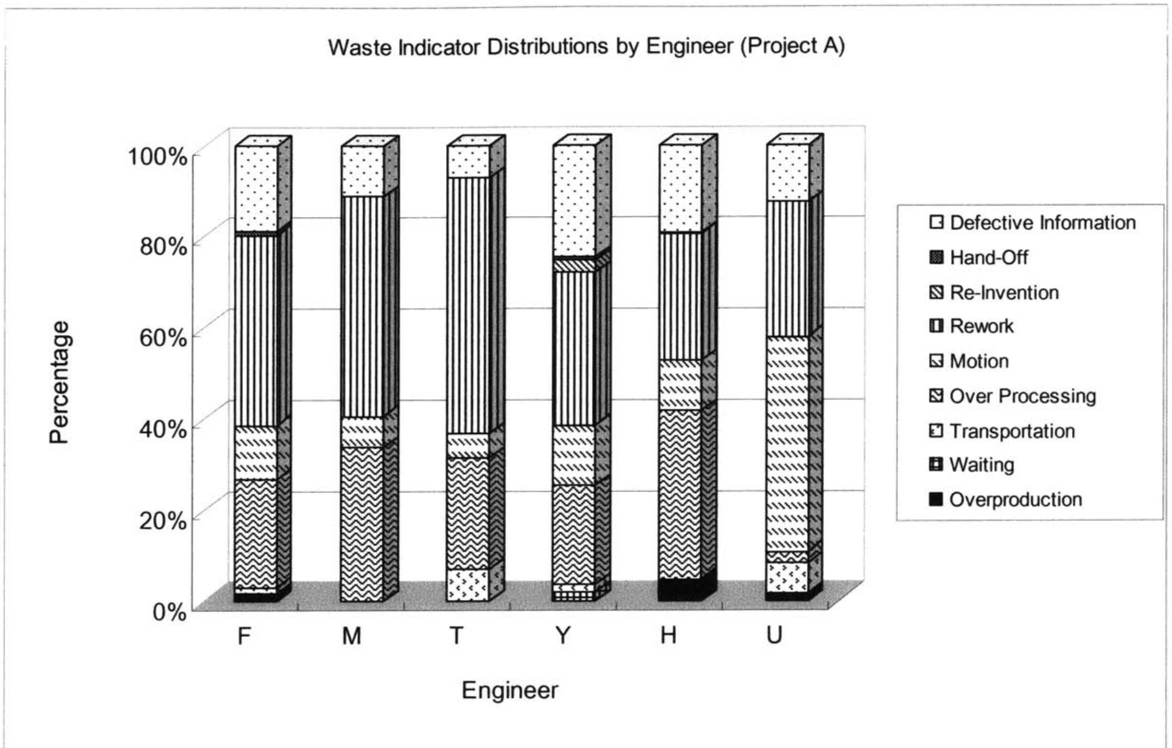


Figure 10.7 Waste Indicator Distributions by Engineer (Project A)

Table 10.1 Each Engineer's Profile

Engineer	
F	Young engineer. First experience in a major software development project.
M	Experienced engineer.
T	Experienced engineer. Seeks for perfection in his tasks.
Y	Young engineer with high motivation.
H	Experienced engineer.
U	Not the project manager, but leads the team in technically like a chief engineer.

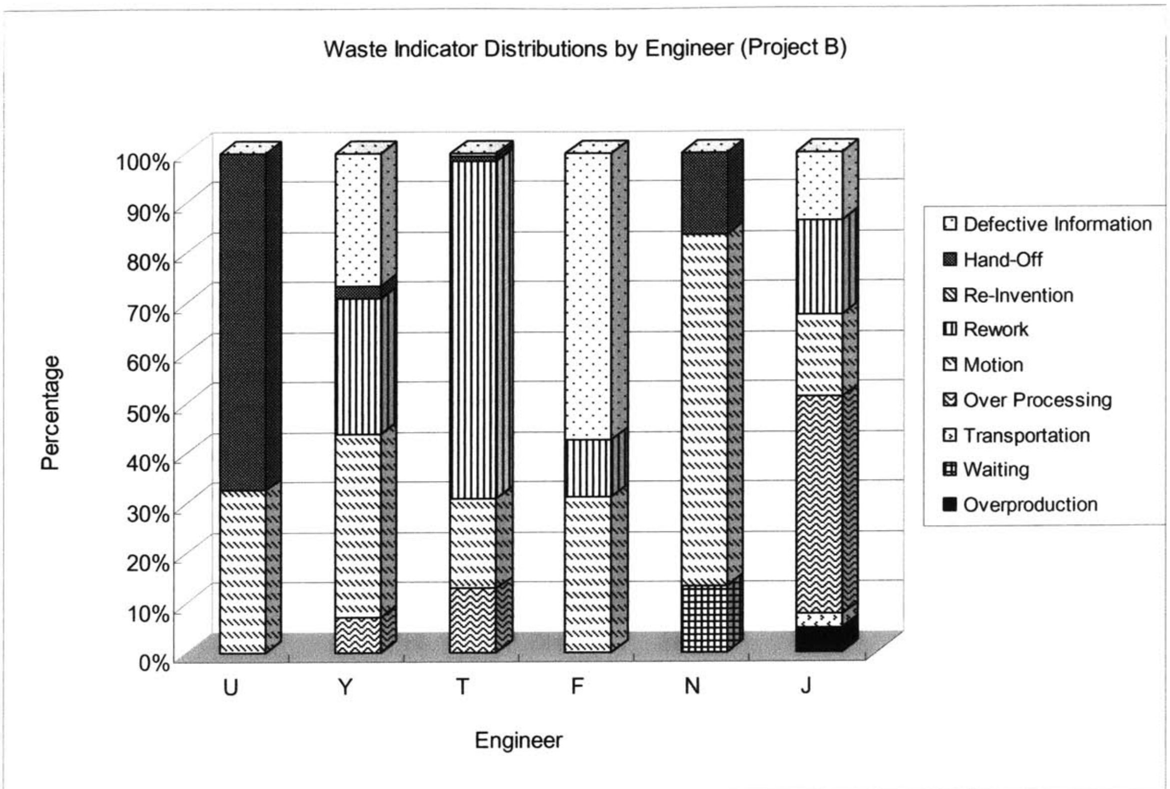


Figure 10.8 Waste Indicator Distributions by Engineer (Project B)

Table 10.2 Each Engineer's Profile (Project B)

Engineer	
U	Working project manager: not only manage the team, but acts like the team's buffer. (see also table 10.1)
Y	(see table 10.1)
T	(see table 10.1)
F	(see table 10.1)
N	Experienced engineer.
J	Temporary engineer. Limited experience in software engineering.

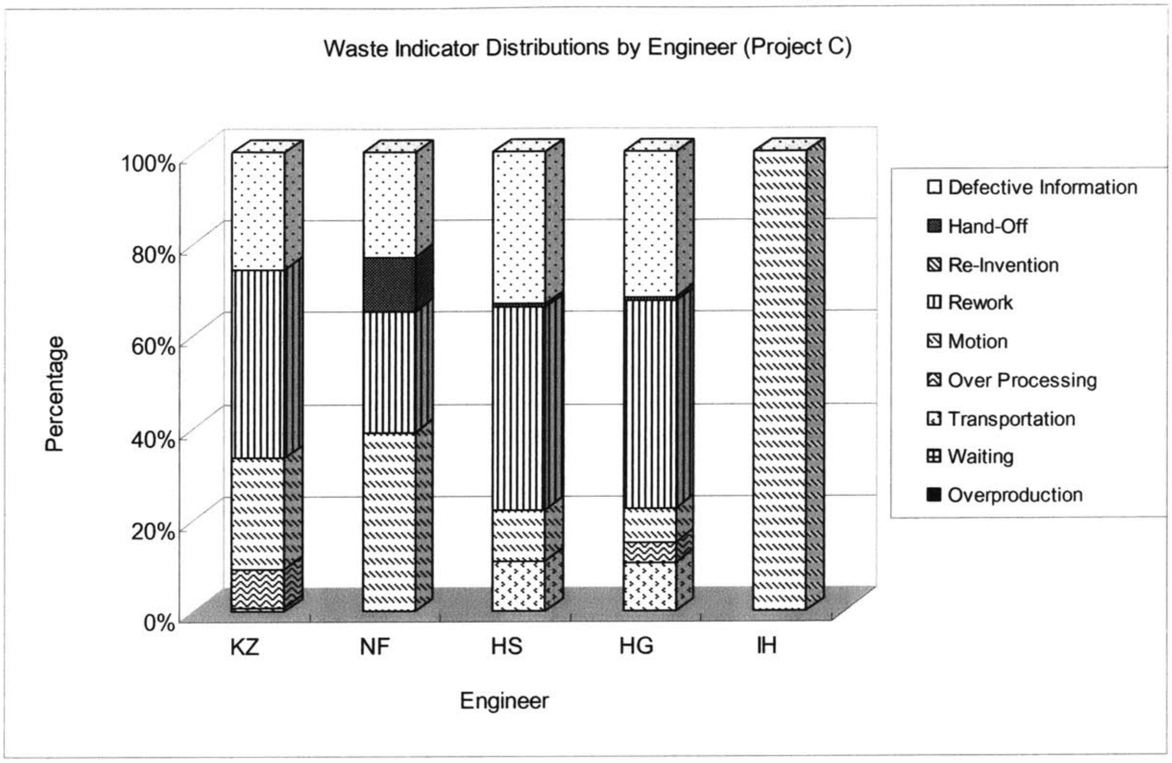


Figure 10.9 Waste Indicator Distributions by Engineer (Project C)

Table 10.3 Each Engineer's Profile (Project C)

Engineer	
KZ	Young engineer with limited experience. Unfamiliar with the company's coding rules/ design philosophy.
NF	Not the project manager, but leads the team in technically like a chief engineer.
HS	Experienced engineer. No experience in the function assigned in Project C.
HG	Experienced engineer. No experience in the function assigned in Project C.
IH	Young engineer.

10.3 DETAILED ANALYSIS ON EACH WASTE INDICATOR

10.3.1 OVERVIEW

The five waste indicators that were not more significant than the others (see figure 10.3), overproduction, waiting, transportation, re-invention, and hand-off, are briefly reviewed in this section. The top four waste indicators, over processing, motion, rework, and defective information are analyzed in detail using the root-cause analysis diagram.

10.3.2 OVERPRODUCTION – WASTED ENGINEERING HOURS BY CAUSES

Figure 10.10 shows the relationship between wasted time on overproduction and the corresponding causes. Overproduction was identified in Projects A and B. Unclear division of labor was the significant cause in Project A. On the other hand, the only cause identified in Project B was under qualification, meaning an engineer's qualification was not enough for the assigned task.

Making use of the architecture of previous model

The engineer added source codes without fully understanding the legacy source codes, causing redundancy source codes.

Premature architecture design

This made the architecture too complex, causing redundancy in design.

Under qualification

Inexperienced engineers tend to make redundant source codes.

Unclear division of labor

Two engineers took care of the same task unintentionally.

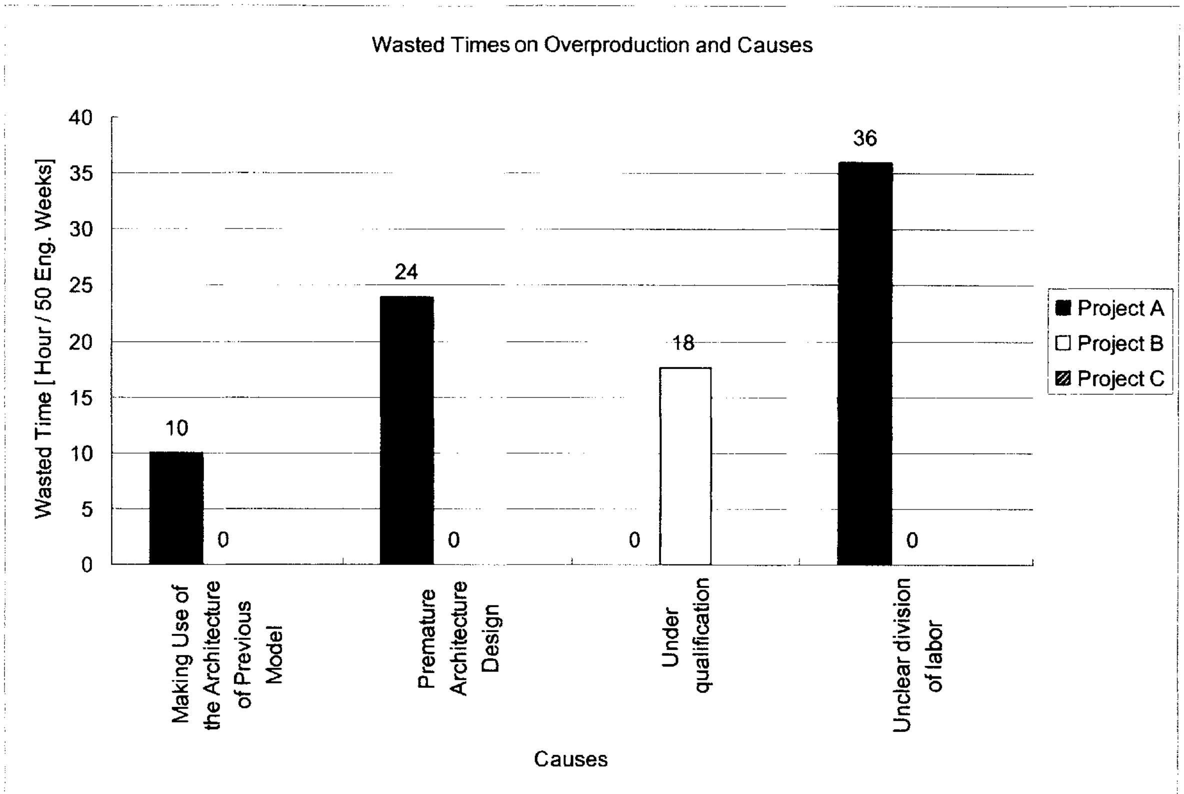


Figure 10.10 Normalized Waste Time on Over Production per 50 Engineering Weeks and the Corresponding Causes

10.3.3 WAITING – WASTED ENGINEERING HOURS BY CAUSES

Figure 10.11 shows the relationship between wasted time on waiting and the corresponding causes. Waiting was identified Projects A and B. Insufficient maintenance of development environment and limited tools/ prototypes/ hardware were the causes respectively. This result implies that engineers are forced to wait only in the unexpected situations in which they encounter some hardware problems; in other situations, they switched their tasks. For example, when some information was necessary to process a task, the engineers started working on another one instead of waiting for the information.

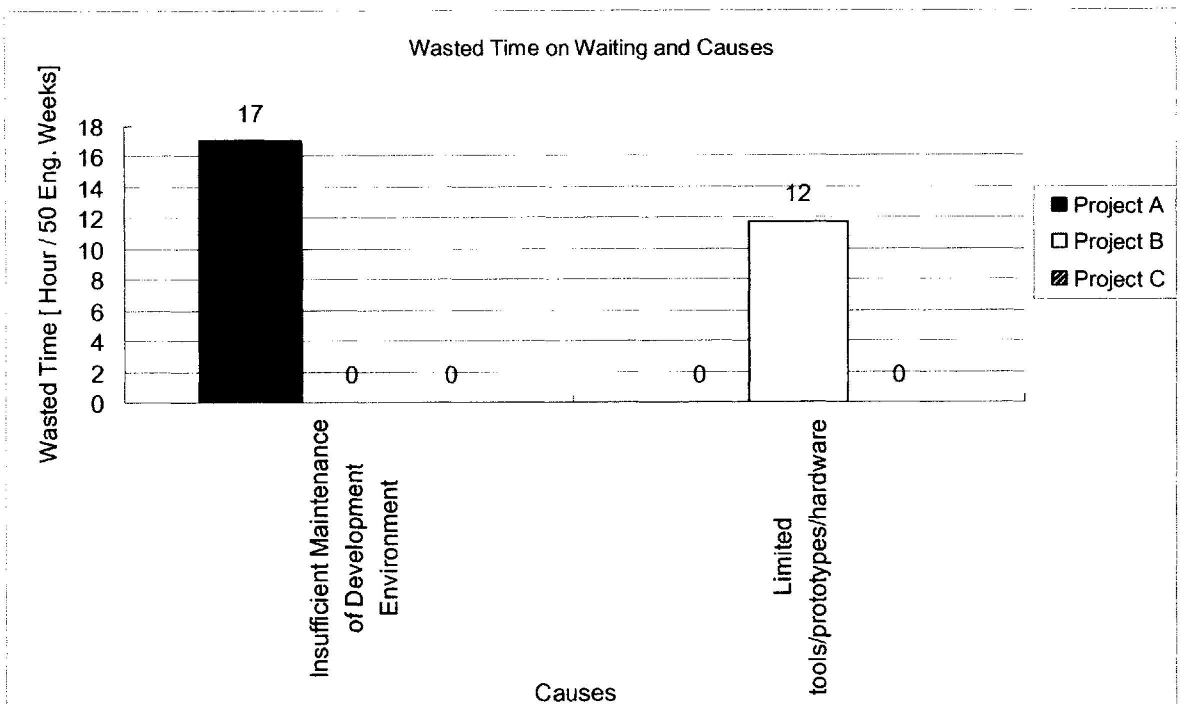


Figure 10.11 Normalized Waste Time on Waiting per 50 Engineering Weeks and The Corresponding Causes

10.3.4 TRANSPORTATION – WASTED ENGINEERING HOURS BY CAUSES

Figure 10.12 shows the relationship between wasted time on transportation and the corresponding causes. Although transportation was identified in all the three projects, their distributions of causes were different from each other: only spatial/structural barrier was shared by multiple projects. In Project A, Changes in design methodology and changes in documenting / database format/ guidelines were the two significant causes. Both causes re-formatting information (figure 10.13).

Changes in design methodology

Design methodology change was decided in favor of higher performance, causing reformatting source codes.

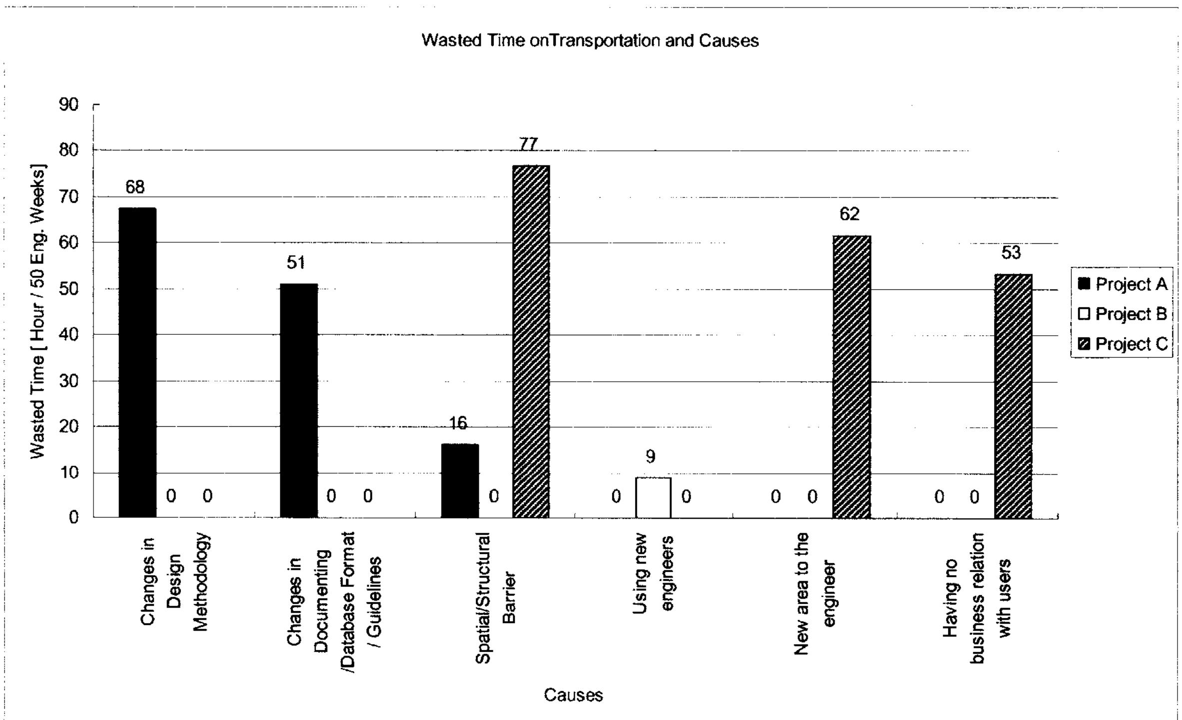


Figure 10.12 Normalized Waste Time on Transportation per 50 Engineering Weeks and the Corresponding Causes

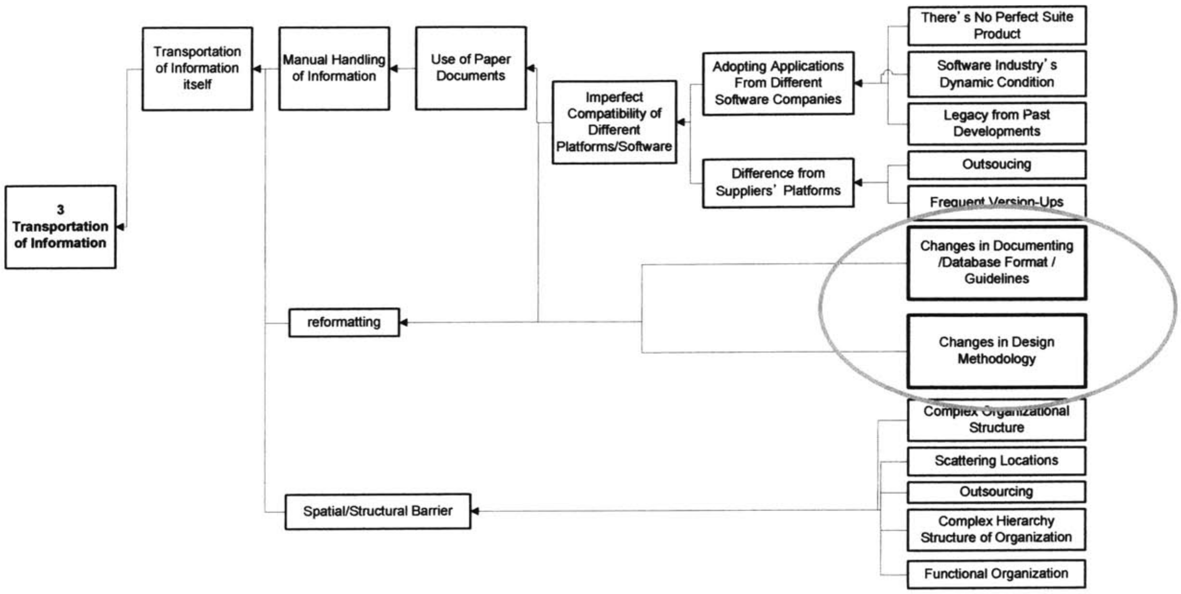


Figure 10.13 Part of the Root-Cause Analysis Diagram of Transportation of Information

10.3.5 OVER PROCESSING

(1)Wasted Engineering Hours by Causes

Figure 10.14 shows the relationship between wasted time on over processing and the corresponding causes. Over processing was more significant in Project A than in the other two. (i) "Undiscovered errors in outputs from upstream", (ii) "Upstream changes/ poor concept design/ marketing information" were the most significant causes in Project A. (iii) "Prototype version confusion" was the most significant one in Project B.

(2) Examples of Root-Cause Analysis

The root cause analysis of the causes (i), (ii) and (iii) are quoted from the root cause analysis diagram discussed in chapter 5 (see figure 10.14).

(i) Undiscovered errors in outputs from upstream.

As can be understood from figure 10.14, the typical root causes for this cause are the following:

- Defective information
- Upstream task's dependency on downstream tasks for verification
- Existence of risks / uncertainties
- Limited resources (3) – Limited capacity of organization

(ii) Upstream changes/ poor concept design/ marketing information

As can be understood from figure 10.15, the typical root causes for this cause are the following:

- PD's nature (3) – Identifying all interfaces in advance is impossible
- PD's nature (2) – Iteration cannot be eliminated
- Poor marketing information
- Poor concept design
- Poor architecture design

(iii) Prototype version confusion

As can be understood from figure 10.14, the typical root causes for this cause are the following:

- Poor work-in-process version management
- Scattered locations
- Complex organizational structure

- Outsourcing
- Functional organization
- Complex hierarchy structure of organization

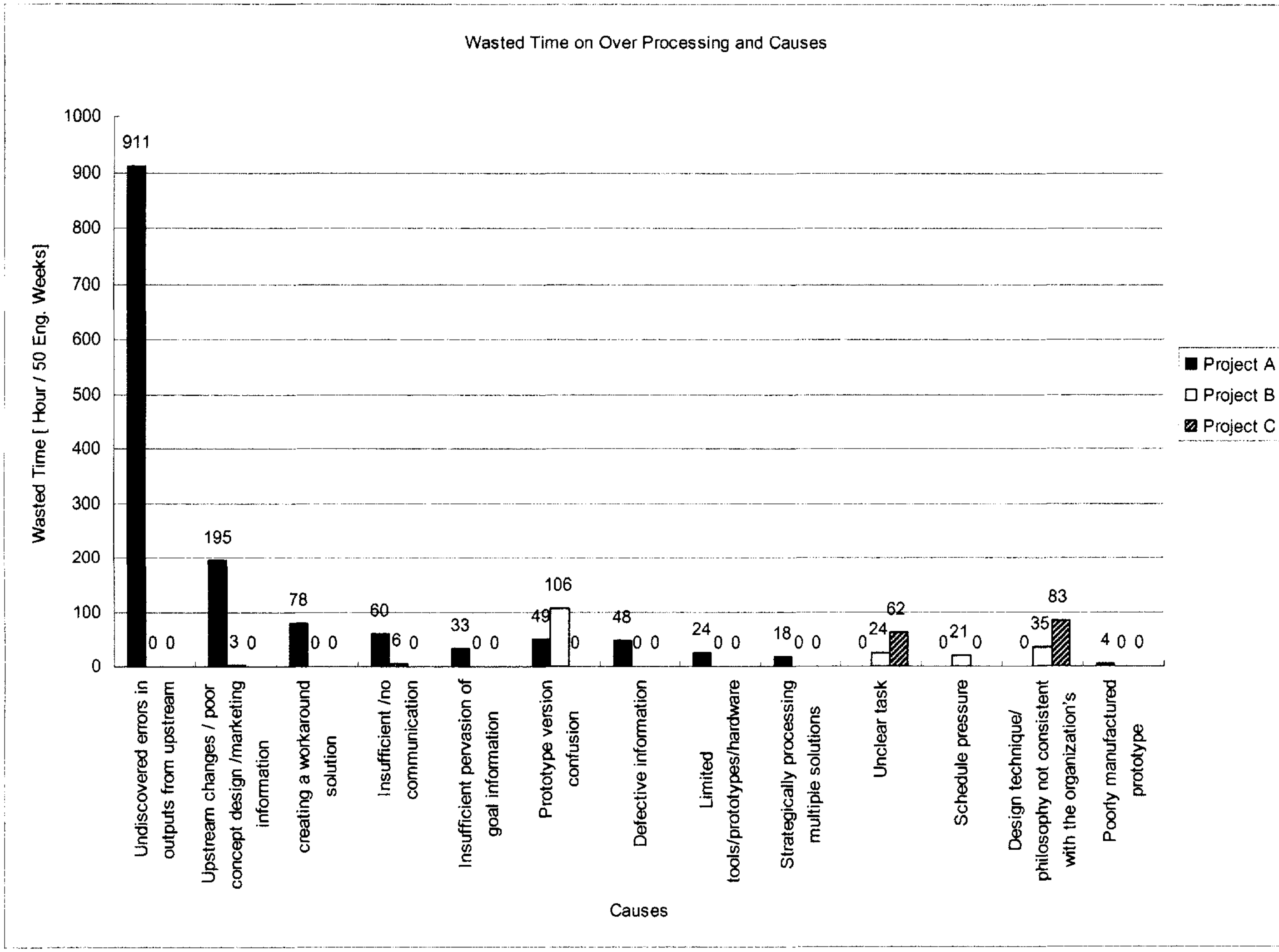


Figure 10.14 Normalized Waste Time on Over Processing per 50 Engineering Weeks and The Corresponding Causes

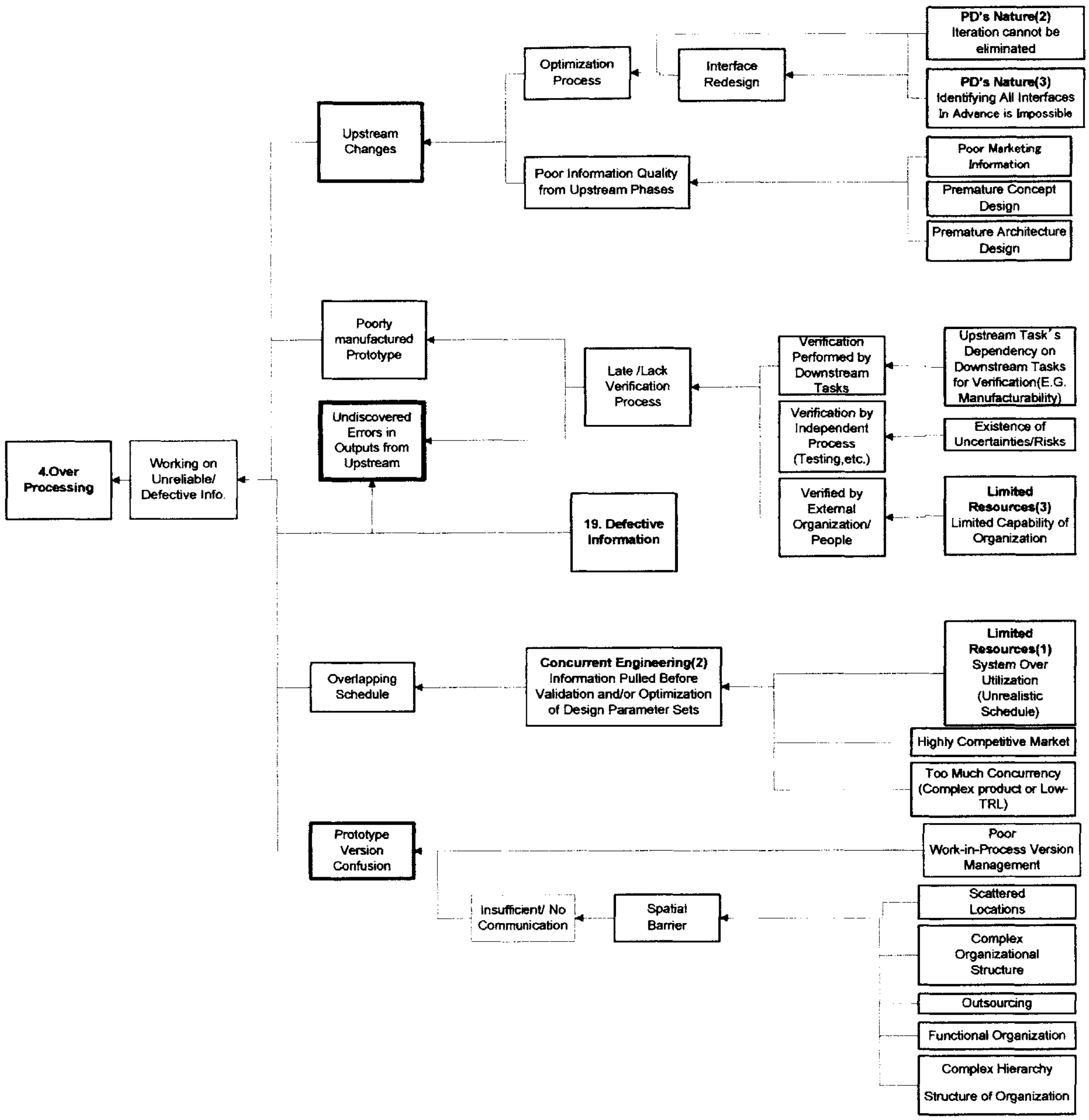


Figure 10.15 Part of the Root-Cause Analysis Diagram of Over Processing

(3) Discussion

The most significant cause of over processing in Project A, "(i) Undiscovered errors in outputs from upstream," is common in many embedded software development projects in which both hardware and software are developed concurrently. The first two root causes, "Defective information" and "Upstream task's dependency on downstream tasks for verification" reveals the wasteful relationship between the hardware and the software team, in which defective information flows down to the software team (figure 10.16). Because of this pattern, the software team needed to waste time on defective information for which they were not responsible. Creating defective information itself is wasteful. Passing the defective information to downstream is also wasteful. If there had been an effective verification processes before handing of information to the downstream team, both teams could have reduced wasted time: the software team can reduce time on over processing, and the hardware team can get feedback quickly. It is generally difficult to test prototype hardware without a complete version of embedded software, but finding ways to check errors effectively before handing off hardware prototype can reduce waste, along with efforts to improve prototypes' quality (this is more essential), can some portion of the wasted time of 911 hours / 50 months (figure 10.13).

The third root cause, "Existence of risks / uncertainties" cannot be controlled. However, wasted time on over processing can be reduced by identifying risks/ uncertainties earlier by introducing front-loaded processes including set-based concurrent engineering, or spiral processes.

The fourth root cause, "Limited capacity of organization," may be controlled. However, it is usually difficult because bringing up embedded software engineers takes longer time than general software engineers because embedded software engineers need to have expertise for specific hardware.

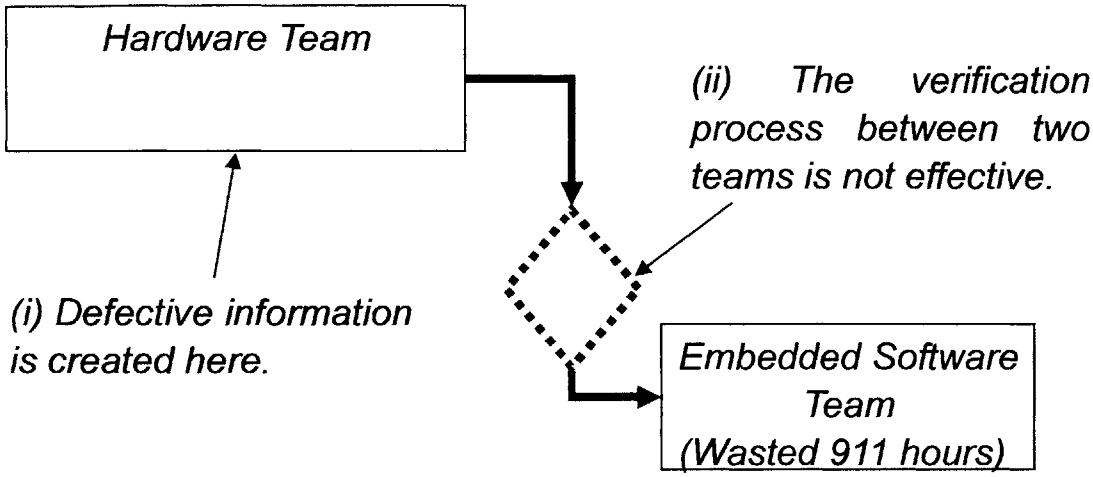


Figure 10.16 Wasteful Information Flow without Effective Verification Processes

The identified root causes for the second significant cause, "(ii) Upstream changes/ poor concept design/ marketing information," suggests that Project A's concept development phase might have been premature. Because the project was processed without mature or good information, the team happened to spend 195 hours /week on over processing. However, some portion of this wasted time is caused by deterioration of information, which is discussed in the next chapter.

10.3.6 MOTION

(1) Wasted Engineering Hours by Causes

Figure 10.17 shows the relationship between wasted time on motion and the corresponding causes. The top three causes for motion were (i) "Documenting", (ii) "Testing/QC" (iii) "Meeting." In these investigations, testing/QC included only review and testing tasks performed by other engineers. This was because self-testing activities were not clearly distinguishable from fixing errors.

Documenting and testing/QC were the most significant causes in Projects A and C, while they were not significant in Project B.

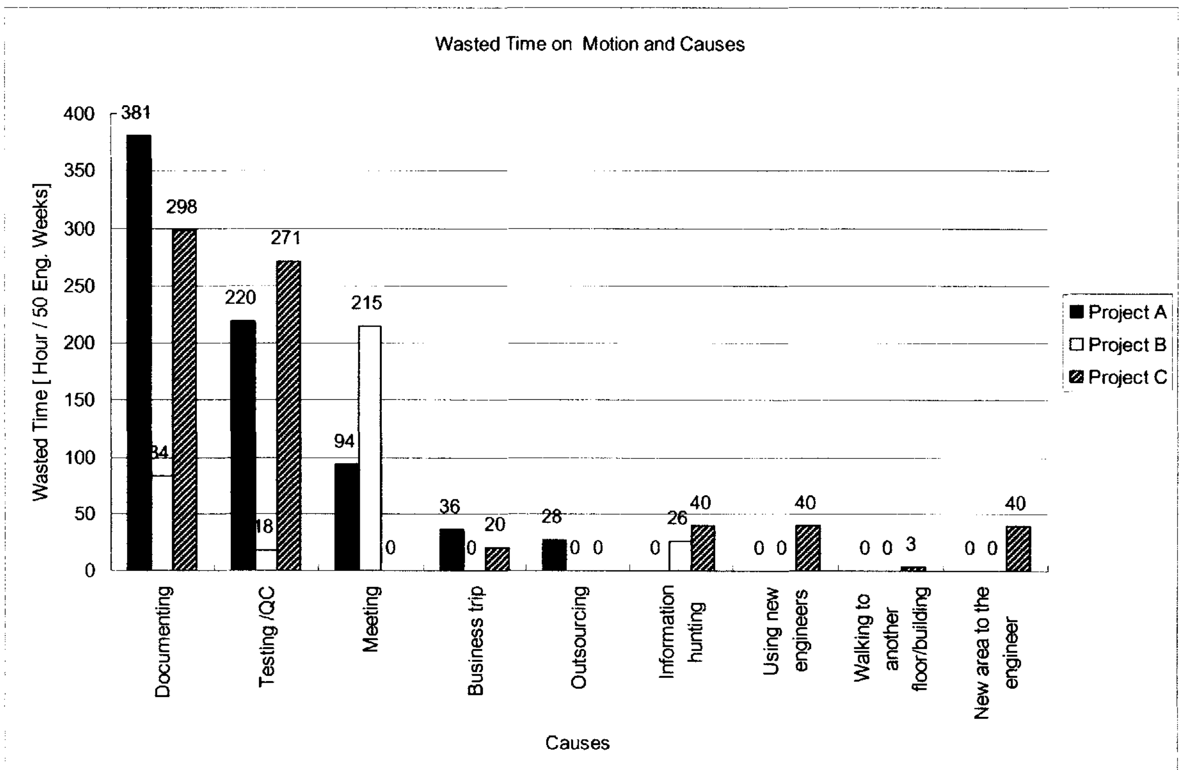


Figure 10.17 Normalized Waste Time on Motion per 50 Engineering Weeks and The Corresponding Causes

(2) Examples of Root-Cause Analysis

The root cause analysis of the causes (i), (ii) and (iii) are quoted from the root cause analysis diagram discussed in chapter 5 (see figure 10.18).

(i) Documenting

As can be understood from figure 10.18, the typical root causes for this cause are the following:

- Required activity
- Transportation

While documenting wastes the ongoing project's time, it may save time later in future projects: without easy-to-access documentation, re-invention waste may occur in the future.

(ii) Testing/ QC

As can be understood from figure 10.18, the typical root causes for this cause are the following:

-Defective information.

-Outsourcing

Testing activities are classified as waste because the better development processes are, the less time for testing is needed. This means that a development process with long testing time is not always a process that guarantees high quality products. Rather, it may be a poor process whose work-in-process information's quality is poor because defective information causes long time for testing that might have been unnecessary without defective information.

(iii) Meeting

Not all meetings are wasteful: because well-organized meetings lead to high-quality information transfer (Graebsch, 2005). On the other hand, meetings for unilateral information transfer and with unnecessary attendants are wasteful, causing other wastes by taking engineers' time.

(3) Discussion

Considering the three major causes for motion identified, measuring time spent on motion is not a good way for waste reduction. Documenting can be considered as investment for the future. Spending long time on testing is waste, but one of the two causes of it is defective information, which is one of the waste indicators. In fact, in Project C, design reviews sometimes worked as a training process. For example, many design reviews can be seen figure 10.19. The project spent long time on the reviews, which lasted 1.5h-2h each, but the reviews helped KZ learn the coding guidelines and the design philosophy of the company. Using design reviews as an opportunity for training may not always be the best way of training, but it can also be regarded as investment for the future. Deciding whether a meeting is wasteful or not requires a careful analysis.

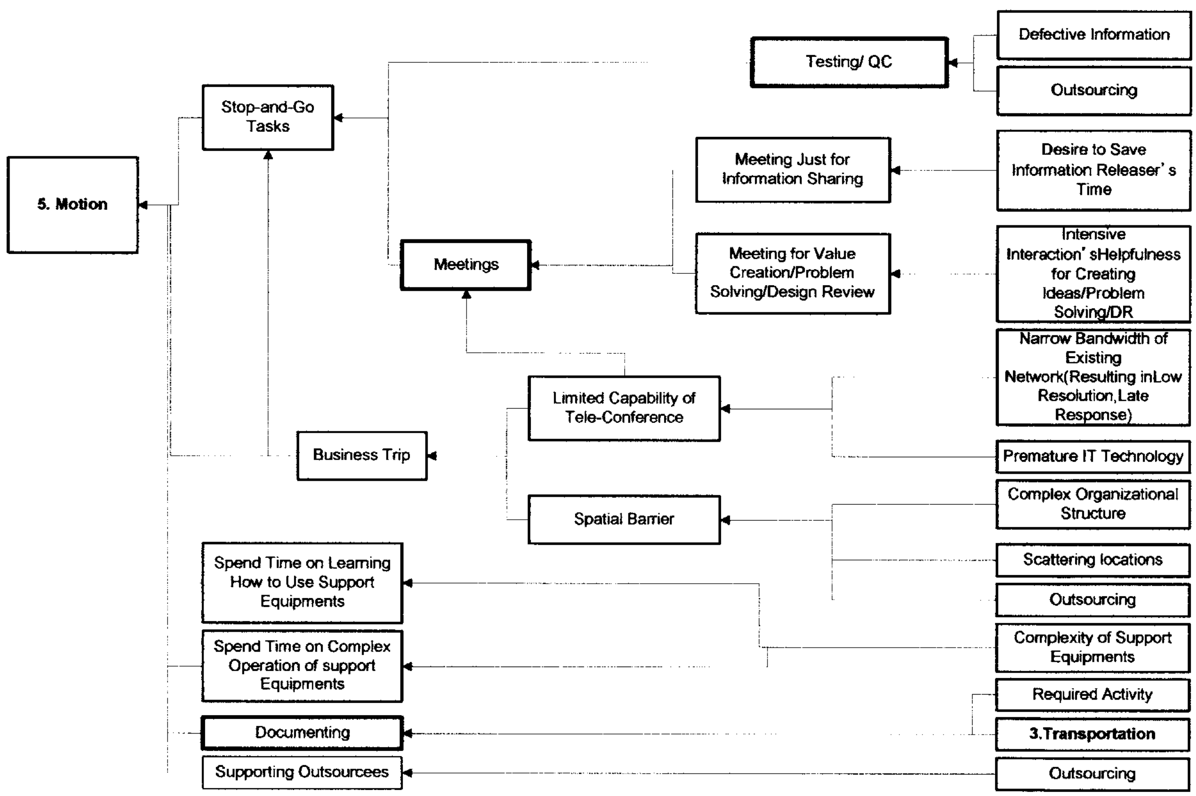


Figure 10.18 Part of the Root-Cause Analysis Diagram of Motion

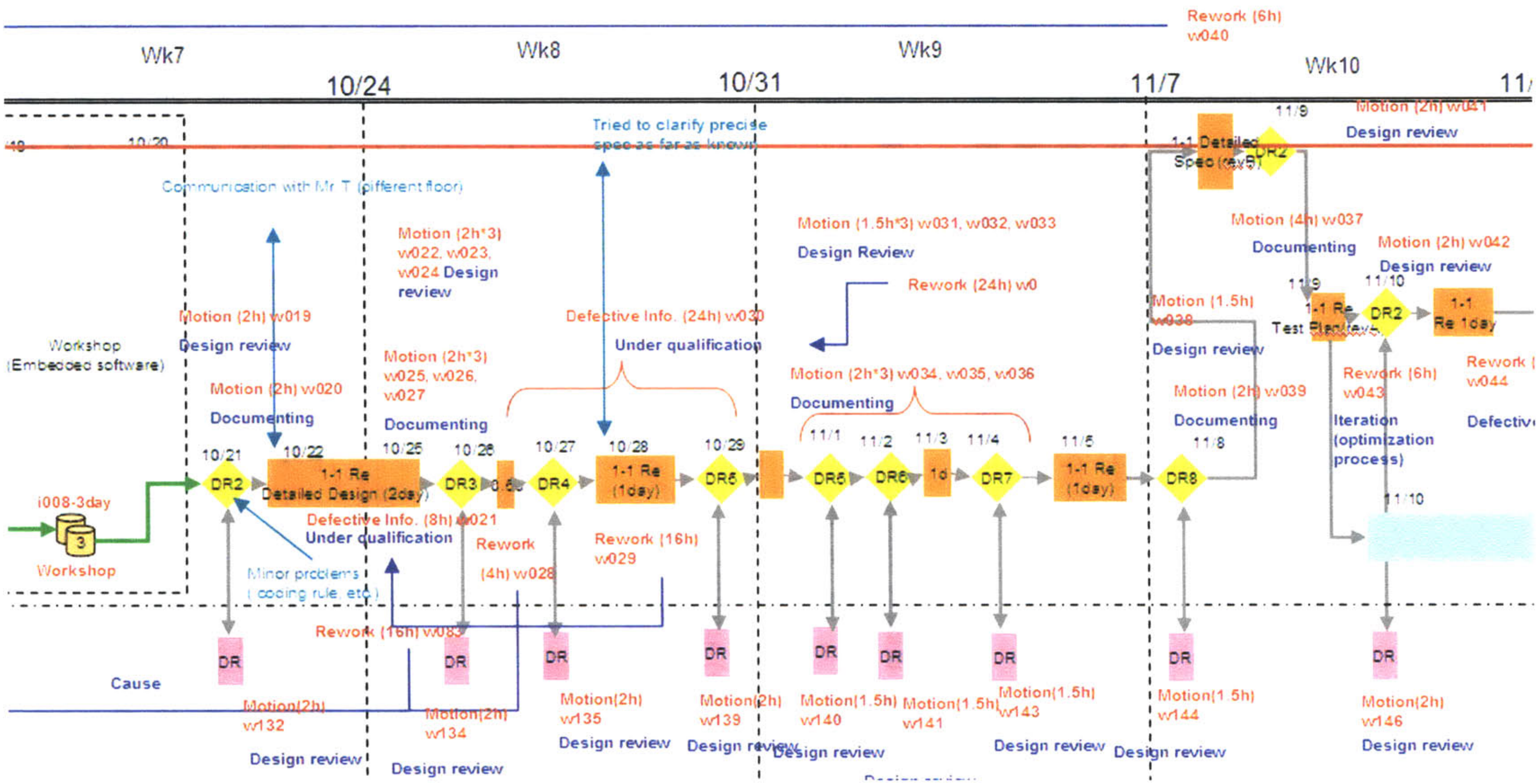


Figure 10.19 Example of a Process with Frequent Design Reviews – Engineer KZ's design Process in Project C

10.3.7 REWORK

(1) Rework in VSM

As can be seen in figure 10.20, rework is conspicuous in value stream maps. It can be easily understand how time is spent on rework by taking a glance of a value stream map; in Projects A and C, some engineers spent more than half of their time on rework.

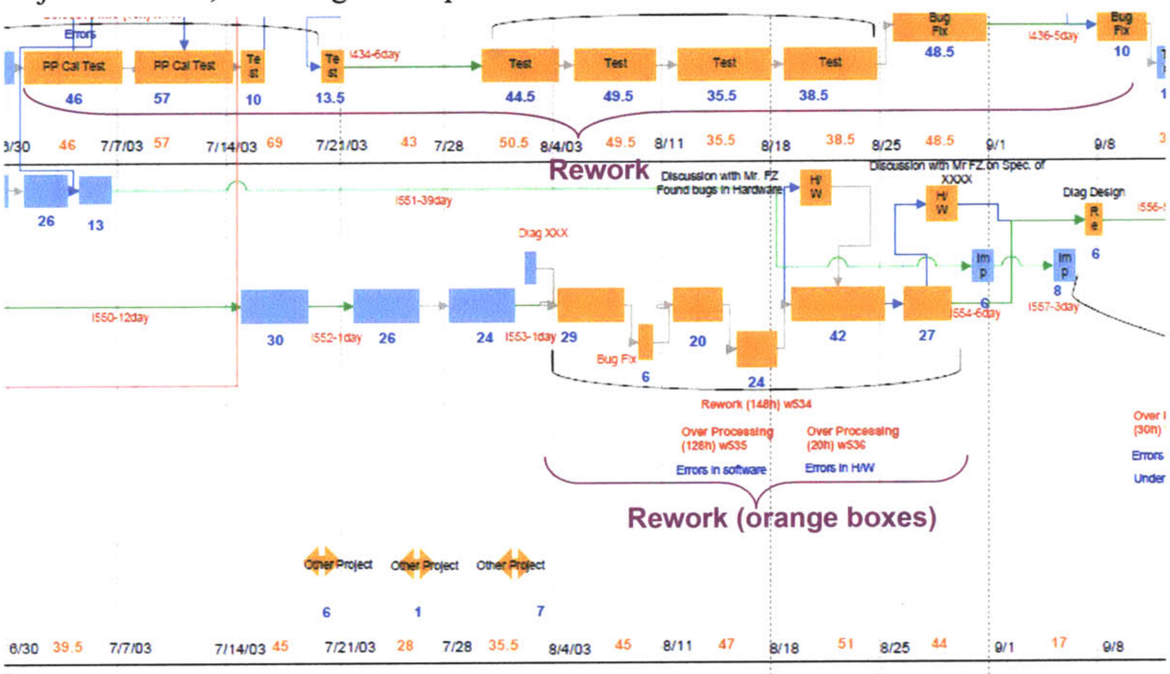


Figure 10.20 An Example of Rework in Value Stream Map (Project A, Engineer T)

(2) Wasted Engineering Hours by Causes

Figure 10.21 shows the relationship between wasted time on rework and the corresponding causes. Rework was significant waste identified in Projects A and C. (i) Troubleshooting and (ii) Defective information were the most significant reasons for wasted time on rework in Projects A and C respectively.

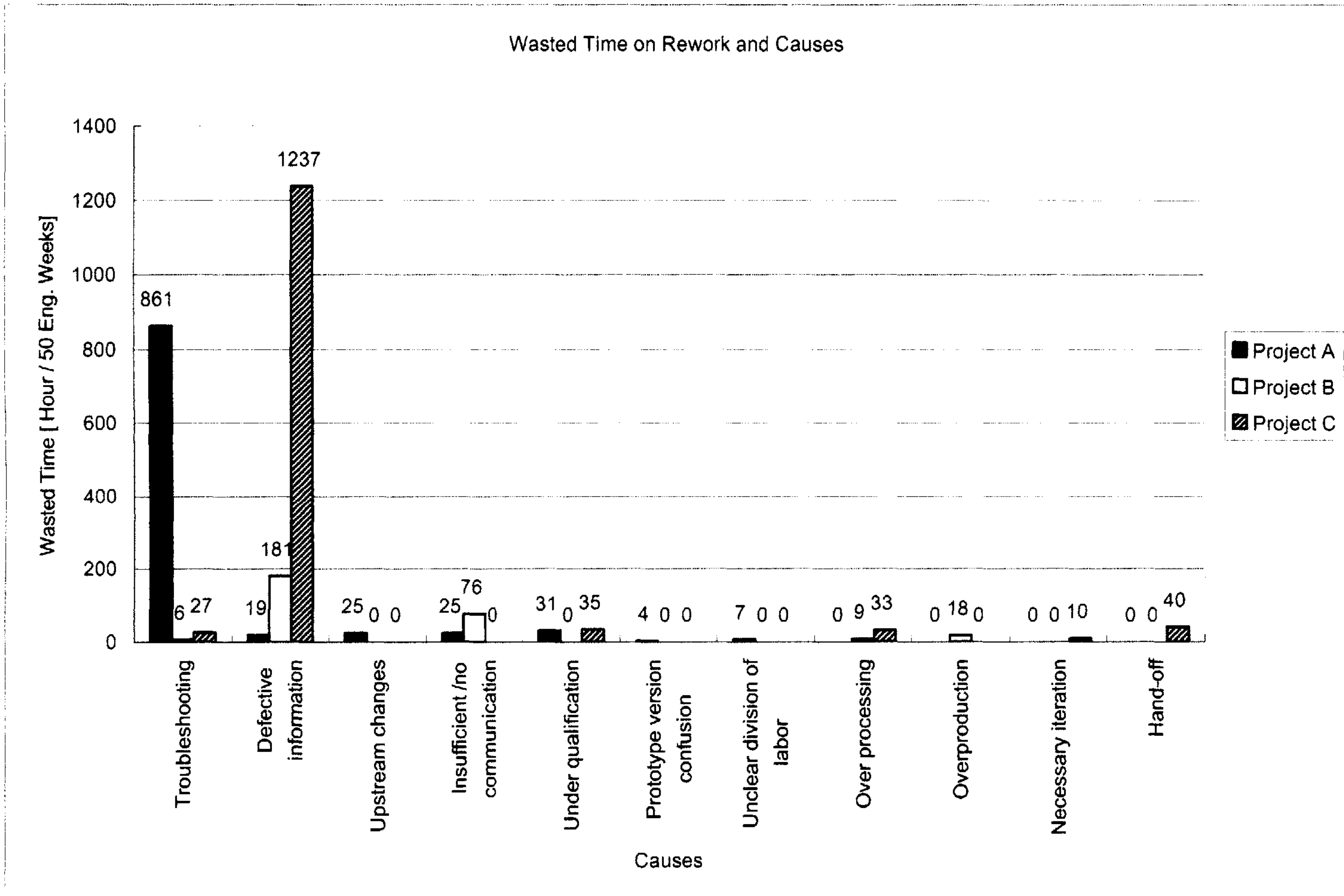


Figure 10.21 Normalized Waste Time on Rework per 50 Engineering Weeks and The Corresponding Causes

(3) Examples of Root-Cause Analysis

The root cause analysis of the causes (i) and (ii) are quoted from the root cause analysis diagram discussed in chapter 5 (see figure 10.22). Troubleshooting is caused by defective information, but, in this investigation, when a engineer had spent his/her time on troubleshooting, the wasted time was classified not as the rework time caused by defective information, but troubleshooting. On the other hand, when a engineer had spent his/time on fixing defective information, the wasted time was classified as rework time caused by defective information.

(4) Discussion

Figure 10.23 explains why 861 hours are spent on troubleshooting in Project A. As can be understood from this figure, the embedded software team needed to take care of the errors both in hardware and software. This made troubleshooting processes complicated. Furthermore, the software engineers had limited knowledge in the prototype hardware. Especially the inexperienced engineers (F and Y) spent long time on identifying where bugs are in. The quality of their work were not as high as those of experienced engineers then (they showed notable improvement in Project B). And, their troubleshooting processes were not as effective as those of the experienced engineers. Their lack of confidence, coming from inexperience, sometimes made them take long time before determining who was responsible for the bugs. It was sometimes difficult for them, partly because of the Japanese culture, to point out errors made by hardware engineers who were more experienced. Thus, engineers wasted 861 hours on troubleshooting.

Part of the huge wasted time on project A is considered to be caused by inexperience of some engineers, but some portion of it might have overcome by appropriate project management. The project managers sometimes ignored alarms by the young engineers, having put priorities on immediate issues: this made rework discovery time longer. Postponed rework makes time spent on it longer; this is discussed in (5).

Suggestions for reduction of troubleshooting time are the following:

1. Find ways to stop defective information flow across teams.
2. Let engineers work at their best conditions: too much overtime and excessive schedule pressure cause low-quality outputs.

3. Try to retrieve alarming information from engineers: hidden problems are sometimes troublesome than visible ones. (This seems to be already realized by U in Project B)

On the other hand Project C spent 1,237 hours on rework caused by defective information. Defective information is discusses in 10.3.9.

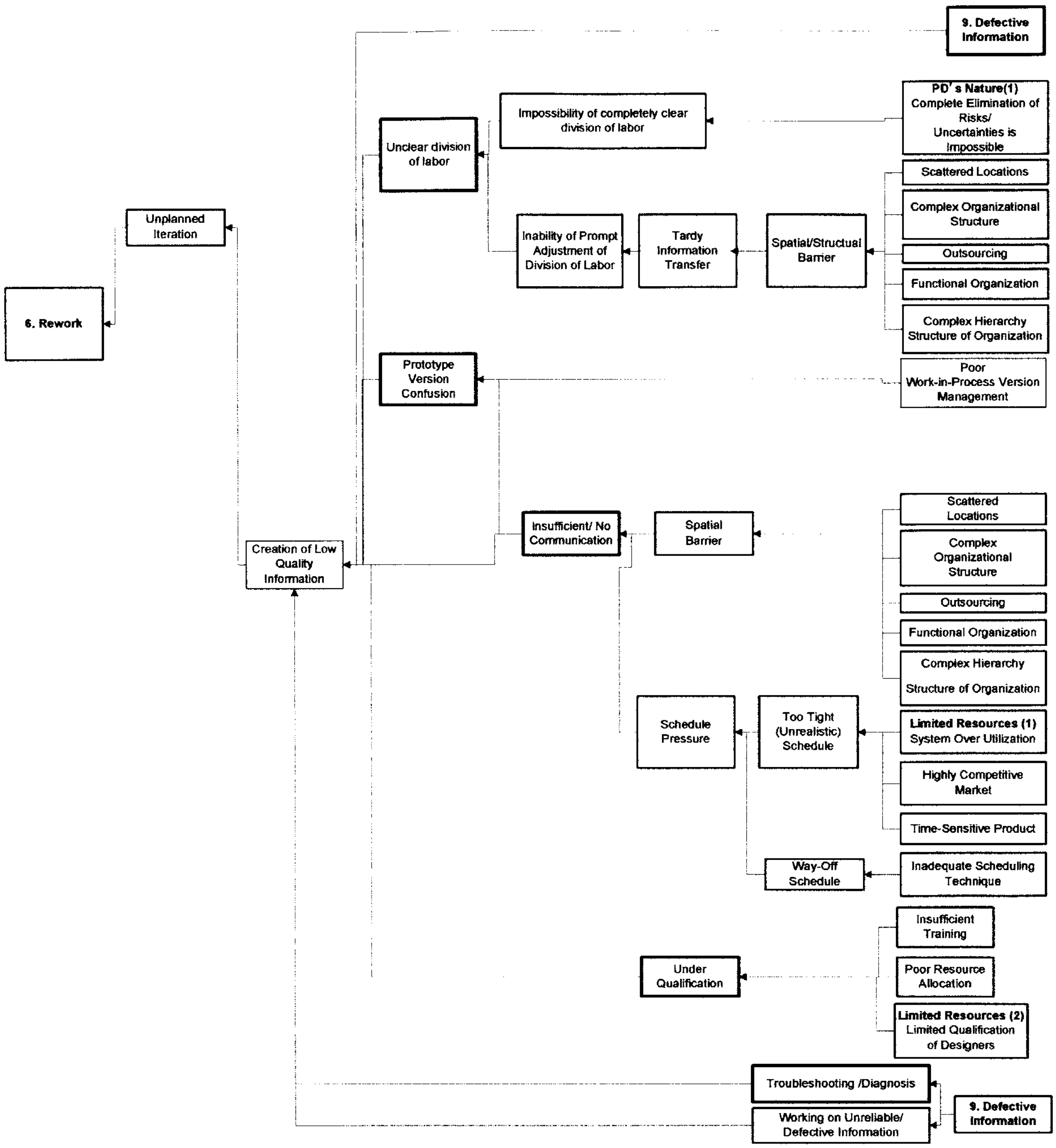


Figure 10.22 Part of the Root-Cause Analysis Diagram of Motion

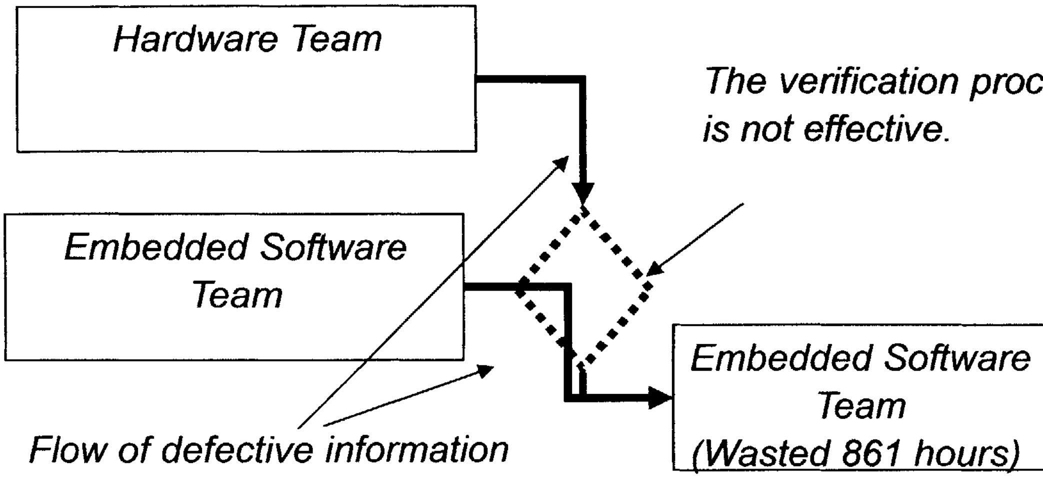


Figure 10.23Wasteful Information Flow without Effective Verification Processes

(5) Changes in Rework Time

Figures 10.24 and 10.25 show changes in the average time spent on one occurrence of rework over time. Projects A and B were put on the same graph. Project A's start week was shifted by fifteen weeks because the tracked period started after the completion of its investigation phase, which took about 13-15[1] weeks in Project B.

The two graphs proved that one occurrence of rework takes longer time near the end of a project than at the beginning of it, and their increases are exponential. Project A's curve fluctuated. This is because information came from the upstream team (hardware) periodically: the software team identified rework after some information releases such as detailed specifications releases, and prototype releases.

These results imply that problems should be identified as soon as possible. Possible solutions are the following:

1. Introducing a suitable spiral process with frequent prototyping
2. Introducing a front loaded process such as set-based concurrent engineering
3. Trying to listen to engineers. Watching for hidden problems.

[1] Not all the engineers switched themselves to the second phase at the same time in Project B.

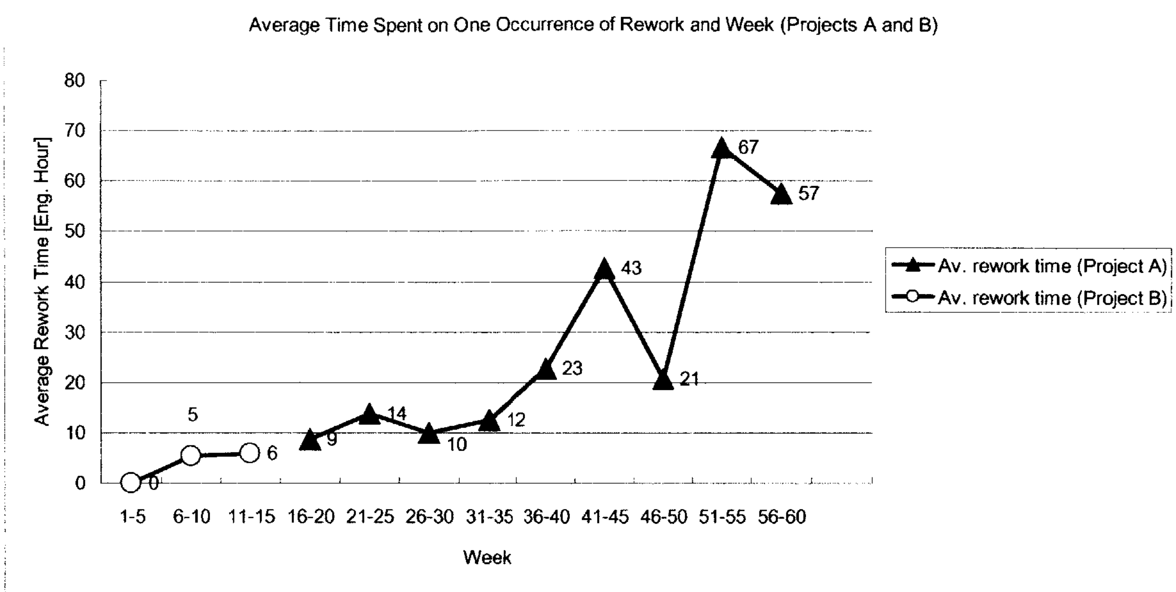


Figure 10.24 Changes in Rework Time (Projects A and B)

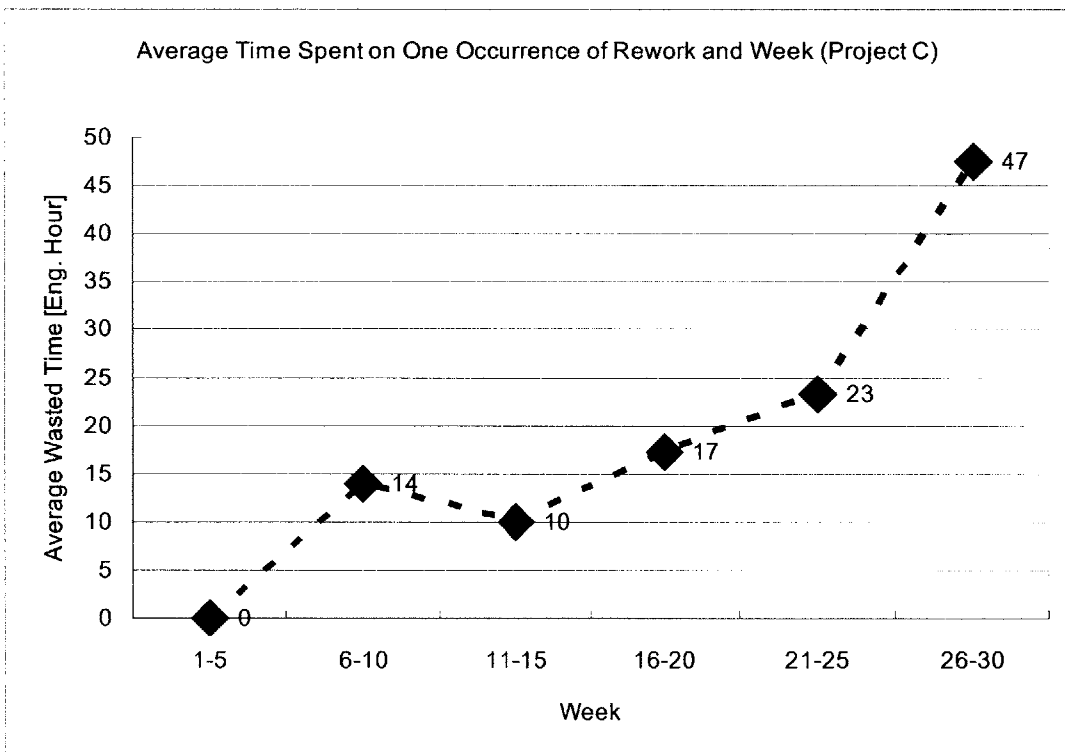


Figure 10.25 Changes in Rework Time (Project C)

10.3.8 RE-INVENTION

Re-invention was identified only in Project A, and the identified primary root cause for it was scattered locations. An engineer spent long time on a function that was similar to the one developed by another engineer.

10.3.9 HAND-OFF – WASTED ENGINEERING HOURS BY CAUSES

Figure 10.26 shows the relationship between wasted time on transportation and the corresponding causes. Transportation was significant in Project C, and the most important cause was "Absence of task owner." This represents the situation in which an engineer is forced to leave his/office for some period, and another engineer has to take over the absent engineer's tasks. This happened to Project B as well. Hand-offs take long time, especially when the previous task owner is hard to reach. Frequent hand-offs, however, leave documentation that may be helpful in the future.

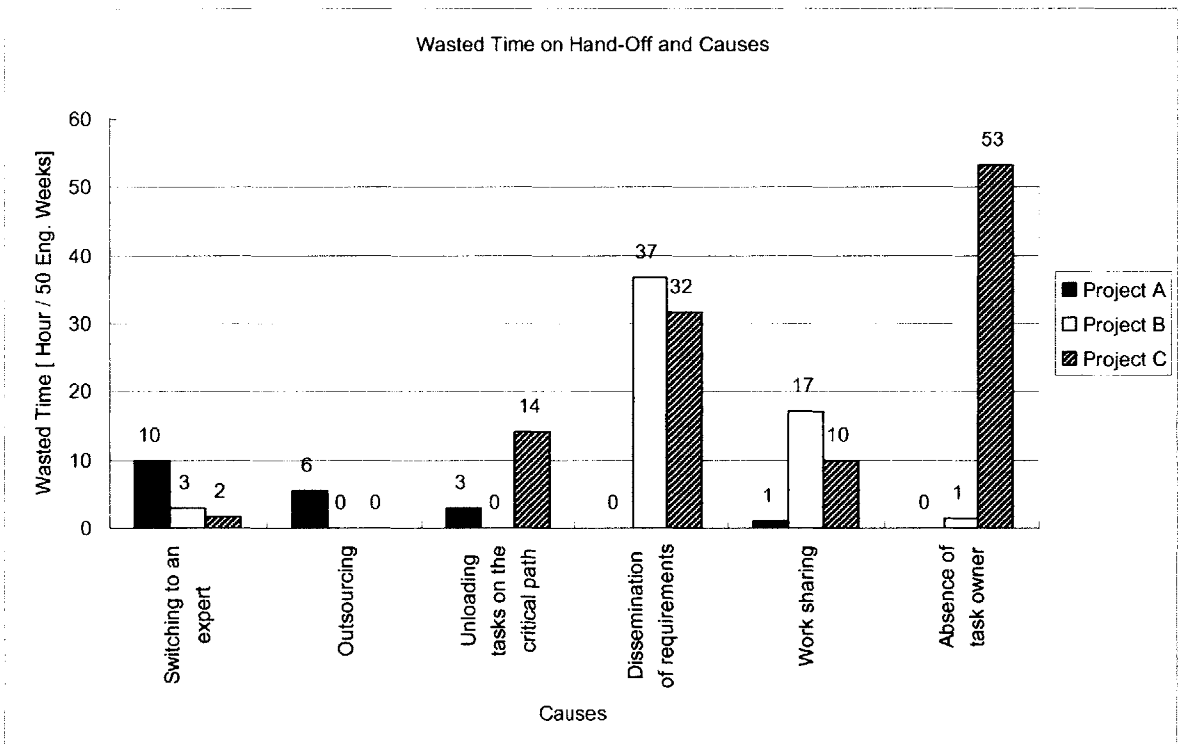


Figure 10.26 Normalized Waste Time on Hand-Off per 50 Engineering Weeks and the Corresponding Causes

10.3.10 DEFECTIVE INFORMATION

(1) Wasted Engineering Hours by Causes

(i) Under processing / errors/ lapses was the most significant cause for defective information in all the three projects (figure 10.27), and the amounts of waste time in Projects A and C by the cause were similar. Under processing means that something is missing in the output of a task: a task that was considered to be complete is not actually complete. (ii) Under qualification was the second significant cause for wasted time on defective information in Projects A and C. (iii) Insufficient communication was the third significant cause in Project A.

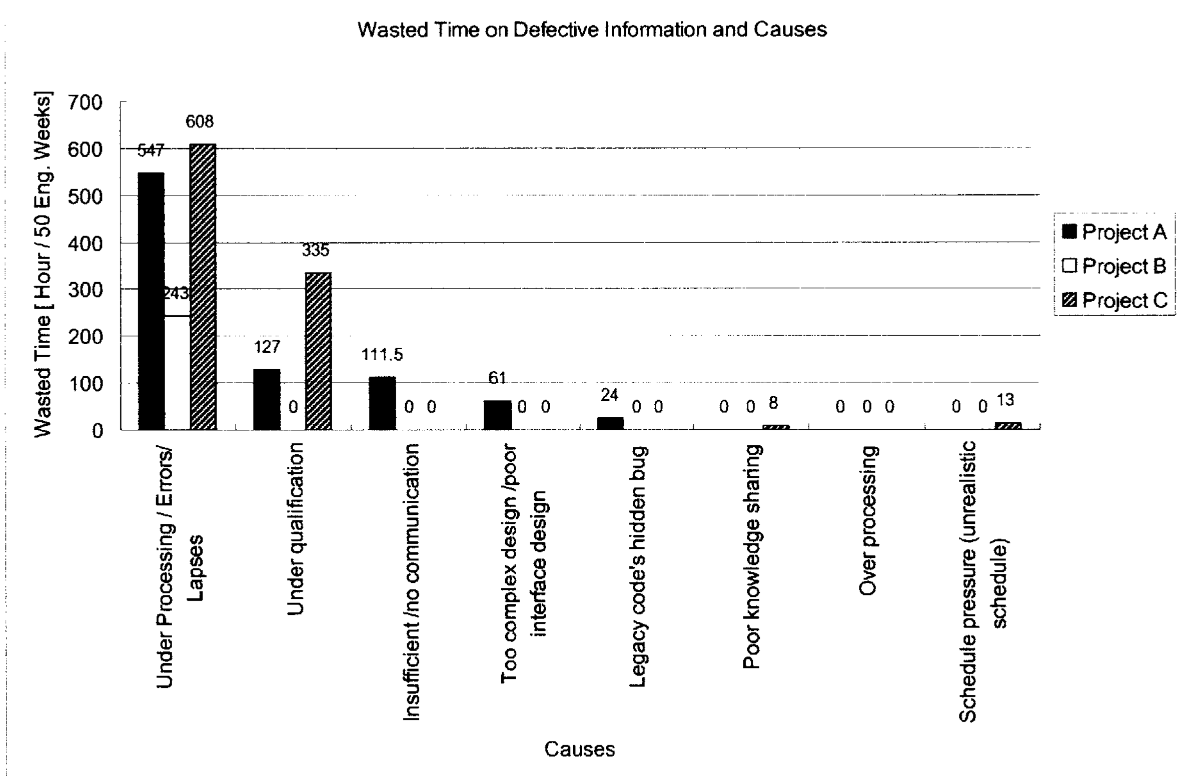


Figure 10.27 Normalized Waste Time on Defective Information per 50 Engineering Weeks and Causes

(2) Discussion

Although Projects A and C similarly suffered from (i) Under processing/ errors/ lapses, the mechanism of how they wasted time was different. Project C's case was specific to suppliers (figure 10.28). Because Company Y sells key components to Company Z, Company Y has no reliable communication channel with user companies. This is critical for them, for how their products are used differs among users, and many users use the products in the ways the designers have never taken into account. For these reasons, designers in Company Z do not exactly know every aspect of the specifications of their products. This is similar to a case in which a user of a key uses it for opening bottles: the user may complain about a new key that cannot be used for his/her unique purpose. Similar cases are prevalent in Company Z's market. Company Z sets specifications of new products without completely covering every use case. Company Y sets more detailed specifications based on the target specifications given from Company Z. Company's engineers ask questions when they encounter problems. This usually ends up finding that the task is more complicated than they thought.

It is difficult for a company in similar situation to Company Y to establish reliable communication channels with users, for all users are different, and sometimes contacting users gives users chances to speak up, leading to more workload (adding more features, etc). However, a part of the 547 hours wasted by (i) was wasted by not knowing the users. Compared to this amount of time, spending time on establishing good rapport with them with leading users by periodically visiting them would save a part of their wasted time.

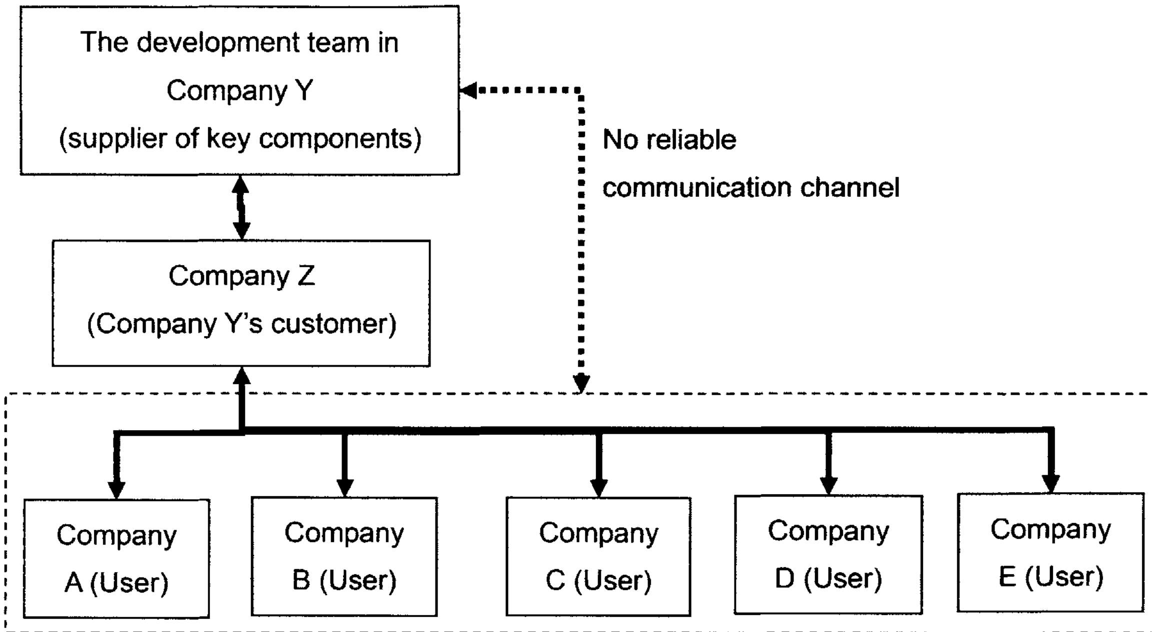


Figure 10.28 Company Y's Relationship with Users – There's Virtually No Communication Channel with Users

10.4 SUMMARY OF ANALYSIS ON NINE WASTE INDICATORS

- Value stream mapping improved in this research was applicable for quantitative measurements of nine waste indicators in all three projects.
- There was no need to add new types of waste indicators.
- Three waste indicators (over processing, rework, and defective information) were more important than the others in terms of wasted engineering hours.

  Motion was also significant in terms of wasted time, but analysis of its causes revealed that trying to reduce time spent on motion is not likely to improve product development processes significantly.
- Root cause analysis diagram was helpful for quickly identifying causes for the occurrences of waste indicators.
- Quantitative analyses of causes for waste indicators showed different patterns among companies and projects, proving that this methodology is helpful for a

company to identify its specific problems.

- Additionally, the empirical idea, “Time to solve a problem increases exponentially as time goes by,” was valid in all projects.

CHAPTER 11: RESULTS OF THE RESEARCH INVESTIGATIONS 2: ANALYSES ON INVENTORY OF INFORMATION

11.1 OVERVIEW OF THIS CHAPTER

Inventory of information measured in the three projects are compared in 11.2 through 11.5. Identified inventory of information is classified into thirteen types in 11.6. Three projects are analyzed measuring inventory by types in 11.7 and 11.8. 11.9 analyzes how quickly information got rotten in Project A. 11.10 summarizes this chapter.

11.2 NUMBER OF OCCURENCES OF INVENTORY

Figure 11.1 shows the number of occurrences of inventory of information per week in the three projects. In Projects A and B, six engineers' activities were tracked. In project C, five engineers were tracked (see table 9.1). Inventory was measured in units of engineering days, and the minimum measured period was one engineering day. In Projects A and B, on average, there was one occurrence of inventory per engineer per week. Compared to the two projects, the frequency of occurrence of inventory was much less in Project C.

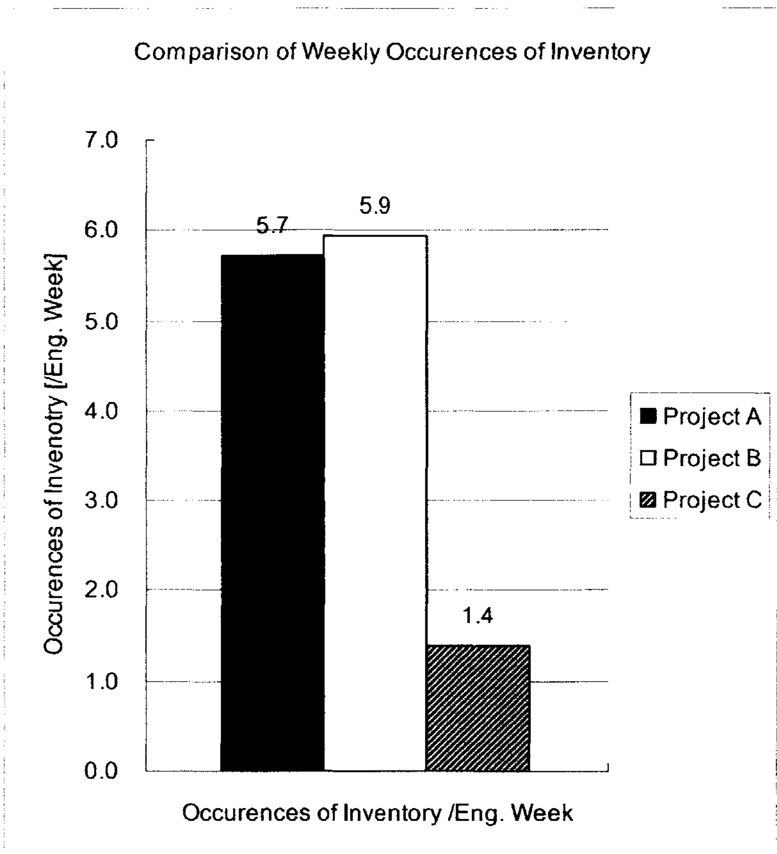


Figure 11.1 Number of Occurrences of Inventory of Information per Week

11.3 AVERAGE INVENTORY PERIOD

Figure 11.2 compares the average periods of inventory in the three projects. For example, in Project A, once a task is stopped, it took twelve days on average before it is restarted. This period is important especially in the contexts in which risks are high (see chapter 8).

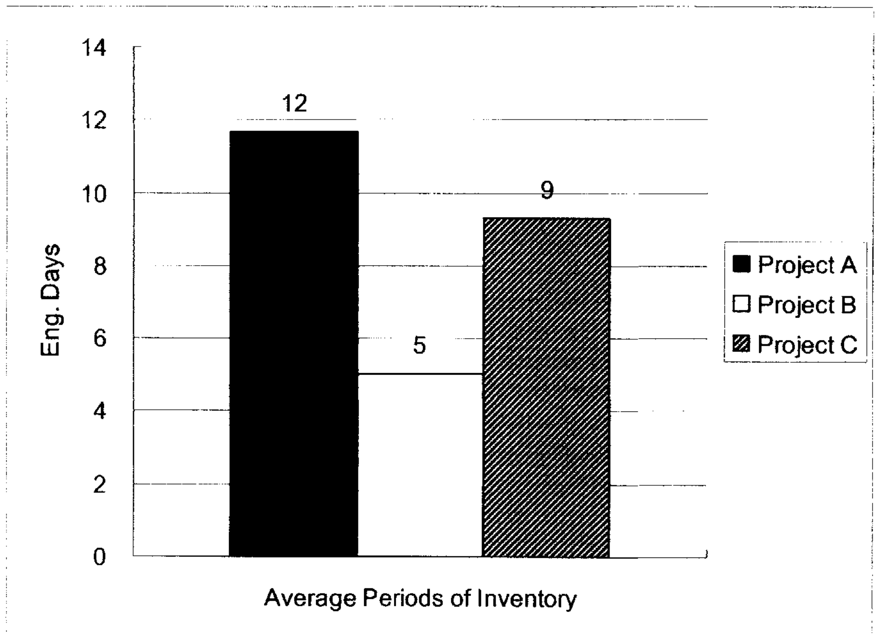


Figure 11.2 Average Periods of Inventory

11.4 TOTAL INVENTORY TIME

Figure 11.3 compares the total inventory time per engineering week in the three projects. Project A had five times more inventory time per week than Project C, although the sizes of the two teams are similar (Project A: six engineers, Project B: five engineers), and they are both in the same development phase.

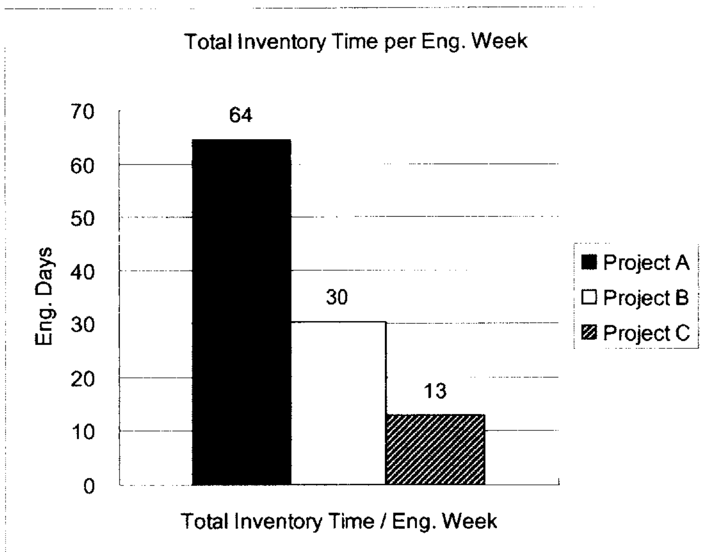


Figure 11.3 Total Inventory Time per Engineering Week

11.5 IMPLICATIONS

The result of total inventory time showed that Project A had a substantial amount of inventory time, indicating that the team suffered low throughput. This situation resembles that of manufacturing processes in 1980's, in which factories had huge amount of work-in-process inventory. Goldratt (1984) attributed the situation to inappropriate performance measurements that was prevalent then, claiming that measuring efficiencies of machines did not lead to improving productivity, but lead to unsynchronized production processes with huge amount of inventory. He recommended, in his theory, TOC, to measure inventory instead of efficiencies of machines.

Today, engineer utilization levels are monitored in many product development organizations, or at least observed by project managers and their bosses. Senior management people tend to think that there is a chance of reduction of number of engineers when they see their people idle, like a plant director who sees workers taking rests. On the other hand, idle information is not as conspicuous as idle engineers. Rather, project managers may be blamed if they fail to give their engineers fewer tasks than they can work on. Goldratt's suggestion was paradoxical in the manufacturing world twenty years ago: because measurements were focused on measuring utilization levels of machines, huge amount of inventory was not paid attention although it indicated low throughput of the production processes. Today's product development organizations may be in the same situation. Engineers are busy because their utilization level is monitored. On the other hand, idleness of information, which gets rotten with time, is significant because it is not monitored.

The results shown in figures 11.3 and 10.1 back up this idea, especially in Project A. In project A, identified waiting of engineers was only four hours in fifty weeks. On the other hand, information was waiting (as inventory) sixty four hours per engineering week. Projects B and C had the same tendency, although they are not as significant. These results imply that today's product development processes are in similar situation as manufacturing processes twenty years ago. Toyota production system has seven wastes (Ohno, 1978). TOC has only three metrics: throughput, inventory, and operating cost (Goldratt, 1984). Toyota production system tries to control both waiting and inventory, while TOC ignores waiting. The results in figures 11.3 and 10.1 implies that today's product development activities are so premature as manufacturing a few decades ago that applying all the seven wastes in product development is too early. The more detailed analysis of inventory in 11.8 reveals the existence of striking similarities in manufacturing processes a few decades ago and Project A's development process.

11.6 CLASSIFICATION OF IDENTIFIED INVENTORY

The identified inventory of information falls into the thirteen types as follows:

(1) Type 1: Taking care of a more urgent task in the project

(2) Type 2: Switching to a higher-priority task outside of the project

(3) Type 3: Waiting for information from another task

(4) Type 4: Review/ testing work

(5) Type 5: Day off

(6) Type 6: Maintenance of Documents

(7) Type 7: Rework discovery

(8) Type 8: Other engineers' availability

(9) Type 9: Downstream engineer's availability

(10) Type 10: Waiting for an answer

(11) Type 11: Ambiguous information

(12) Type 12: Limited availability of tool/board/system

(13) Type 13: Others

(1) Type 1: Taking care of a more urgent task in the project

This occurs when a group/ an engineer needs to stop working on a process and switch to another process in the same project because the latter process has higher priority. Typical appearance of this in VSM is shown in figure 11.4.

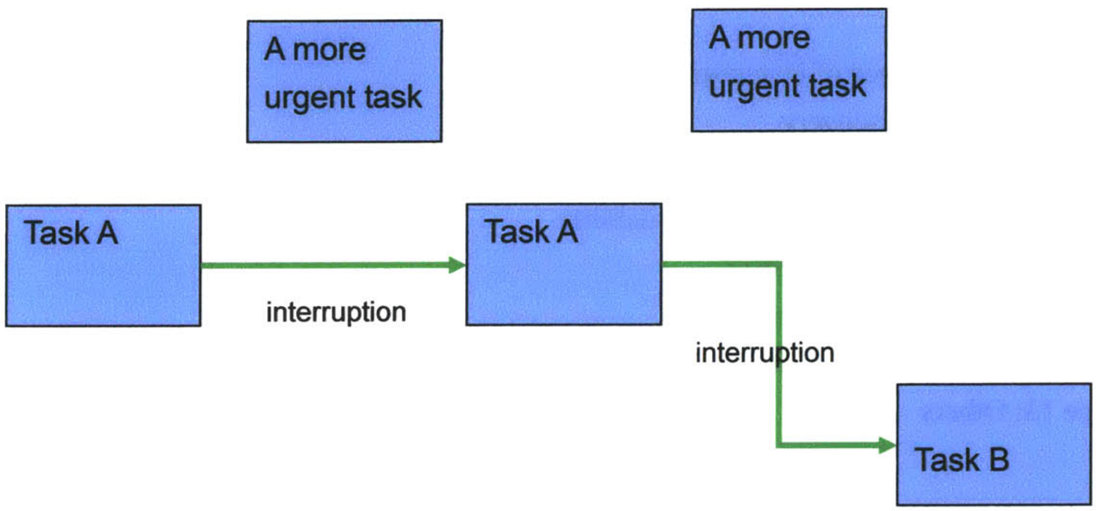


Figure 11.4 Example of Type 1 Inventory

Example from the investigation

I328 in figure 11.5 is an example of this type of inventory of information. The implementation phase of XXX FW (XXX is a function's name; it cannot be shown due to the confidential agreement with the company) had not been worked on for five days because the engineer needed to finish the documentation of another task. He needed to switch to the latter task because its due date was closer than that of the implementation phase of XXX FW. As can be seen in figure 11.5, XXX FW was interrupted several times mainly by tasks inside of the project.

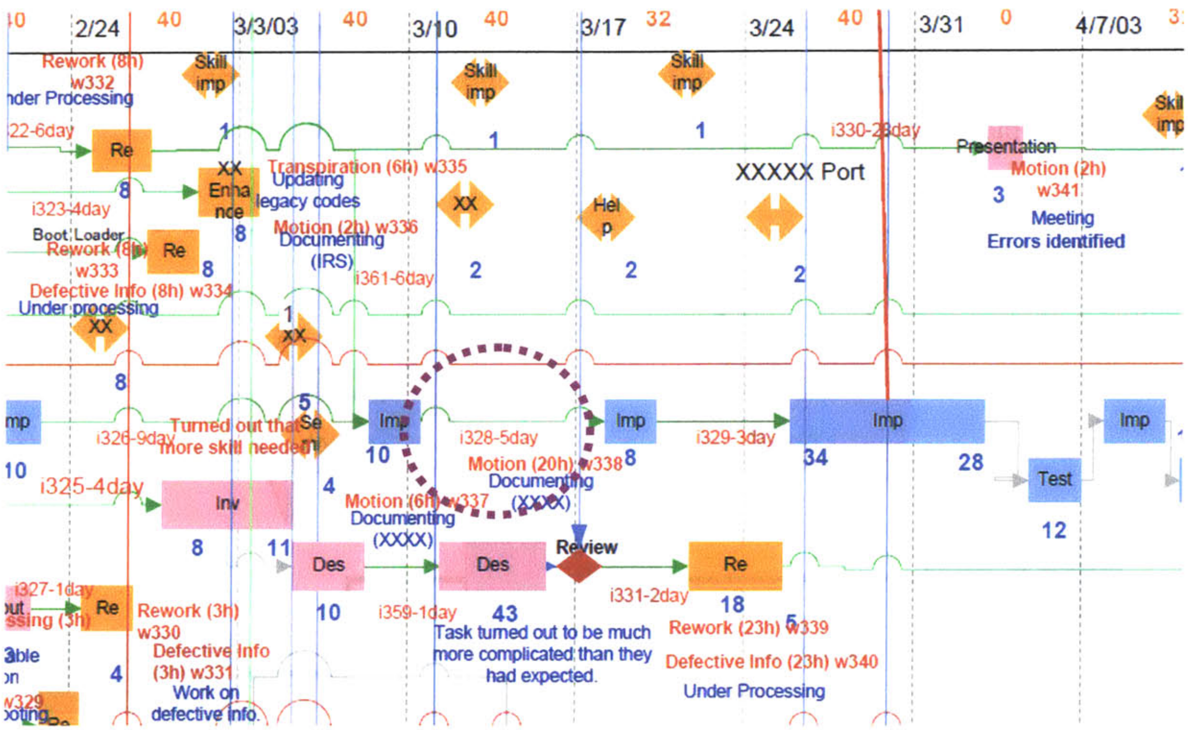


Figure. 11.5 I328, Inventory of Information at the Center of This Figure was Caused by an Interrupting Event from Inside of the Project (Engineer Y, Week11)

(2) Type 2: Switching to a higher-priority task outside of the project

This type is the same as type (2) except that the interrupting processes do not belong to the project. The typical appearance of this in VSM is shown in figure 11.6.

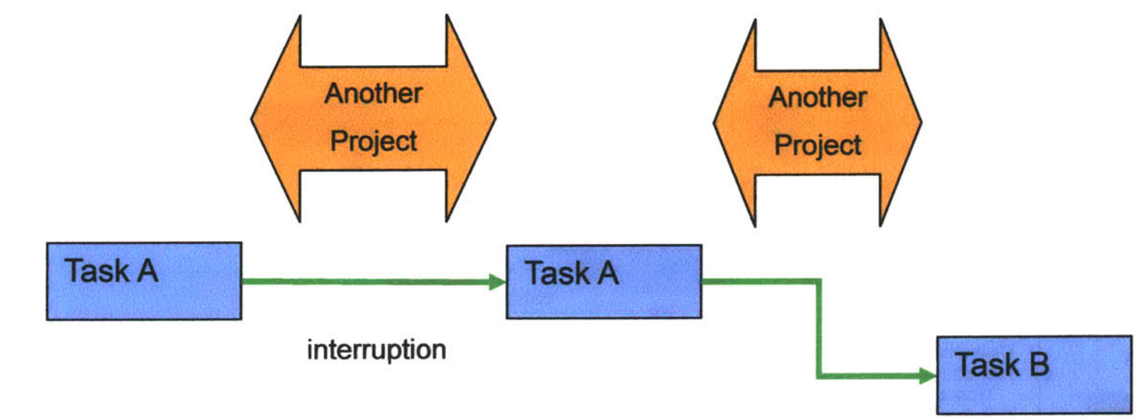


Figure 11.6 Example of Type 2 Inventory

Example from the investigation

I408 in figure 11.7 is an example of this type of inventory of information. System-Level Services Task was interrupted by a supporting work for another project. This interruption caused i408.

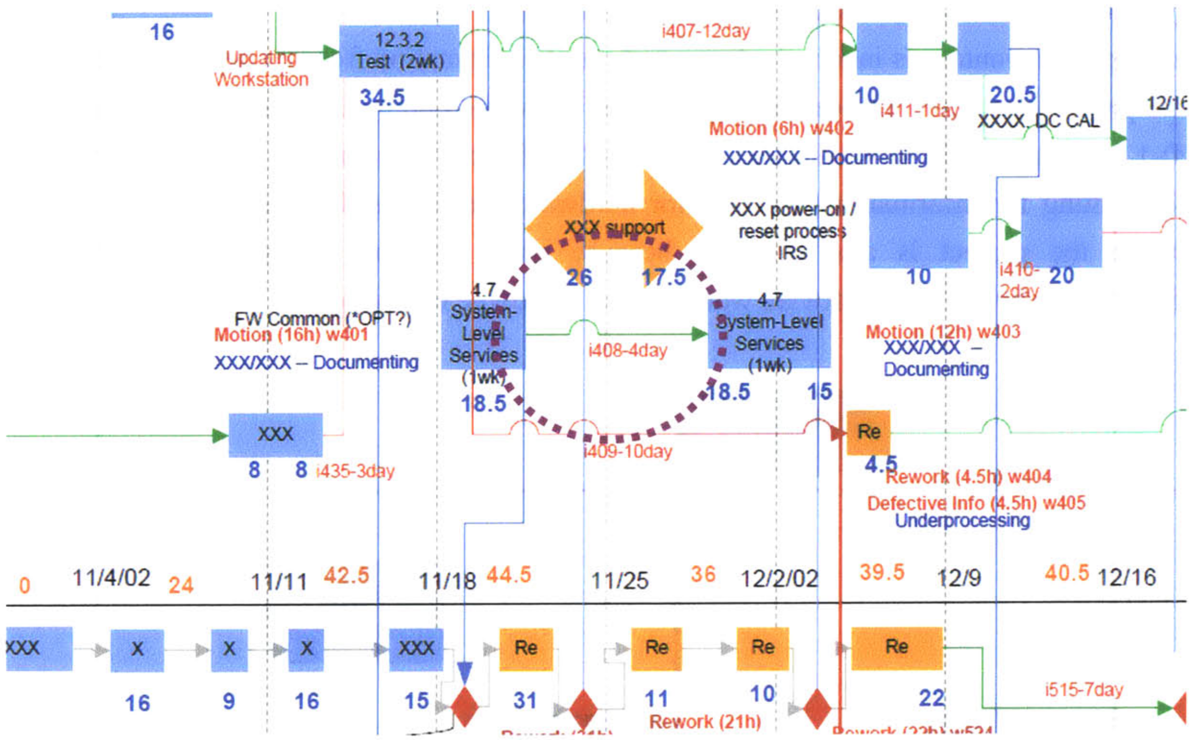


Figure 11.7 Example of Switching to Higher-Priority Task outside of the Project – System-Level Services Task was Interrupted by a Support Work Outside of Project A

(3) Type 3: Waiting for Information from another Task

This type of inventory is incurred when a task needs information from its dependent task(s) to be processed further. The dependent task may be worked on by the same group/ engineer ((3)-1 in figure 11.8) or by different one ((3)-2 in figure 11.8). In this classification, not only waiting for information, but also for hardware prototypes that had been developed through the project is also included because what software engineers need is the information whether their software works on the prototype hardware.

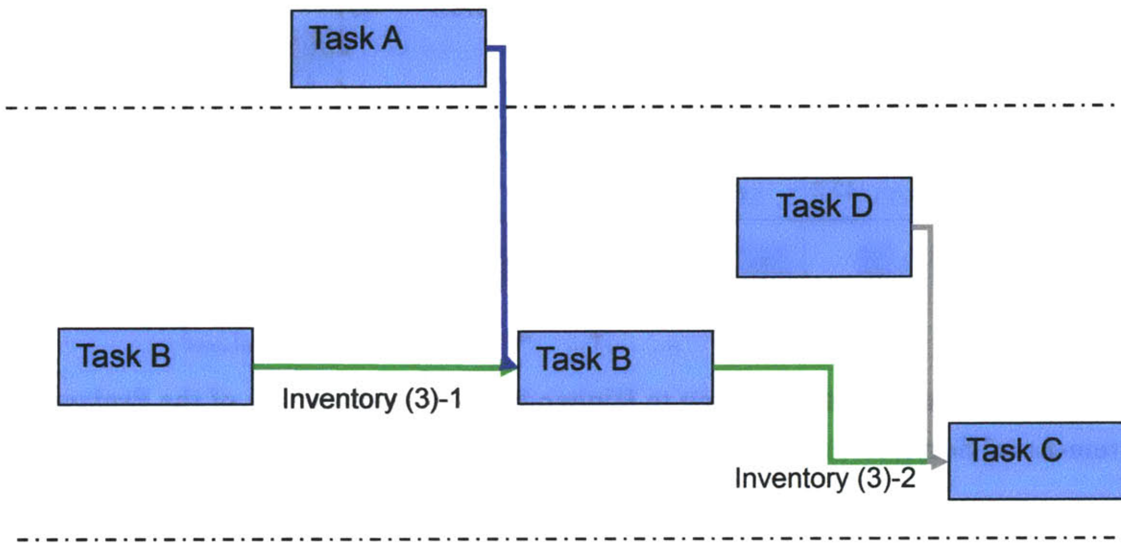


Figure 11.8 Waiting for Information from another Task

Examples from the investigation

(1) Example of Inventory (3)-1

Figure 11.9 shows two appearances of inventory of information of this type. The two tasks circled in this figure were testing processes; both needed an updated prototype of a hardware component that had been developed by the hardware development team. Especially, i539 was kept for as long as fifty four days because the hardware component it was waiting for had been redesigned due to serious defects.

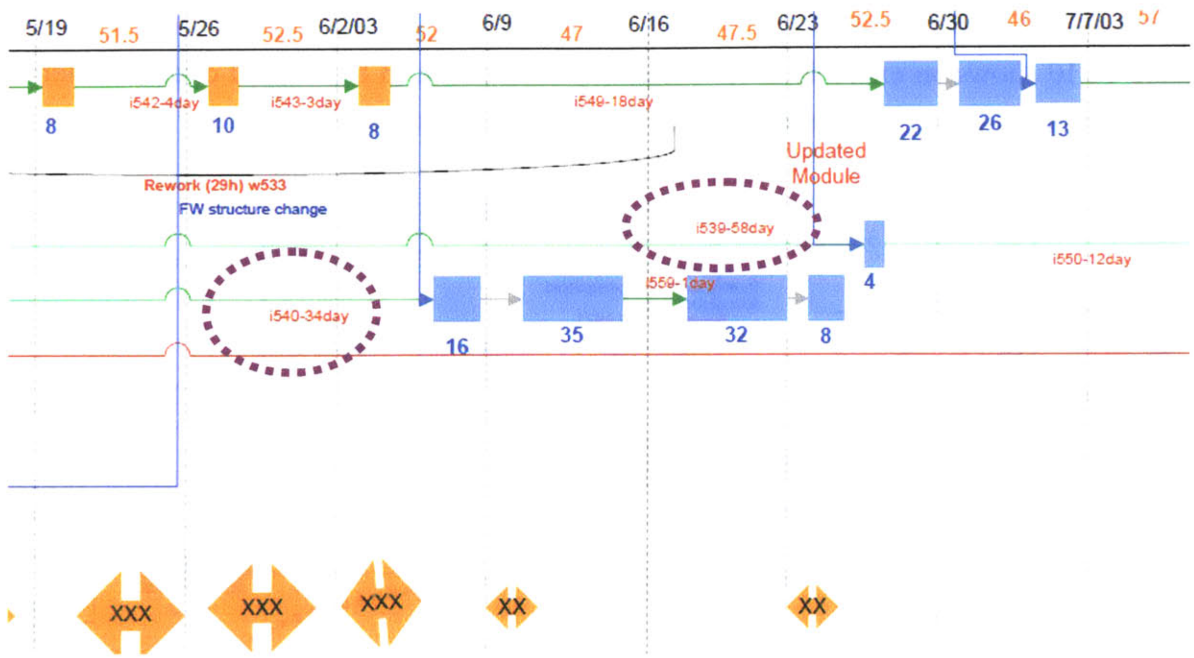


Figure 11.9 Example of Inventory (3)-1 – The Two Tasks Circled in This Figure were Both Testing Processes.

(2) Example of Inventory (3)-2

Figure 11.10 shows an example of inventory (3)-2. The task in the dashed circle was left untouched until the downstream task got all the information it needed.

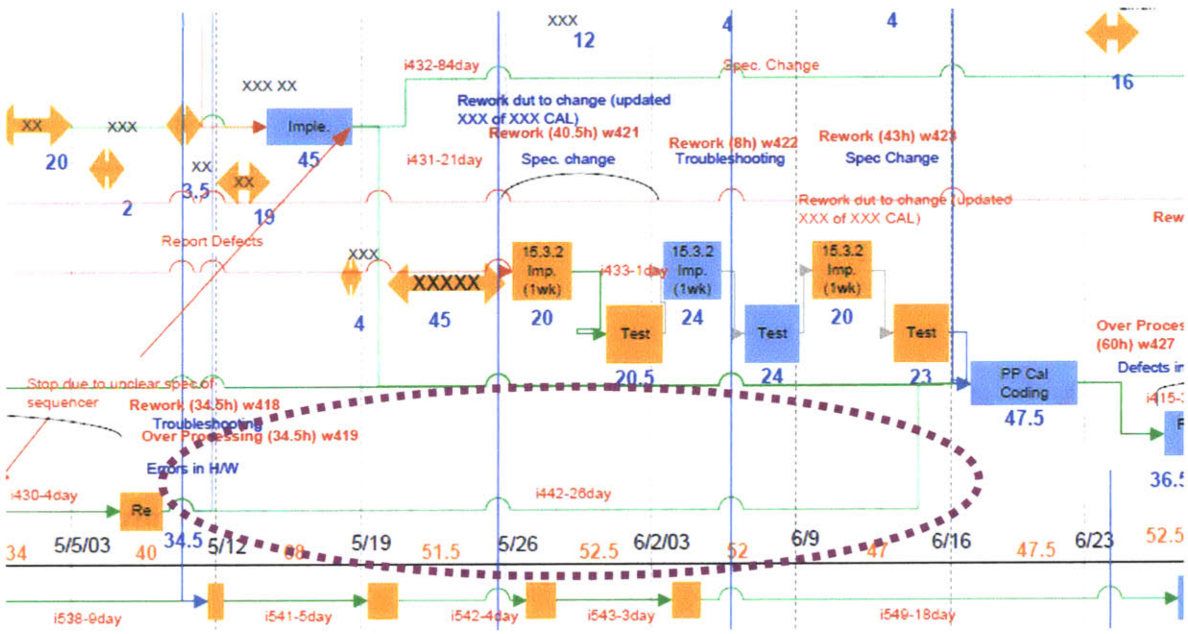


Figure 11.10 Example of Inventory (3)-2 – the Task in the Dashed Circle was Left Untouched until the Downstream Task Got All the Information It Needed.

(4) Type 4: Review/ Testing Work

This type of cause was separated from type 1 because review/testing task had often (25 times) appeared throughout the development phase (figure 11.11). Review/ testing work has high priority because of the following reasons:

- Conducting it earlier reduces rework discovery time.
- Several tasks may be processed based on tentative information from the task yet to be reviewed.
- Some types of reviewing involve several busy key-players, causing difficulty in making date changes.

Examples from the investigation

Figure 11.11 shows examples of inventory of information caused by review/ testing work. I016 was caused solely by an external specifications review work. The interrupted task was additionally interrupted by two other external specifications reviews, one code review, and some other tasks. Since this engineer's role is working manager, he could scarcely have concentrated on one design task without interruptions.

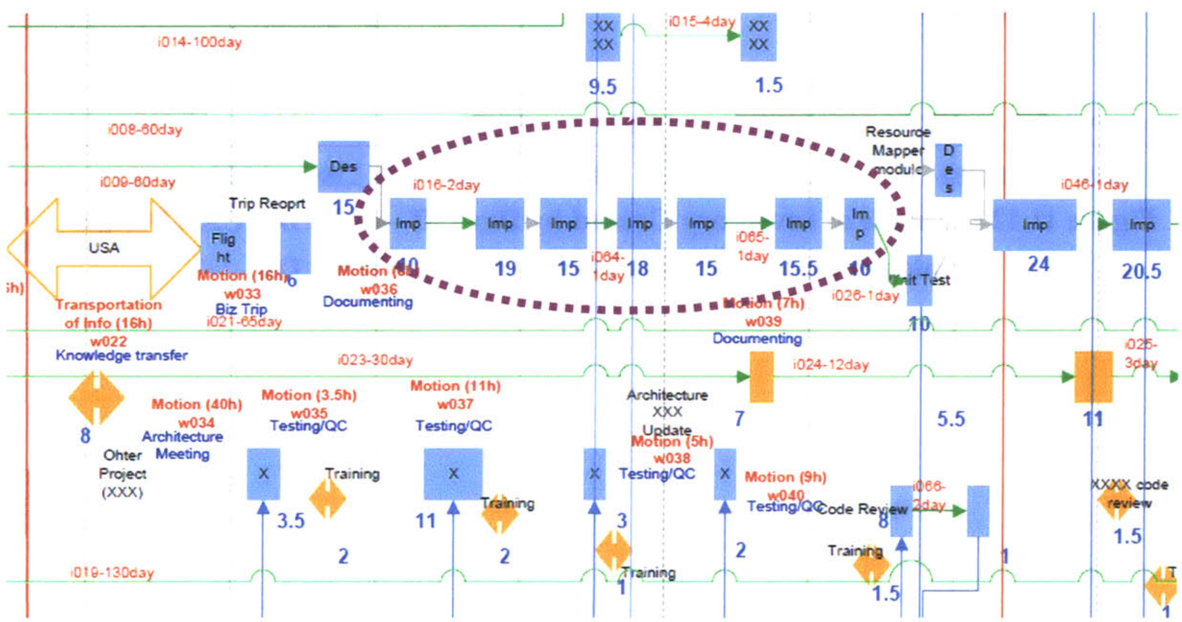


Figure 11.11 Example of Inventory Caused by Review/ Testing Work

(5) Type 5: Day Off

Although day offs is not waste of engineering hours, they cause inventory of information. Figure 11.12 shows an example of inventory caused by day off – XXX DC CAL Design/ Implementation task was interrupted by a week off. Since the engineer took a week off then, the total labor hours was zero (see the number in orange in figure 11.12).

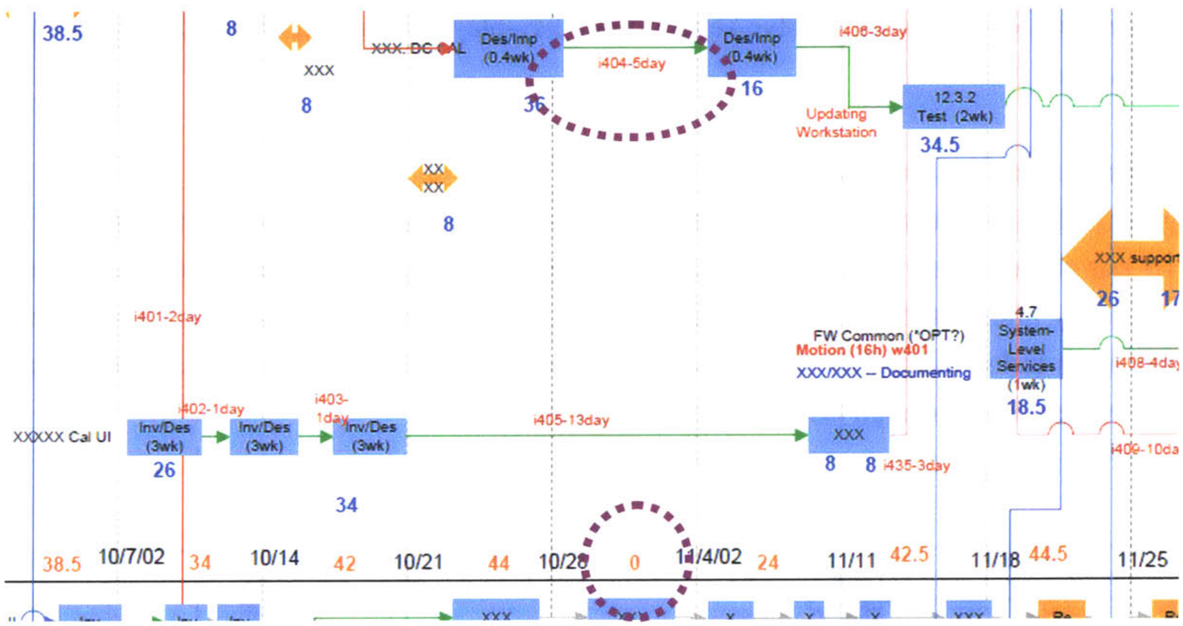


Figure 11.12 Example of Inventory Caused by Day Off – PMU DC CAL Design/ Implementation Task was Interrupted by a Week Off.

(6)Type 6: Maintenance of Documents

As is shown in figure 11.13, this type of inventory appears when engineers postpone documentation and its completion. Although Glodratt (1997) argues that documentation should not be done immediately after every task because it can cause delay on the critical path, delay in documentation could incur time to remember and memory loss.

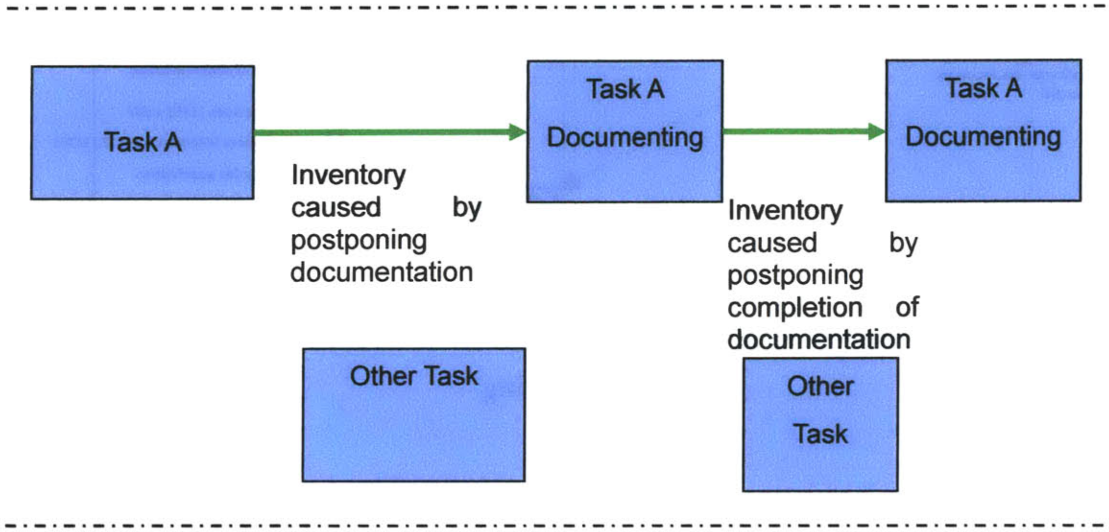


Figure 11.13 Inventory of Information Caused by Maintenance of Documenting

Example from the investigation

Figure 11.14 shows examples of inventory of information caused by postponing documentation.

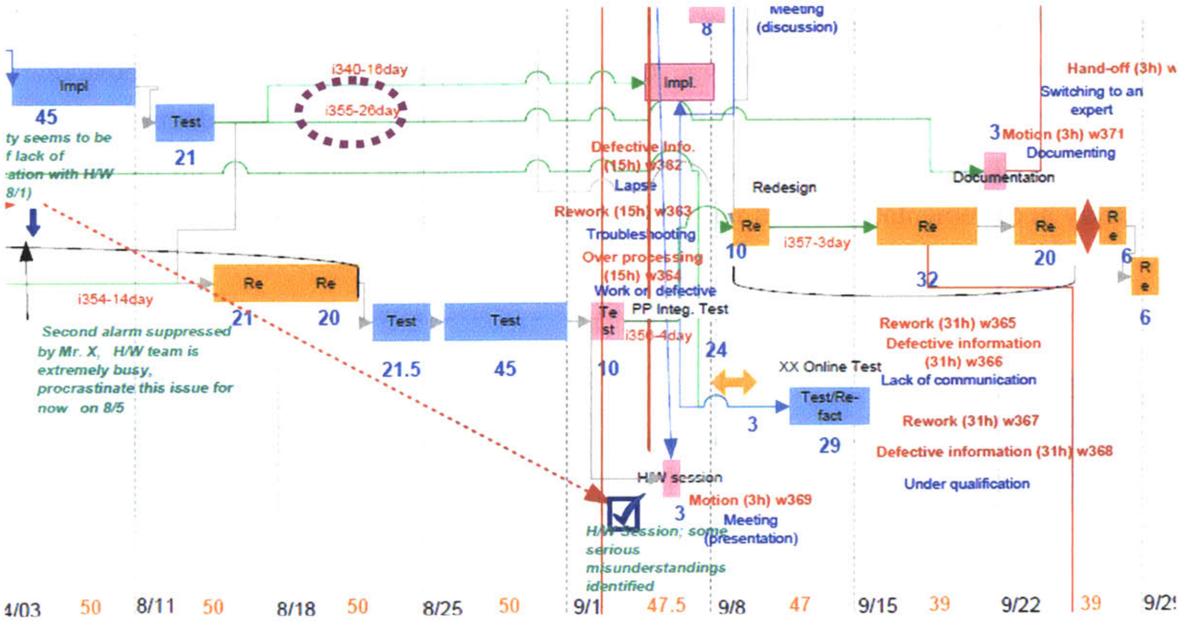


Figure 11.14 Example of Inventory of Information Caused by Maintenance of Documenting

(7)Type 7: Rework Discovery

This inventory appears when a task that was perceived to be completed is reworked (figure 11.15). Since this inventory is related to rework discovery time – one of the important metrics in system dynamics – reduction of this inventory could prevent rework caused by "upstream changes" (pp. x) and rework caused by "working on unreliable/defective info."

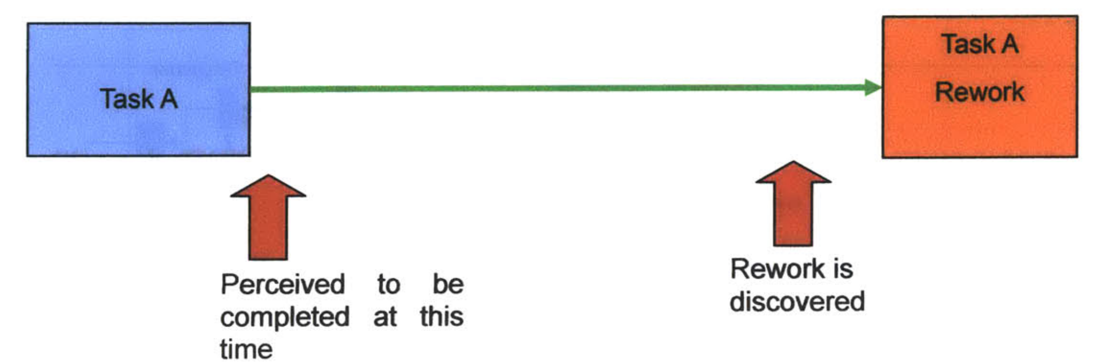


Figure 11.15 Inventory of Information Caused by Rework Discovery

Example from the investigation

Figure 11.16 shows an example of inventory of information caused by rework discovery. In this case, a task needed to be reworked because the engineer was notified a change in H/W design on which his task was dependent.

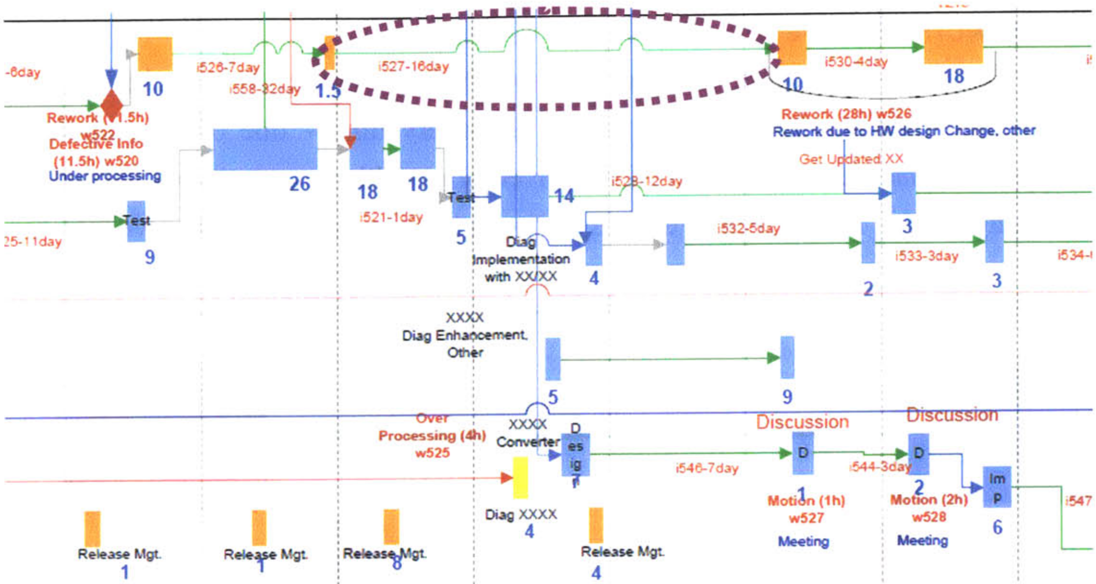


Figure 11.16 Example of Inventory of Information Caused by Rework Discovery

(8) Type 8: Other Engineers' Availability

Example from the investigation

Working in a team, sometimes tasks need to be stopped until its output is confirmed acceptable for the upstream engineer. This can take long because the needed engineer may have some high-priority tasks. Figure 11.17 shows an example of this type. In this case, the testing task needed to be reviewed by another engineer who is taking care of its dependent task. If the needed engineer is to process a downstream task, inventory of information is classified as (9): downstream engineer's availability.

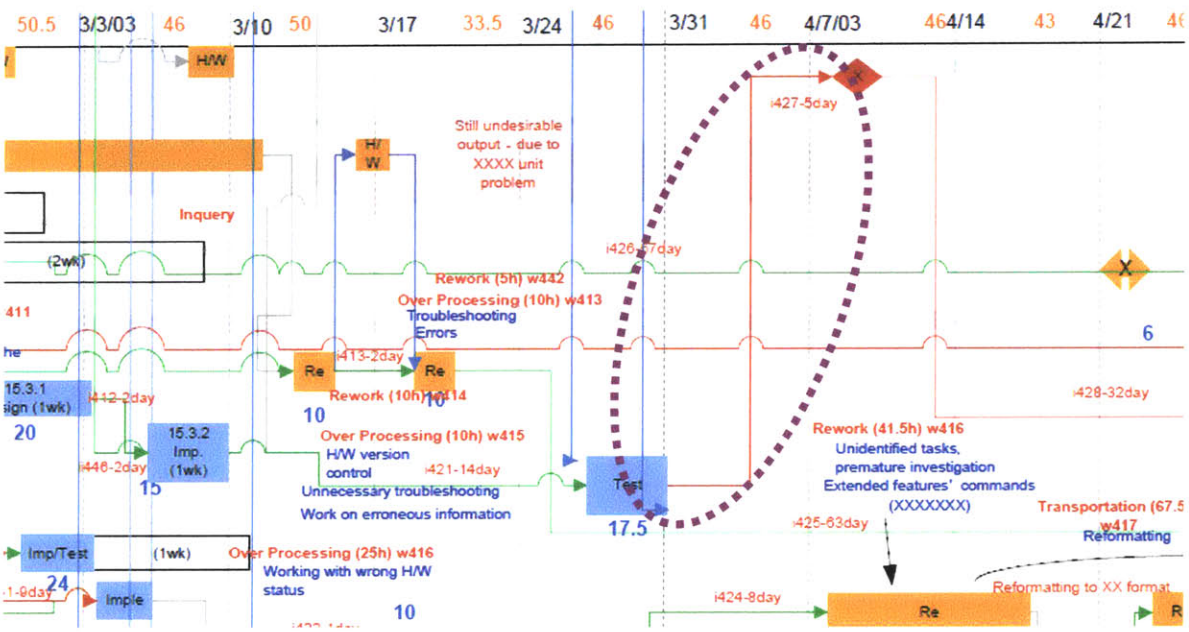


Figure 11.17 Example of Inventory of Information Caused by Other Engineer's Availability

(9) Type 9: Downstream Engineer's Availability

As is shown in figure 11.18, this type of inventory appears when handed-off information is left untouched for some period.

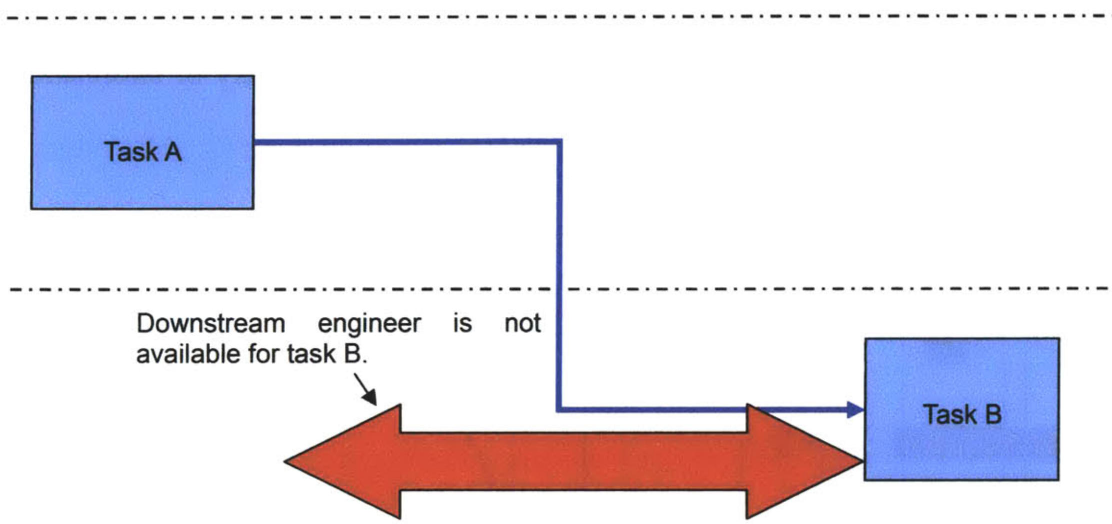


Figure 11.18 Inventory of Information Caused by Downstream Engineer's Availability

Example from the investigation

Figure 11.19 shows an example of this type. In this case, it took nine days before the engineer used the information from the H/W team.

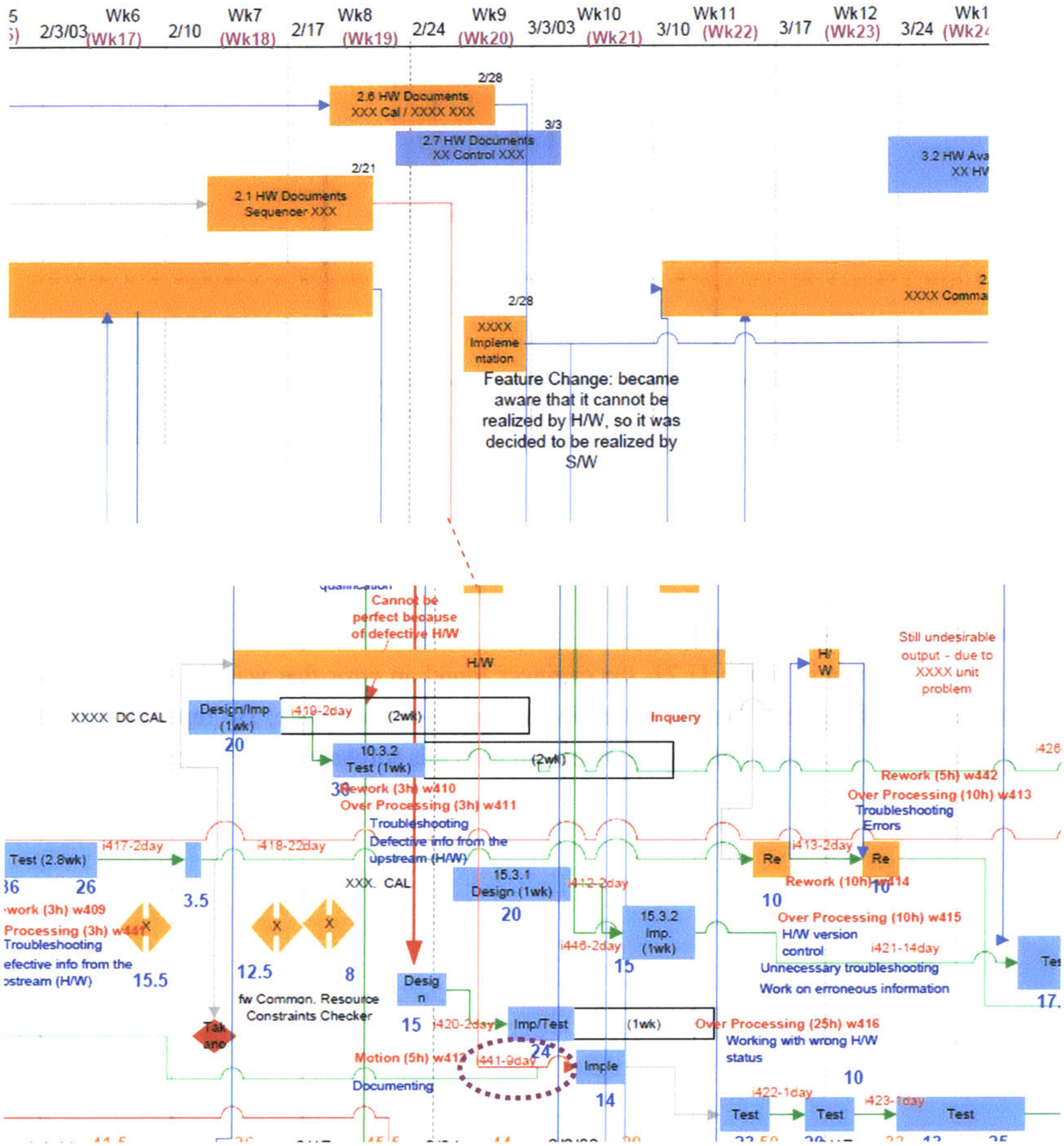


Figure 11.19 Example of Inventory of Information Caused by Downstream Engineer's Availability

(10) Type 10: Waiting for an answer

As is shown in figure 11.20, this type of inventory occurs when an engineer finds a question, which takes some time before he/she gets an answer for it.

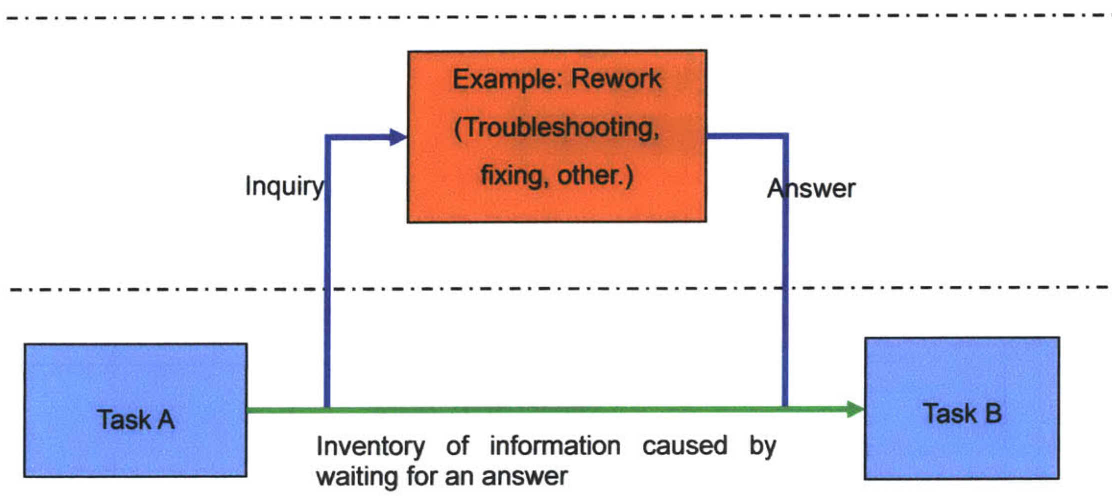


Figure 11.20 Inventory of Information Caused by Waiting for an Answer

Example from the investigation

In the case shown in figure 11.21, the engineer found an error that was not caused by his software. Thinking that there is something wrong in the hardware, He reported it to the H/W engineer. The task could not be processed until this software engineer got an answer.

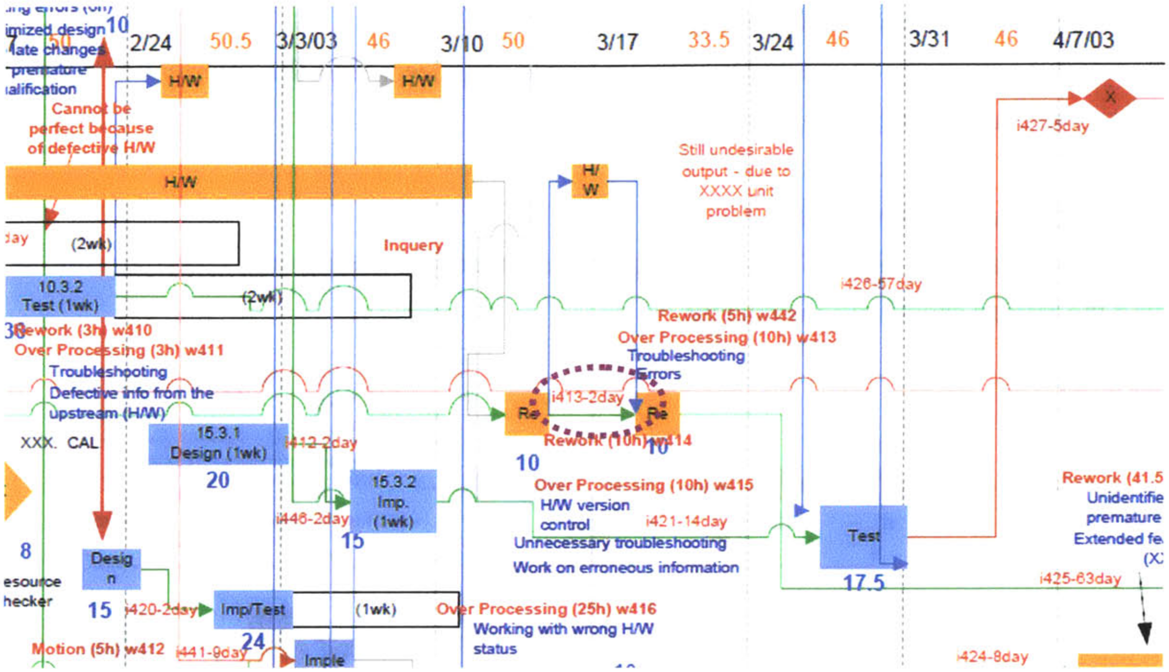


Figure 11.21 Example of Inventory of Information Caused by Waiting for an Answer

(11) Type 11: Ambiguous information

This type of inventory of information occurs when engineers find the information on his hand too unclear or unreliable to be used.

Example from the investigation

In the case shown in figure 11.22, an engineer stopped processing two tasks because he found the related specifications unclear.

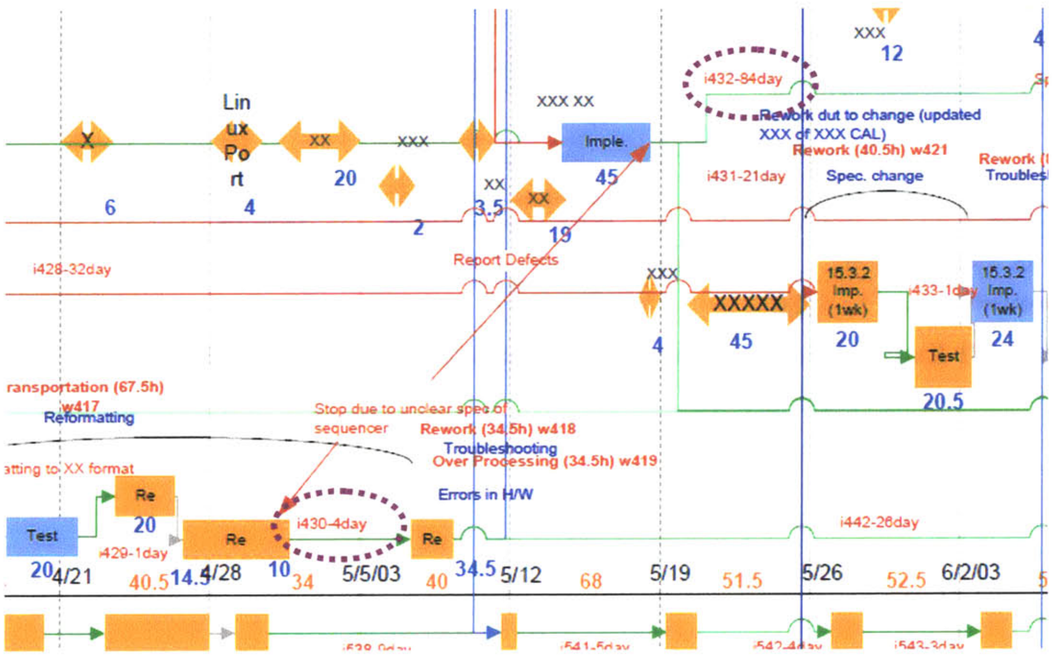


Figure 11.22 Example of Inventory of Information Caused by Ambiguous Information

(12) Type 12: Limited availability of tool/board/system

Example from the investigation

In the example shown in figure 11.23, a limited number of prototype H/W boards were shared by the software team, causing inventory of information against engineers' will.

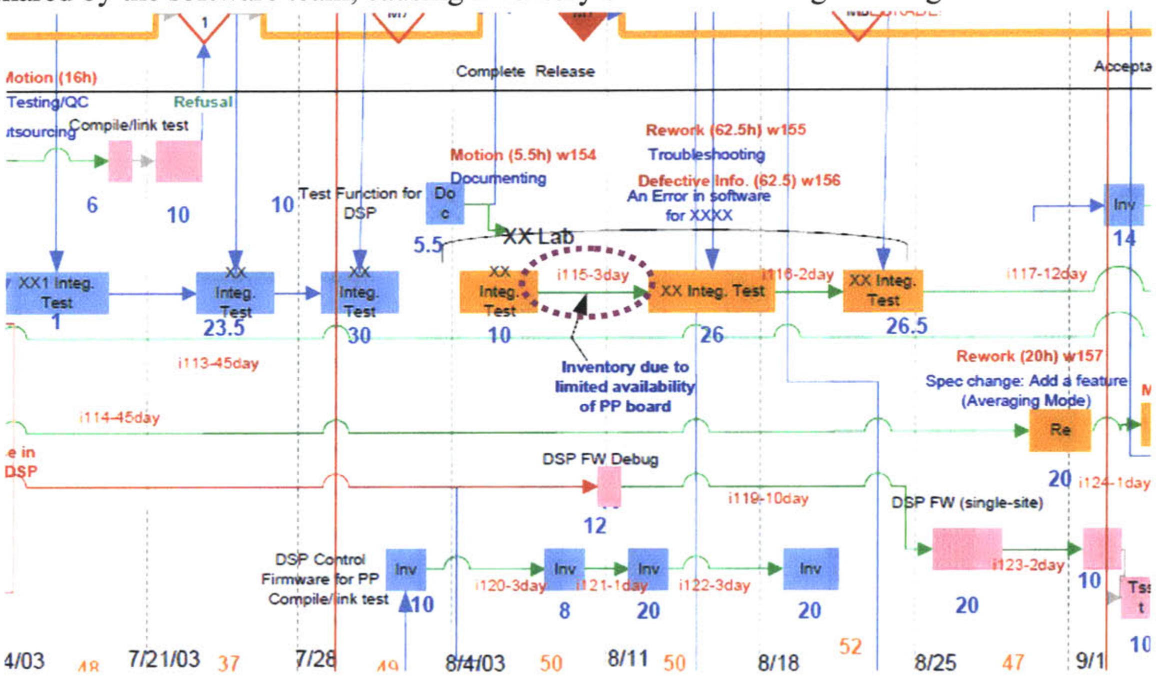


Figure 11.23 Example of Inventory of Information Caused by Limited Availability of Tool/Board/System

11.7 DISTRIBUTION OF INVENTORY TYPES AMONG ENGINEERS

Figures 11.24, 11.25, and 11.26 compare the distributions of types of inventory of information by engineers. These results revealed that the distributions differ among engineers and projects.

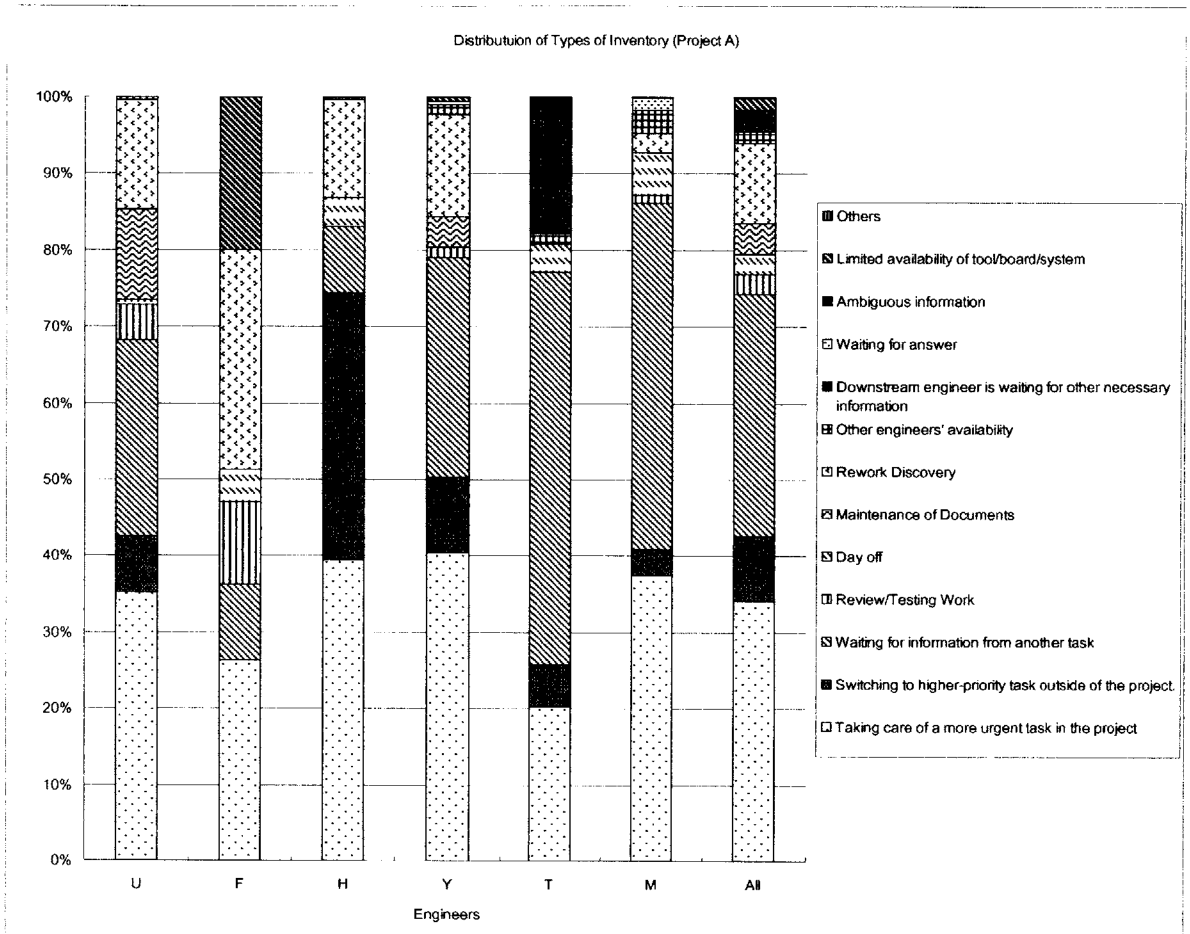


Figure 11.24 Distribution of Types of Inventory (Project A)

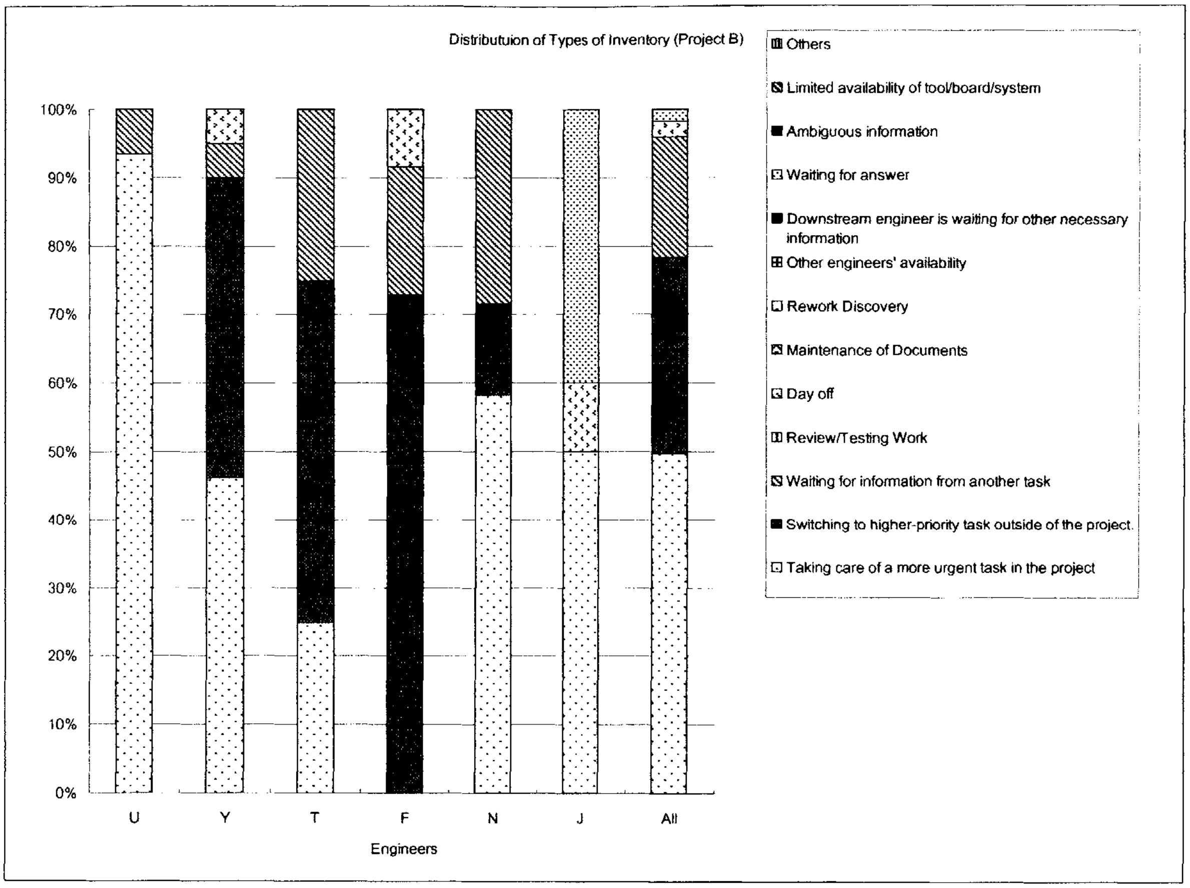


Figure 11.25 Distribution of Types of Inventory (Project B)

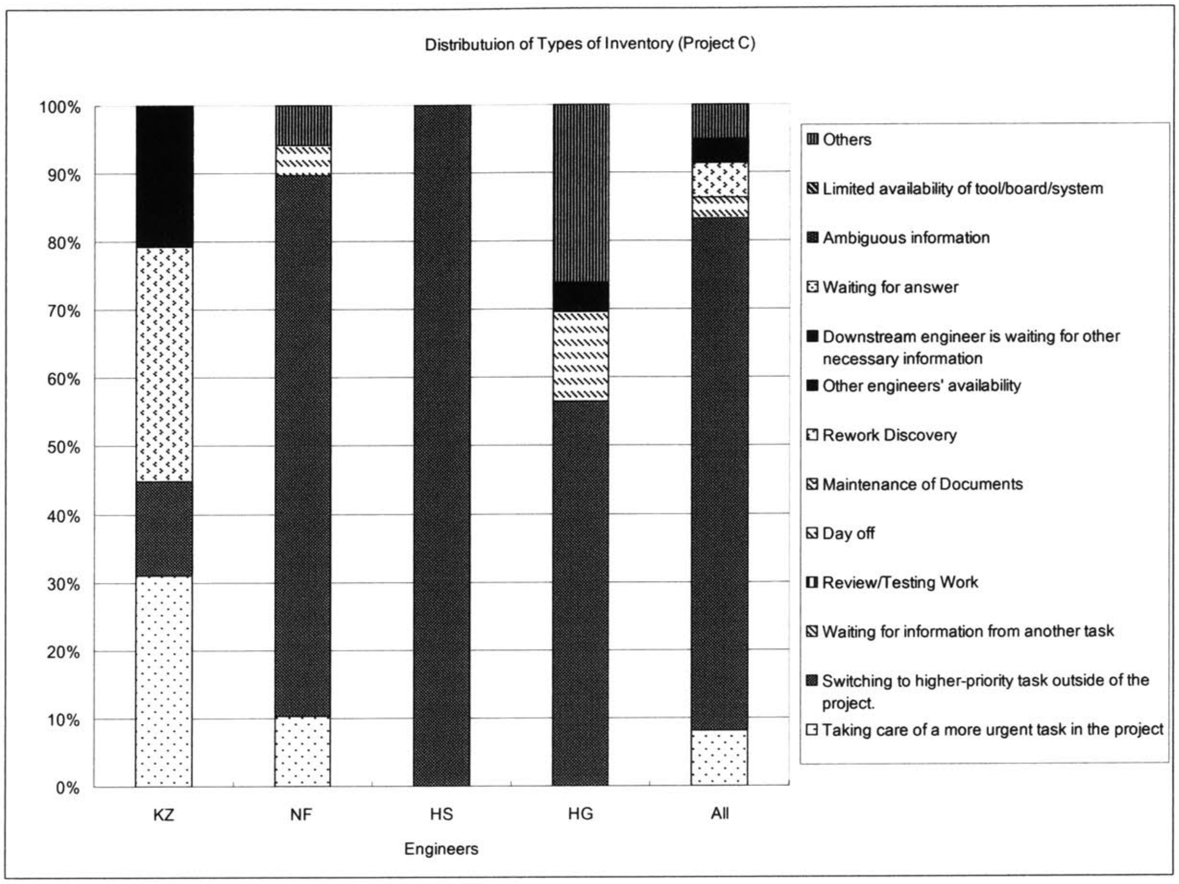


Figure 11.26 Distribution of Types of Inventory (Project C)

11.8 ANALYSIS ON INVENTORY OF INFORMATION BY ITS TYPES

11.8.1 NUMBER OF OCCURENCES

Figure 11.27 compares the three project's number of occurrences of inventory by types. In Project A, type 1, taking care of a more urgent task in the project, was the most frequent type of inventory. In Project B, type 2, switching to higher-priority task outside of the project, was the most frequent inventory of information, implying that the engineers were frequently interrupted by other commitments. Project C had no outstanding type of inventory in terms of number of occurrences, the most frequent one being type 1.

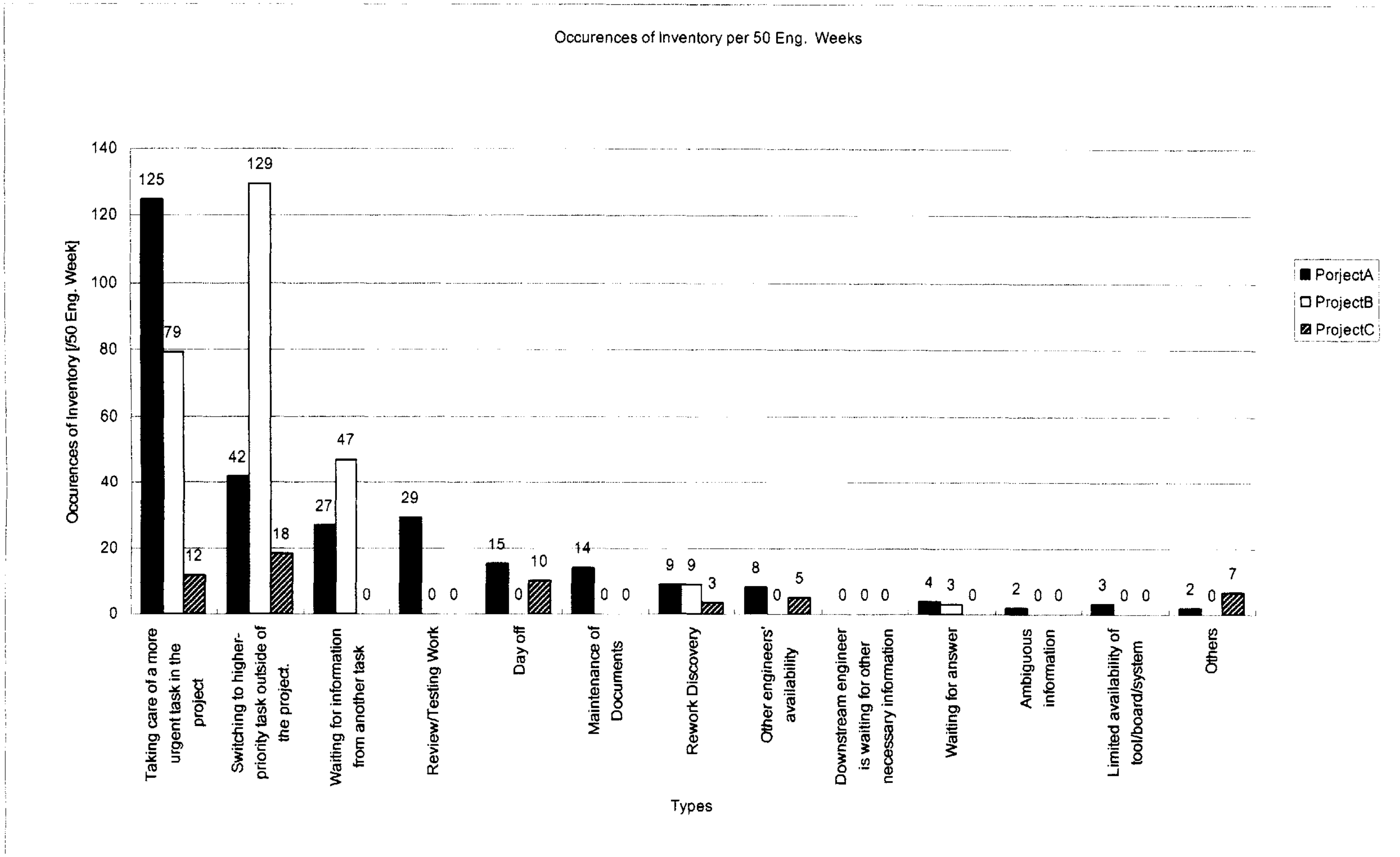


Figure 11.27 Normalized Occurrences of Inventory of Information per 50 Engineering Weeks and the Corresponding Types

11.8.2 AVERAGE INVENTORY PERIODS

Figure 11.28 compares the three project's average lengths of inventory by types. In Project A, "Ambiguous information (type 11)," "Rework discovery (type 7)," and "Waiting for information from another task (type 3)," were the most outstanding ones with about the average of forty engineering days. This is because the embedded software team depends on its upstream process, hardware development. Rework discovery sometimes take long time because some types of bugs in software cannot be identified without hardware prototypes. Software engineers sometimes suspend their tasks until prototypes become available. Project B had much less average inventory periods of the three types above. This was mainly because there were no major changes in hardware specifications. Project C was similar to Project B in terms of the three types of inventory, for the project did not involve major hardware changes. In Project C, downstream engineer's availability (type 9) caused the longest average inventory period, 21 days. This implies that Project C is managed in a way not to interrupt engineers.

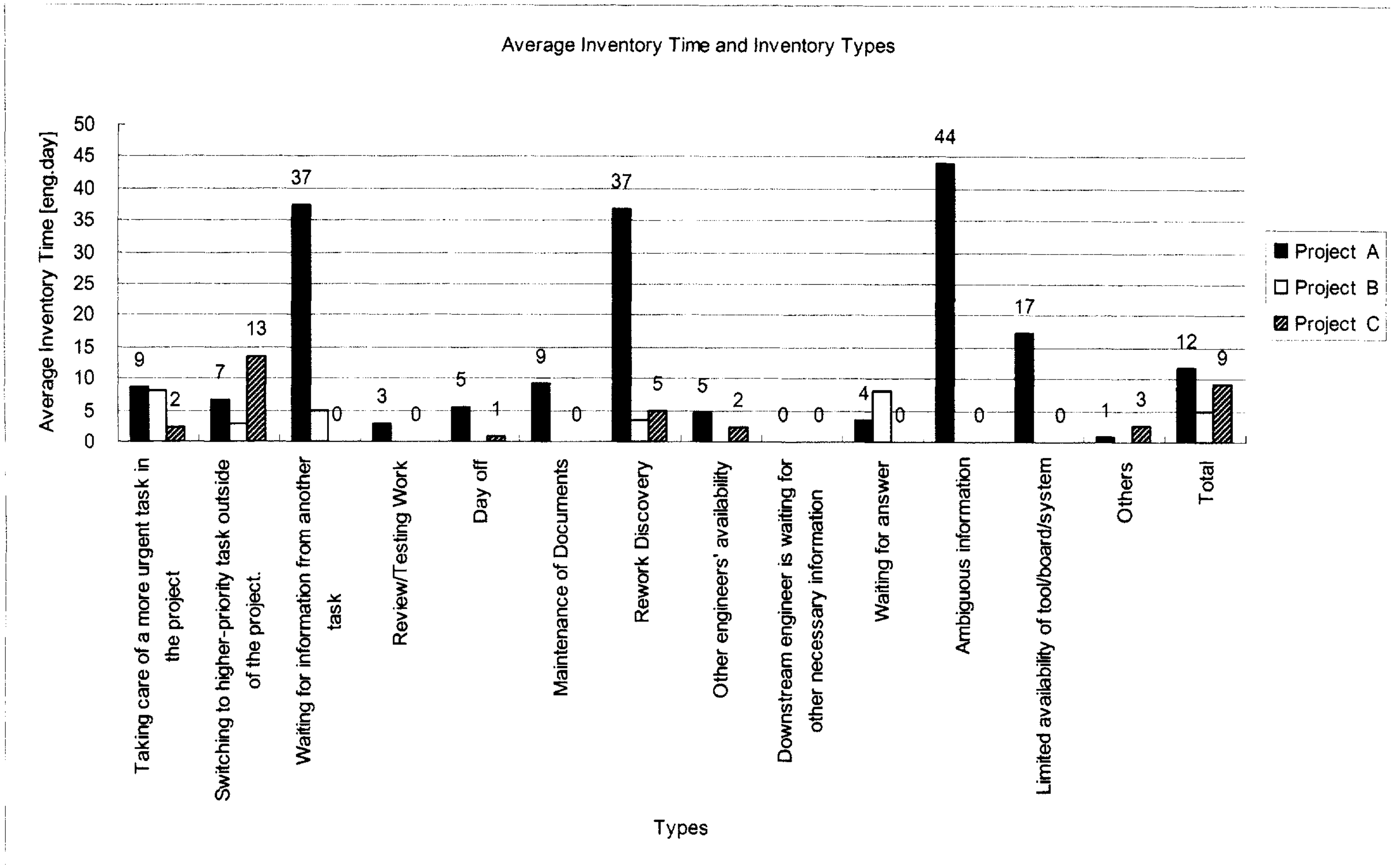


Figure 11.28 Average Inventory Periods and the Corresponding Types

11.8.3 TOTAL INVENTORY TIME

The overall inventory time by types is shown in figure 11.29. The two most significant types of inventory in Project A were "Taking care of a more urgent task in the period (Type 1)" and "Waiting for information from another task (type 2)." The two types take about 2/3 of the total inventory time of Project A. Project B had the longest inventory time in "Switching to higher-priority task outside of the project" (type 2) of all the three projects.

11.8.4 OVERALL DISCUSSION

Figure 11.30 describes unsynchronized production processes described in "The Goal" (Goldratt, 1984), in which throughput is low although all machines are busy. Goldratt attributes this situation to the production management prevalent in 1980's, in which inventory is not measured. In such unsynchronized production processes, the measurements of utilization levels leads only to local optimization, leading to production of huge amount of work-in-process inventory. In figure 11.30, WIP (a) is waiting because Machine A is busy with other WIP. WIP (c) is waiting because other parts necessary for assembly are missing.

Project A is in a similar situation (figure. 11.31). WIP (a) in figure 11.30 corresponds to information inventory (a), which is type 1 inventory and WIP (c) in figure 11.30, information inventory (c). Therefore, Project A, with high amount of types 1 and 2 inventory, is considered to be as less productive as the manufacturing process in figure 11.30.

Morgan (2002) listed "over utilization" as one of twelve product development process wastes, arguing utilization level over 80% significantly decreases the system's throughput. Project A's unsynchronization implies that the project chronically suffered over utilization.

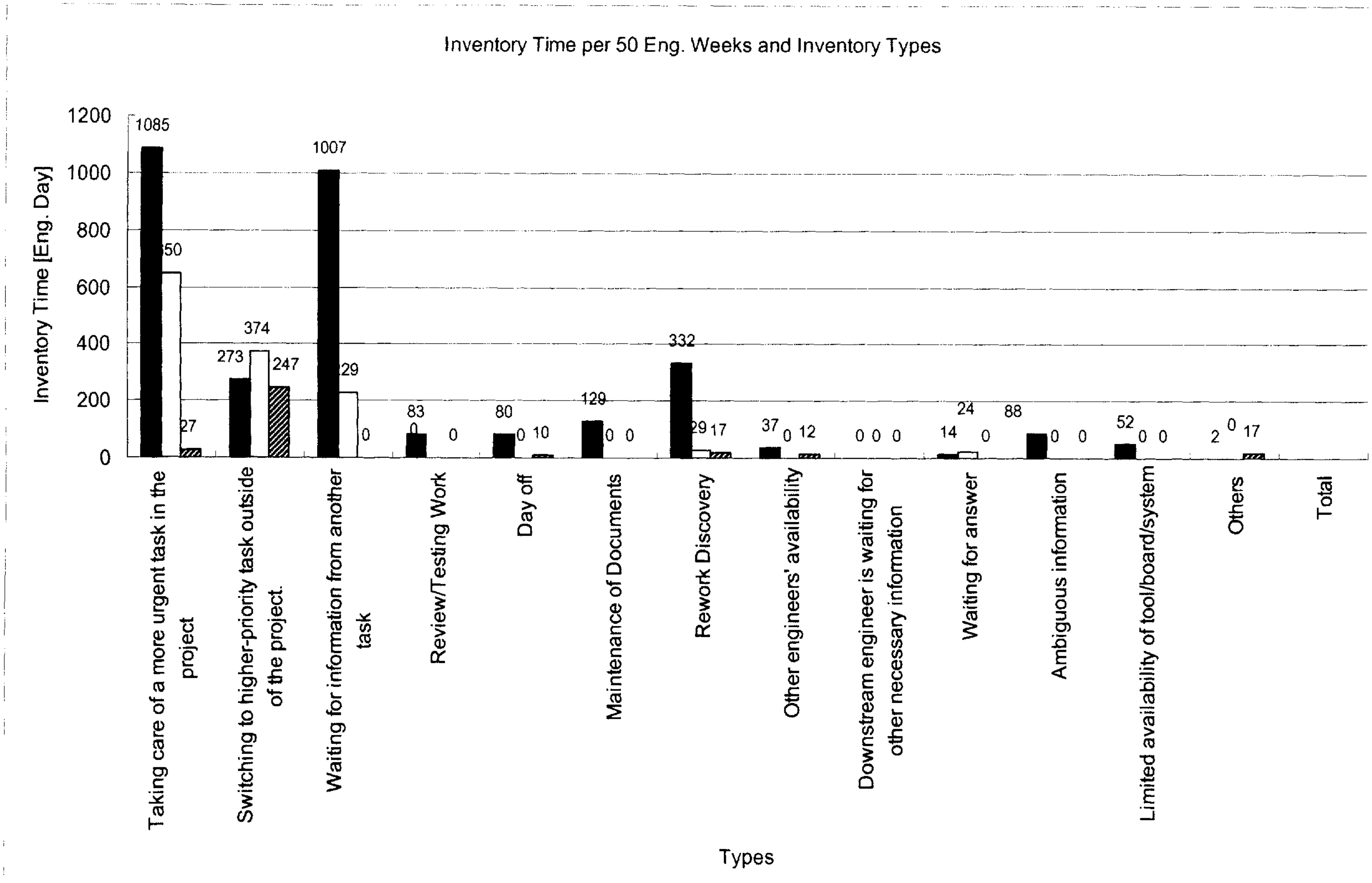


Figure 11.29 Total Inventory Time and the Corresponding Types

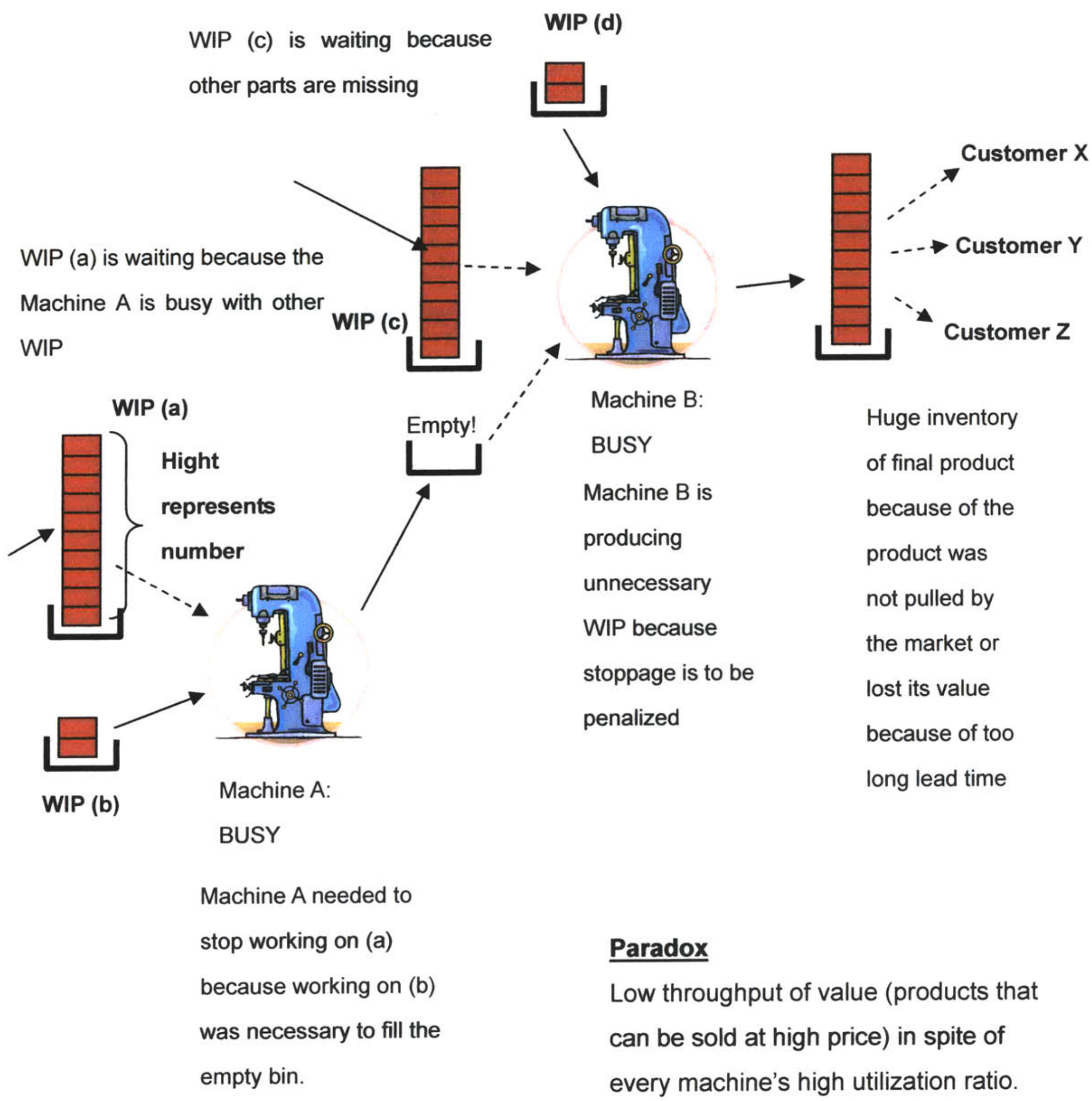


Figure 11.30 Unsynchronized Manufacturing Process Described in "The Goal" (Goldratt, 1984).

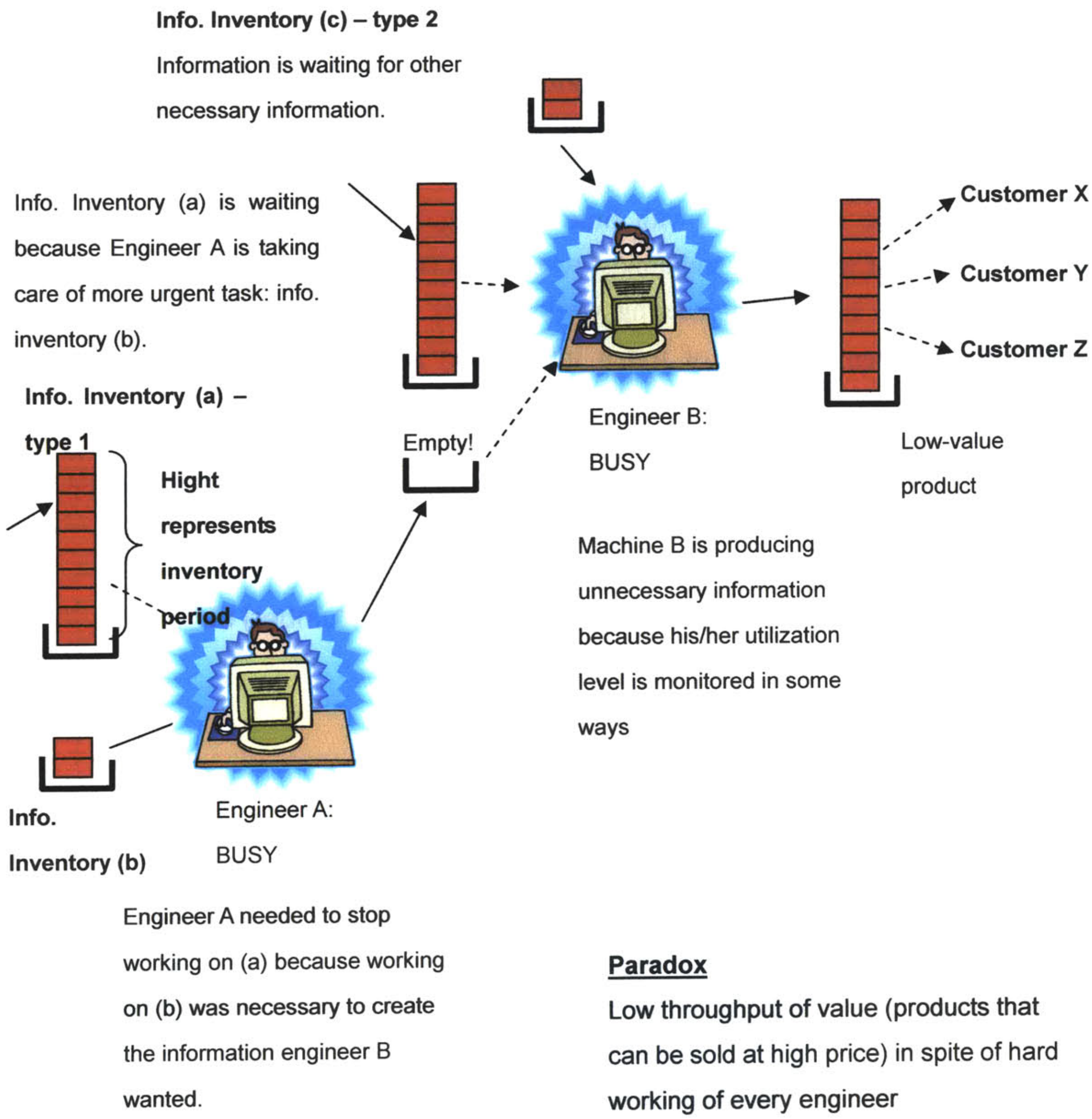


Figure 11.31 Unsynchronized Development Process of Project A

11.9 ROTTEN AND FRESH INVENTORY

11.9.1 RATIO OF ROTTEN INVENTORY AND TIME

Investigation in rotten inventory was performed based on the idea described in chapter 8. Figure 11.32 shows the percentage of rotten information identified in Project A; only 5 % was found to be partially or completely rotten. Figure 11.33 shows changes in ratio of rotten inventory of information with time. This figure reveals that the ratio of rotten inventory increased with time, and almost twenty percent of information got rotten when it was kept for three to four engineering weeks. Figure 11.34 shows the trend line of the relationship between the ratio of rotten information and time. The trend line was the following:

$$y = 0.0054x + 0.8094 \qquad \text{(Equation 11.1)}$$

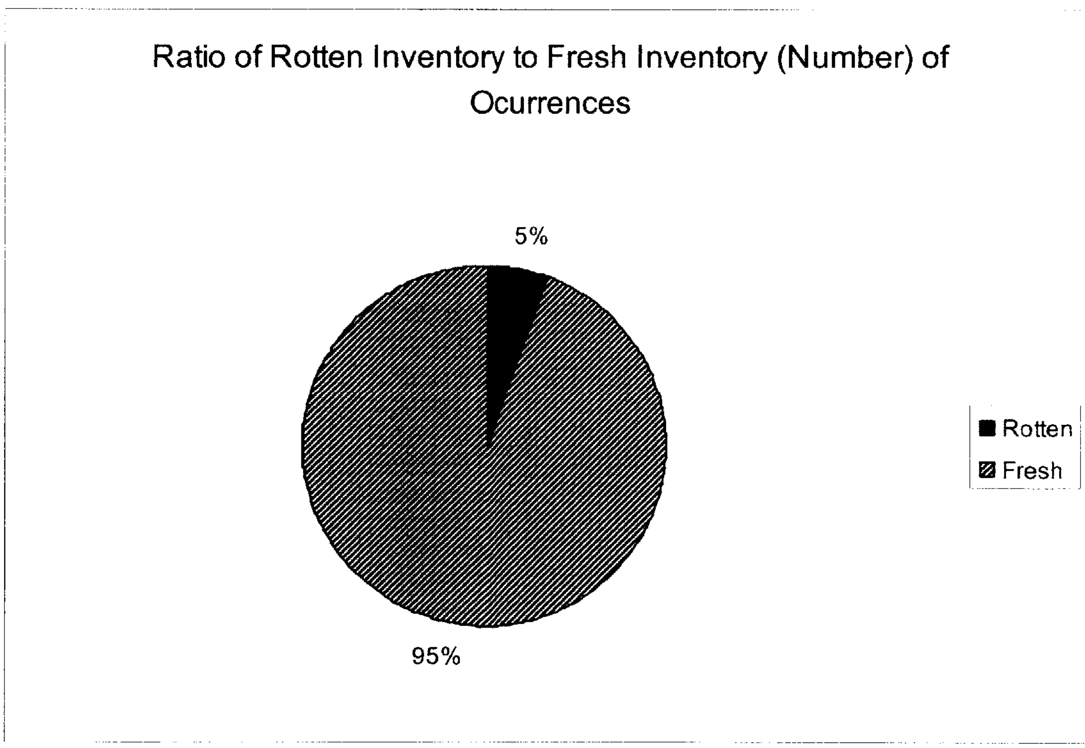


Figure 11.32 Ratio of Rotten Inventory of Information in Number (Project A)

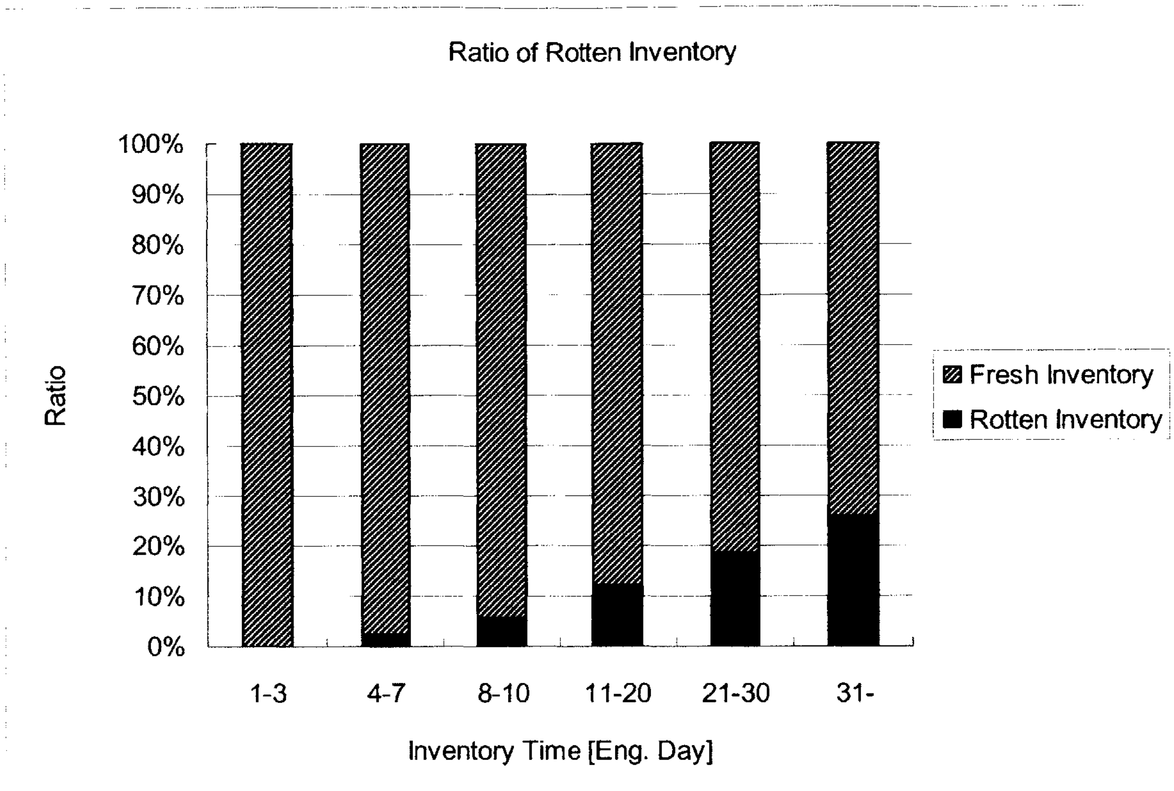


Figure 11.33 Changes in Ratio of Rotten Inventory of Information with Time (Project A)

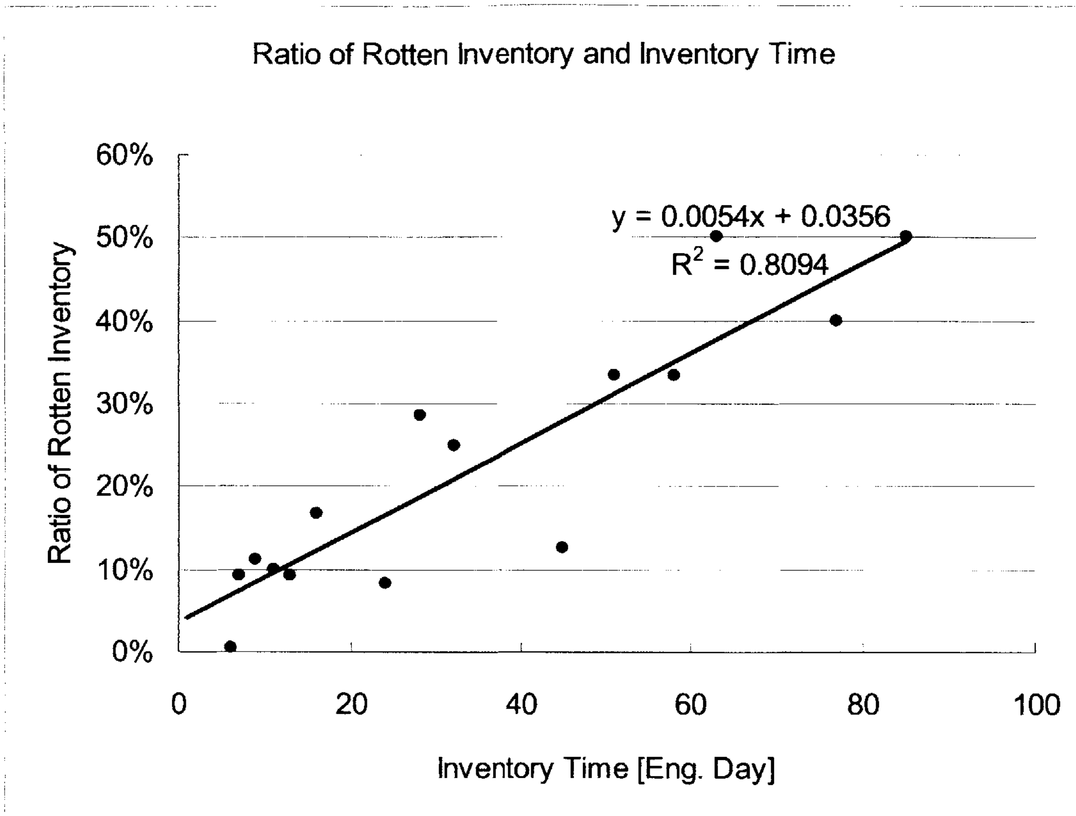


Figure 11.34 Trend Line of Changes in Ratio of Rotten Inventory with Time (Project A)

11.9.2 RATIO OF LOST VALUE IN ROTTEN INFORMATION

Rework Ratio can be calculated with the following equation:

$$\text{Rework Ratio} = (\text{Time Spent on Rework})/(\text{Time Spent on Original Work}) \quad \text{(Equation 11.2)}$$

Figure 11.35 shows the relationship between rework ratio and inventory time of all the rotten information in Project A; there was no strong correlation between them. The average rework ratio was 53%.

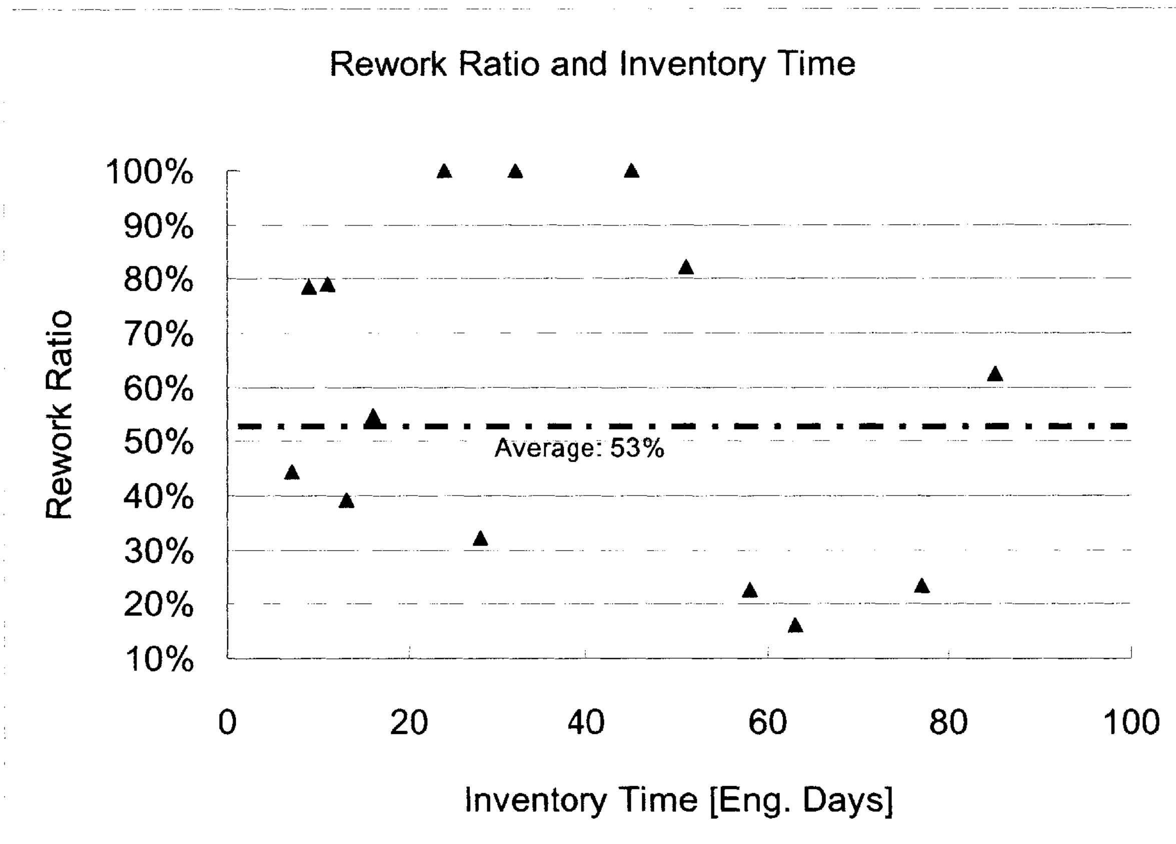


Figure 11.35 Changes in Rework Ratio with Time

11.9.3 MONTHLY INTEREST RATE CALCULATION

Because the rotten inventory increased linearly with time (equation 11.1), and the rework ratio had no correlation with time, inventory of information can be considered to have some interest rate. The monthly interest rate can be calculated with the following equation:

0.0054[/ Eng. Day] * 21[Eng. Day / Eng. Month] * 0.53= 6% [/ Eng. Month] (Equation 11.3)

This implies that if information is kept as inventory for a month, engineers need to work extra 6% on average to make up for the loss.

This interest rate is considered to be useful for re-designing organizational structures: products in the highly unstable market should have processes in which the amounts of inventory of information are minimized.

11.9.4 TYPES OF ROTTEN INVENTORY -- DISTRIBUTION OF TYPES OF ROTTEN INVENTORY

Figure 36 shows the ratios of rotten inventory by their types. 56% of "Rework discovery" (type 7) were rotten; 44% of rework discovery occurred due to hidden errors in the information itself. Rotten information was also identified in waiting "Waiting for information from another task" (type 3), "Day off" (type5), "Switching to higher-priority task outside of the project" (type 2), and "Taking care of a more urgent task in the project" (type 1).

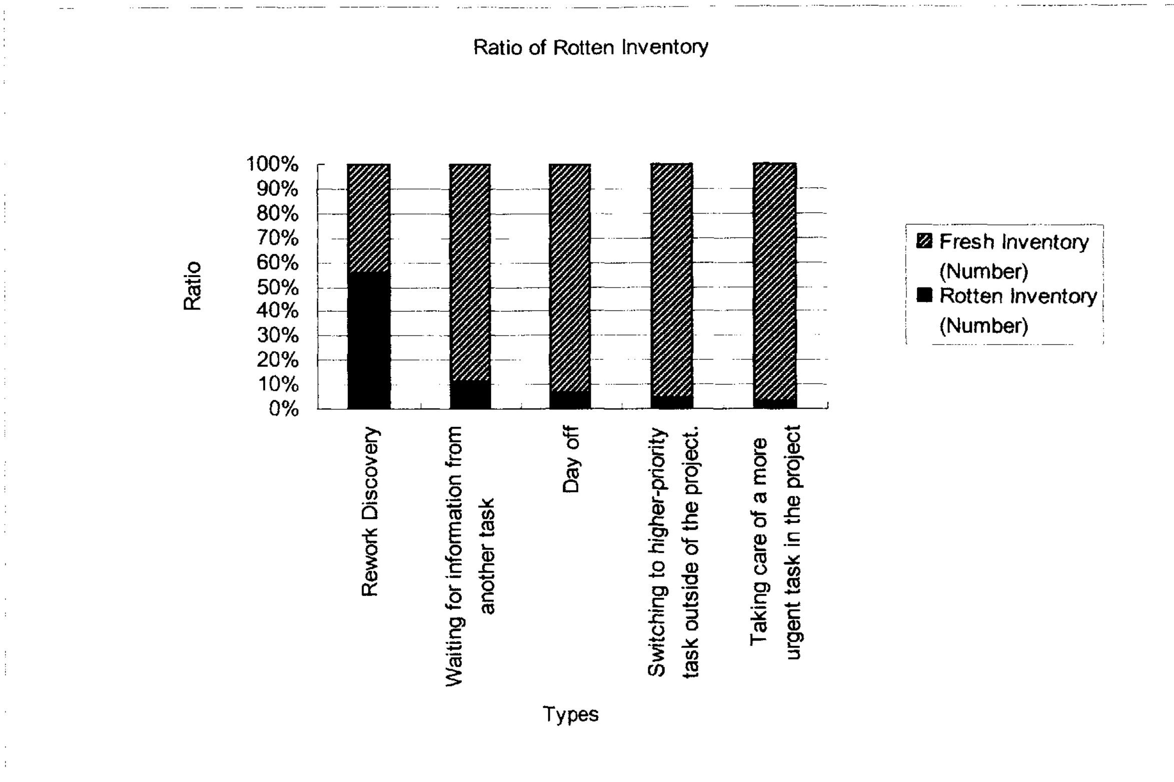


Figure 11.36 Relationships between Ratio of Rotten Inventory and the Corresponding Types

11.10 SUMMARY OF THIS CHAPTER

Value stream mapping was also applicable for quantitative measurements of information inventory. Quantitative measurements revealed how much and why engineers' activities are interrupted – this result can help companies revise their future strategies in project management, scheduling, prioritization, and resources allocation. The analysis suggested that Project A' information flows resembled those typical in manufacturing factory several decades ago.

Rotten information was identified and measured in Project A. On average, 6 % of value adding effort became waste if the output information had been stored for a month in project A.

CHAPTER 12 FUTURE WORK

12.1. NINE WASTE INDICATORS

The three waste indicators, over processing, rework, and defective information were more prevalent of all nine waste indicators in the three projects investigated in this research. Still, the results of three investigations are not sufficient to justify ignoring the waste indicators that were not significant in them. Performing more case studies, especially fields other than development processes of embedded software, will be effective for reduction of waste indicators.

12.2 INVENTORY OF INFORMATION

Inventory of information was found to be prevalent in all the three projects, and the study in one of the projects revealed that information got rotten rapidly. Although a specific interest rate was deduced, this interest rate is expected to depend on contexts specific to projects.

Another topic related to the interest rate of information inventory is the exploring the relationship between the interest rate of information inventory (X in figure 12.1) and the reduction of released product's value (Y in figure 12.2). X's causes include market, requirement, and technical risk and Y's cause is market risk. Therefore, X and Y should be correlated and X should be more than Y. Deducing X needs drawing a value stream map; Y can be deduced from sales information, which takes less effort. If the correlation between X and Y becomes known, X, which is useful for re-designing organizations, can be deduced without detailed analysis requiring value stream mapping.

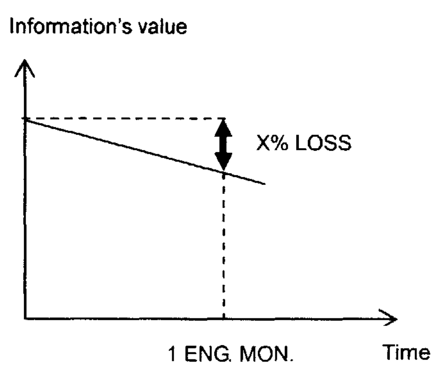


Figure 12.1 Interest Rate of Information Inventory

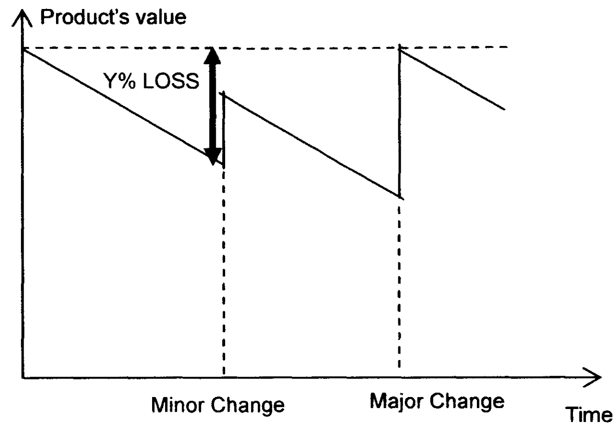


Figure 12.2 Reduction of Released Product's Value

12.3 SUBSTITUTING TASK INVENTORY FOR INFORMATION

Task inventory is idle tasks that are ready to be worked on by engineers. Task inventory can be counted by engineers. It can also be counted with a value stream map; in figure 12.3, inventoried tasks of this engineer on 2/19 are six. Counting task inventory is easier than measuring inventory time, which is performed in this research. However, unlike in manufacturing, time needed for a task varies significantly. And, engineers tend to start working on the easiest task. Therefore, numbers of inventoried tasks may not have the same meaning throughout a project. However, investigation of the correlation between information inventory and task inventory may verify the possibility of replacing information inventory with task inventory.

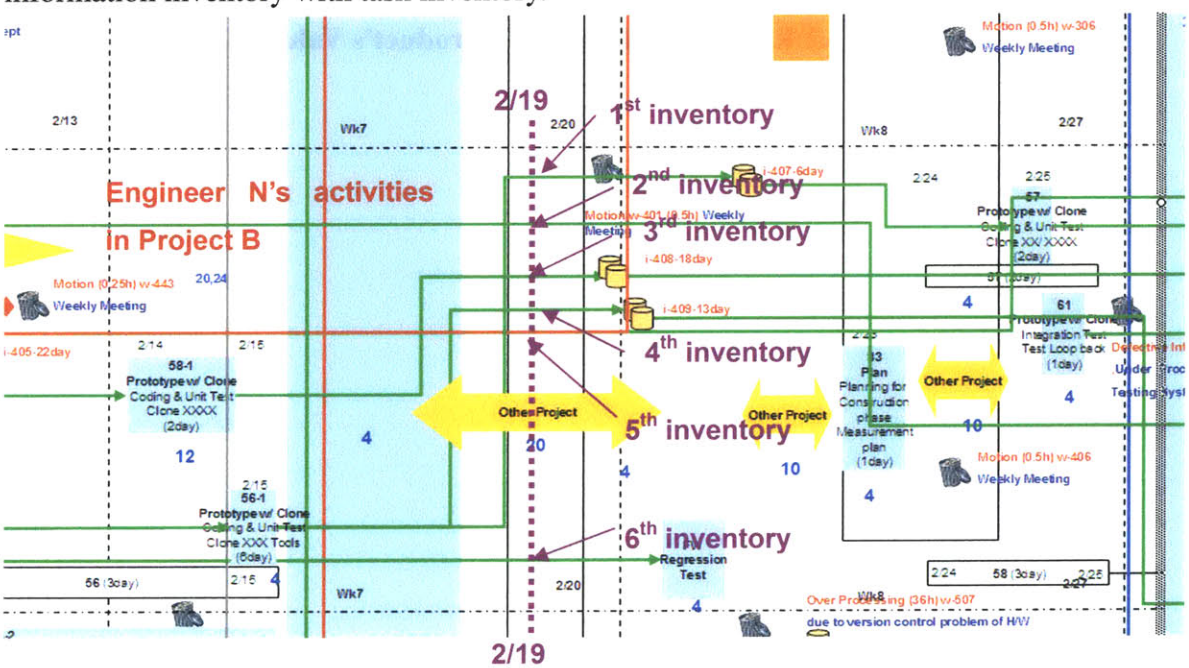


Figure 12.3 Counting Task Inventory with a Value Stream Map (Project B) – Task Inventory Can be Calculated by Counting Green or Red Lines Crossing a Day.

12.4 ARE TOC'S THREE METRICS SUFFICIENT?

Analysis of Project A (see chapter 11) revealed that the project's development process is similar to the manufacturing process described in "The Goal" (Goldratt, 1984). This implies that the three metrics (throughput, inventory, and operating expense) of Theory of Constraints (TOC) (Goldratt) may be sufficient for addressing waste in Today's product development processes. Verification of this idea needs the following processes.

1. Defining "throughput" in product development processes.
2. Test the three metrics for measuring waste in product development processes.
3. Examine the three metrics address most of waste in product development processes.

12.5 SHOWING WASTED TIME EXPLICITLY IN VALUE STREAM MAPS

In chapter 7, wasted time was explicitly displayed in value stream maps (figure 12.4). Although this technique was not applied in drawing value stream maps in this research, it may make wasted time more distinct from value-adding time.

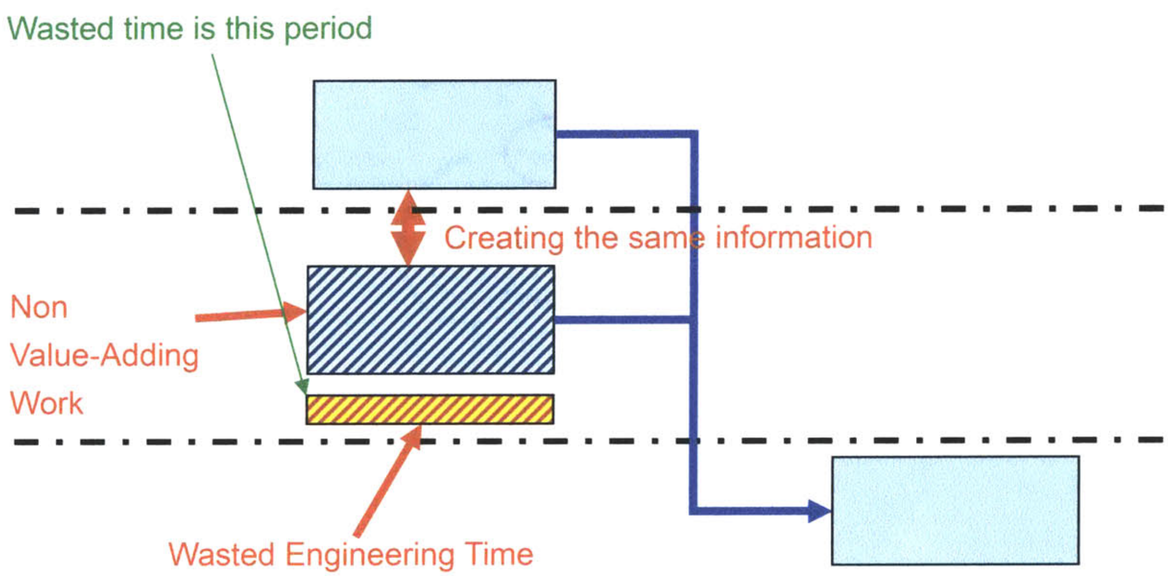


Figure 12.4 Measuring Time Spent on Overproduction – A Hatched Process Box Mean Overproduction (Same as Figure 7.1)

12.6 EXPLORATION OF ENGINEER UTILIZATION LEVELS

The value stream maps in this research have all the information necessary for measuring engineer utilization levels. Measuring utilization levels, which was not performed in this research, may make it clear how utilization levels affect product development processes.

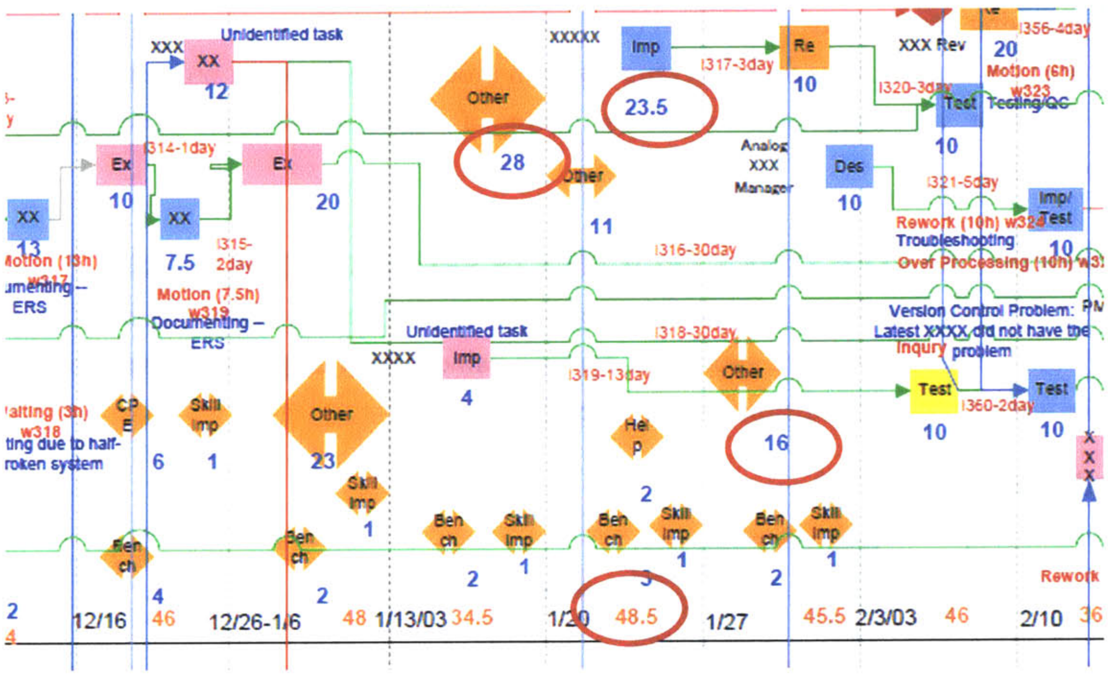


Figure 12.5 Time Spent on Tasks Shown in a Value Stream Map (Project A, Engineer Y)

CHAPTER 13 CONCLUSIONS

Conclusions from the Perspective of This Research's Goal

- The nine plus one waste indicators for measurement of waste were determined based on the analyses of causal relationships among various factors of low performances in product development processes.
- As a result of these analyses, the root-cause analysis diagram was produced. The diagram was a database of causal relationships among factors of low performances.
- The value stream mapping for quantitative measurement of waste was developed through preliminary case studies. A process for measuring waste with waste indicators and value stream mapping was developed.
- These tools and processes have successfully been applied to three industrial case studies in two companies, in which value stream maps have been developed through intensive interviews with project managers and engineers.
- The three case studies has proved the following:
  - The nine waste indicators were sufficient for identifying and measuring waste in product development processes.
  - Inventory of information was prevalent in product development processes.
  - The root-cause analysis diagram was useful for identifying typical root-causes for waste.
- The lean tools and processes developed in this research have proved to be able to identify problems both peculiar and common to the organizations.
- Therefore, these lean tools and processes can deliver to product development organizations information that leads to continuous improvement of their value-creating processes.

Findings and Implications

- Among the nine waste indicators, three waste indicators, over processing, rework, and defective information were more significant than the others, implying the possibility of reducing the number of waste indicators.

- It has been proved that time per one occurrence of rework exponentially increases as time spent on the project increases.
- Analysis of inventory of information has revealed that the development processes of the investigated projects has turned out more or less similar to the unsynchronized manufacturing processes several decades ago.
- In one of the investigated projects, information got rotten at the rate of 6% a month. This indicates information inventoried for a month causes additional engineering work by 6%.

APPENDIX I EXECUTIVE SUMMARY

1. GOALS AND OBJECTIVES

The goal of this research is to develop a process for continuous creation of lean value in product development organizations. Creation of lean value means realizing value with minimum wasteful process.

To achieve this goal, the objective of this research was determined to develop ideas and methodologies of lean product development into tools and processes that can help product development organizations (1) identify and measure the waste in their teams' processes; (2) identify causes and measure their impacts on PD processes; and (3) finally learn the best strategies to pursue to improve their PD processes.

2. RESEARCH PROCESS

2.1 Define nine plus one waste indicators for waste measurement (table A-1).

Table A-1 Nine plus One Waste Indicators

	Waste Indicator	Description
Nine Waste Indicators	1. Overproduction of Information (Duplication)	Different people/groups are unintentionally creating the same information.
	2. Waiting of People	People are waiting.
	3. Transportation of Information (Preparing and forwarding information)	Information is in transportation.
	4. Over Processing	Engineers create information that won't contribute the value of product.
	5. Motion of People (Information hunting, travel, reviews, documentation, and meetings)	People have to spend time on non value-adding motions.
	6. Rework	Redoing tasks perceived to be finished for some reason
	7. Re-Invention	Designing similar things without utilizing past experience.
	8. Hand-Off (Hand-off inside of project)	Information is handed off with its responsibility between two groups/people.
	9. Defective Information (Coupled to Over Processing and Rework)	Erroneous or incomplete information.
Inventory of Information		Work-in-process inventory of information.

2.2 Create the root-cause analysis diagram that is useful for quickly identify root-causes for waste (figure A-1).

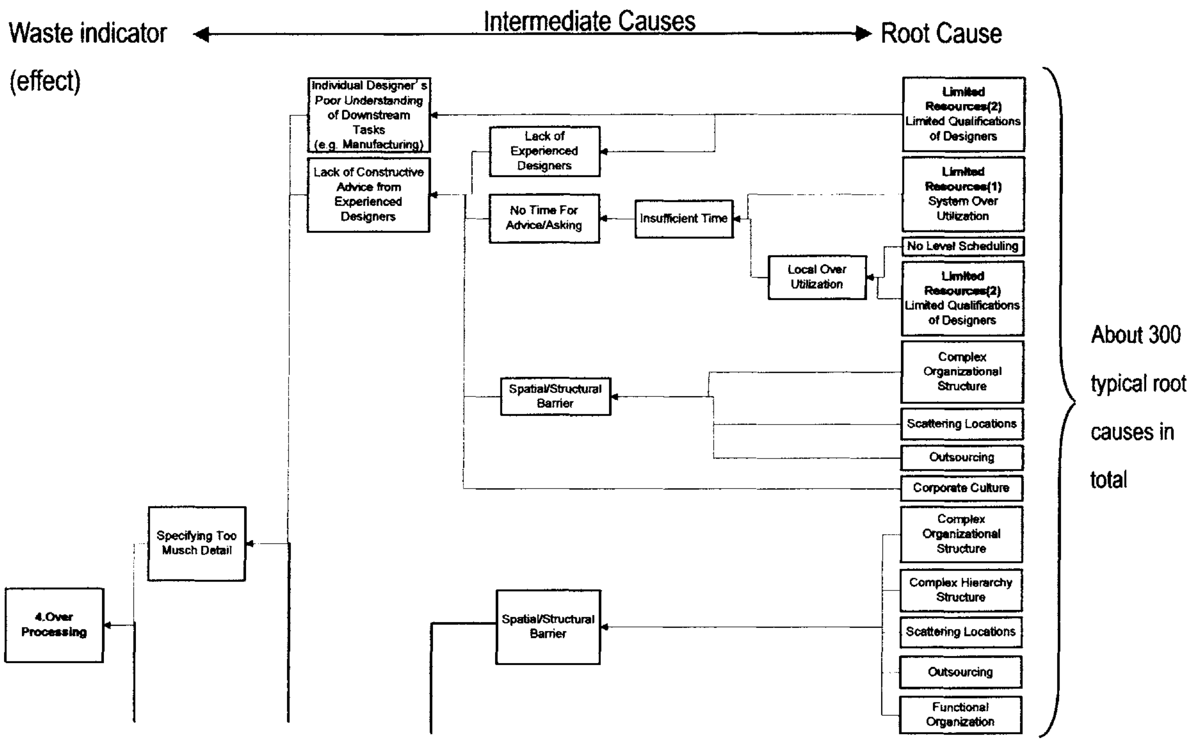


Figure A-1 An Example of Root-Cause Analysis Chart

2.3 Optimize value stream mapping for measuring waste identified by the nine plus one waste indicators (figure A-2).

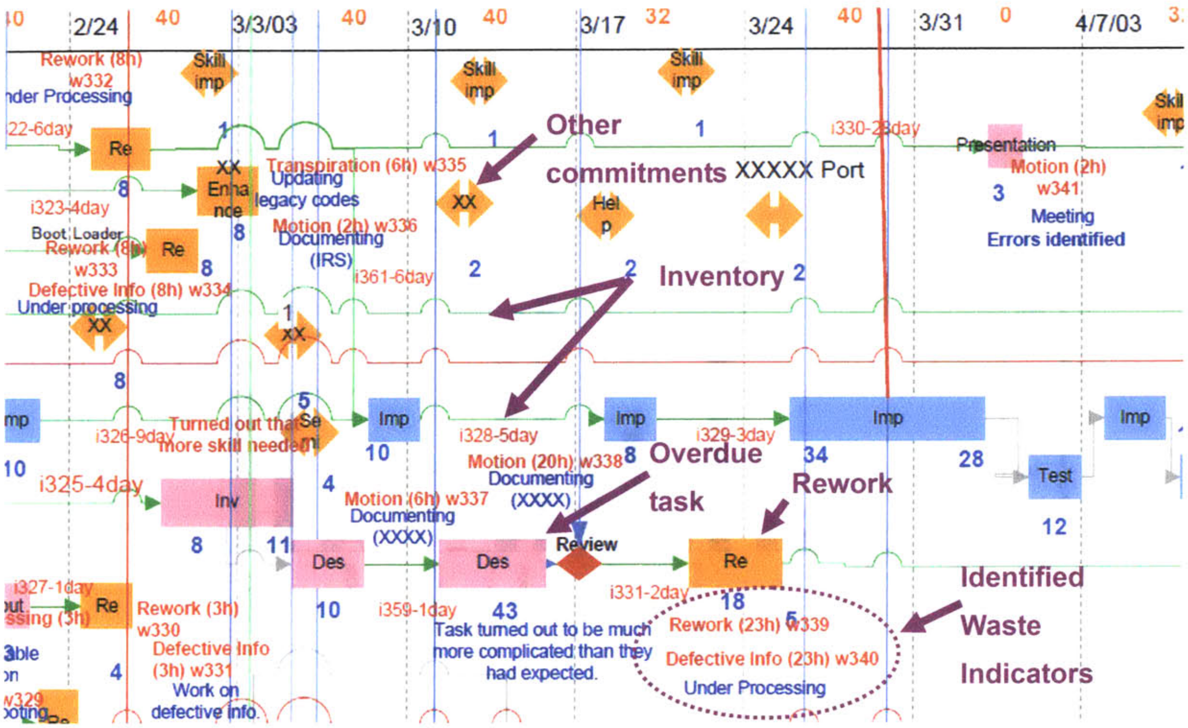


Figure A-2 An Example of the Value Stream Map for Quantitative Analysis

3. RESULTS1: NINE WASTE INDICATORS

The value stream mapping improved in this research made it possible to measure waste time on each category of waste (figure A-3). These results revealed that "Over Processing," "Rework", and "Defective Information" are more prevalent that the other waste indicators.

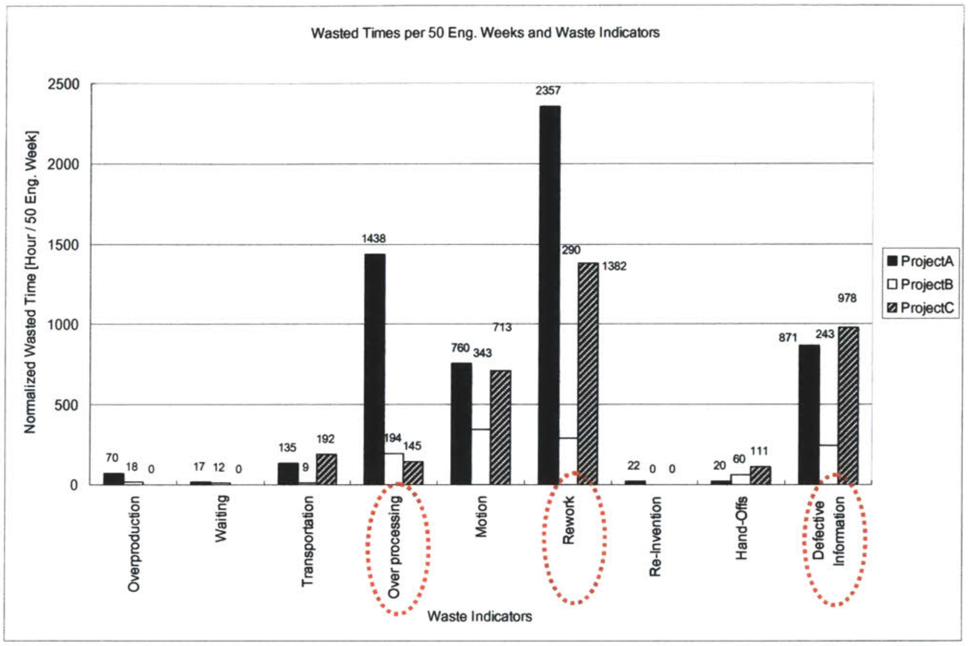


Figure A-3 Relationship between Wasted Time and Nine Waste Indicators

Detailed analyses using the root-cause analysis diagram made it possible to identify problems specific to each company and show how many hours are wasted on them. (Figure A-4).

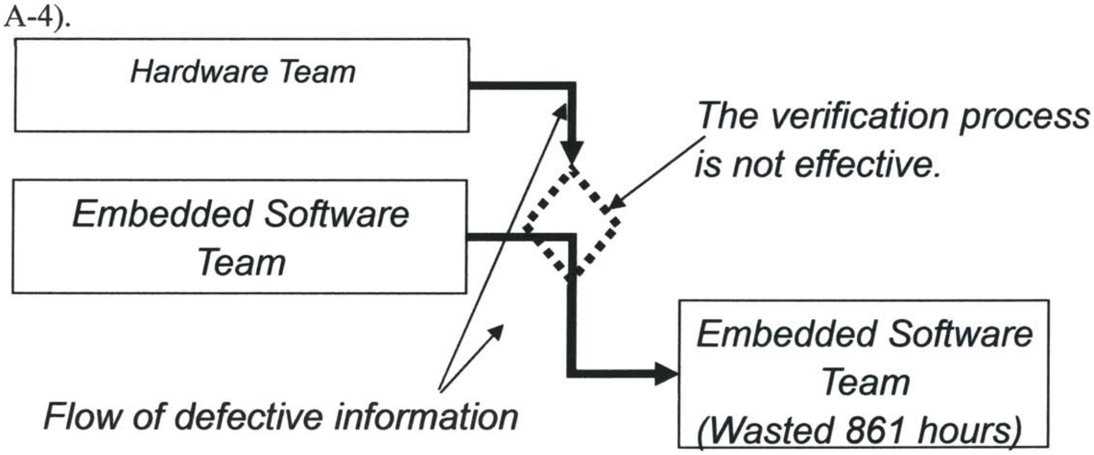


Figure A-4 An Example of Identified Problem

One occurrence of rework turned out to take more time near the end of projects than at the beginning of it (figure. A-5).

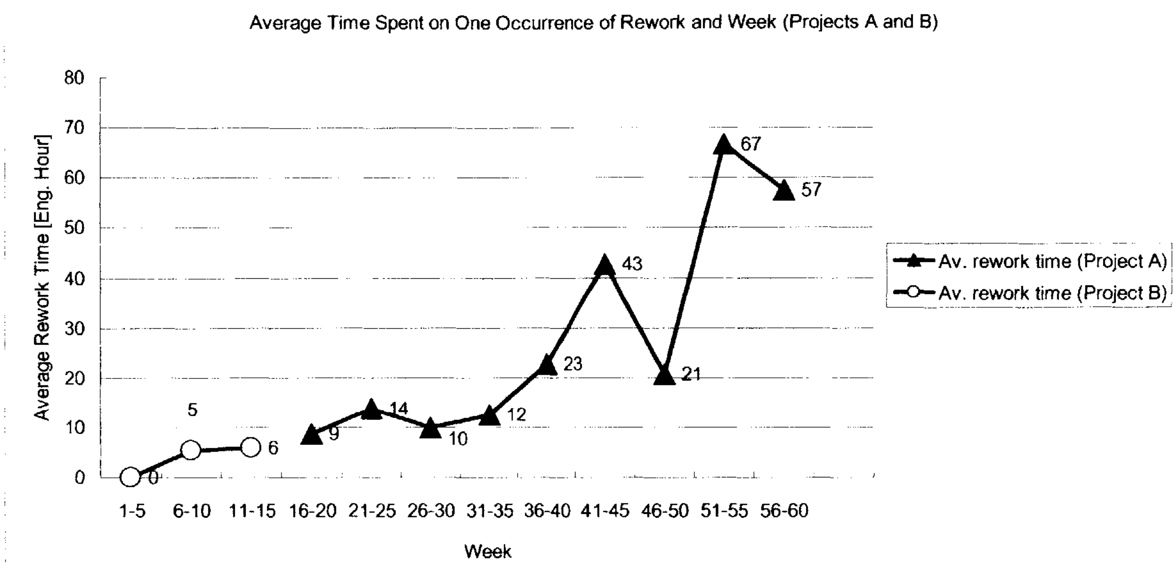


Figure A-5 Average Time Spent on One Occurrence of Rework

4. RESULTS2: INVENTORY OF INFORMATION

4.1 TOTAL INVENTORY TIME

Figure A-6 shows the total inventory Especially, Project A's inventory time was significant: 64 engineering – day inventory time in a engineering week on average (it had 6 engineers).

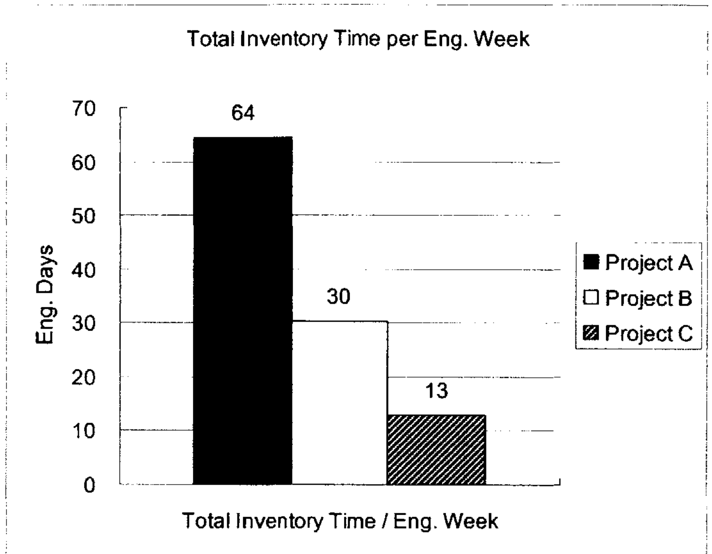


Figure A-6 Total Inventory Time per Engineering Week: Number of Engineers in Projects A, B, and C are 6, 6, and 5 Respectively.

4.2 INVENTORY TIME AND THE CORRESPONDING TYPES

Figure A-7 shows the relationship between inventory time and the types of inventory of information. As can be understood from this graph, the following two types were dominant in Project A.

Type 1: Taking care of a more urgent task in the project

Type 2: Waiting for information from another task

This result implies that Project A's development process was an unsynchronized one: In Project A, although engineers were switching tasks frequently not to delay the project, many tasks were not able to be started because some information was missing. This tendency resembles the manufacturing processes several decades ago.

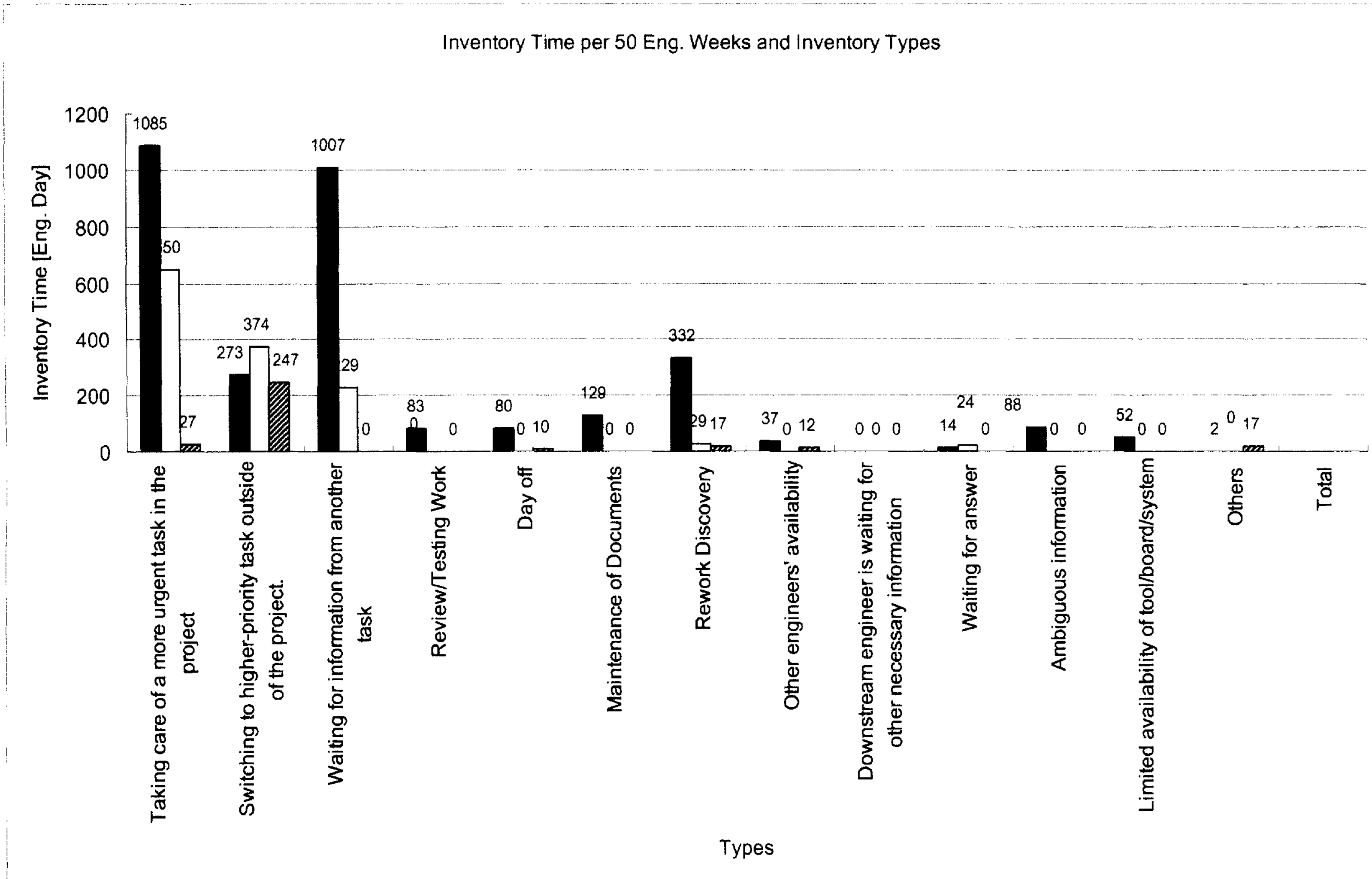


Figure A-7 Total Inventory Time and the Corresponding Type

4.2 ROTTEN INVENTORY

Figure A-8 shows how the number of rotten inventory increases with time. Figure A-9 shows the relationship between rework ratio and time. Rework ratio is defined by the following equation:

Rework Ratio = (Time Spent on Rework) / (Time Spent on Original Work)

Monthly interest rate of information inventory can be calculated as follows:

0.0054[/ Eng. Day] * 21[Eng. Day / Eng. Month] * 0.53= **6% [/ Eng. Month]**

If information is kept as inventory for a month, engineers need to work extra 6% on average to make up for the loss.

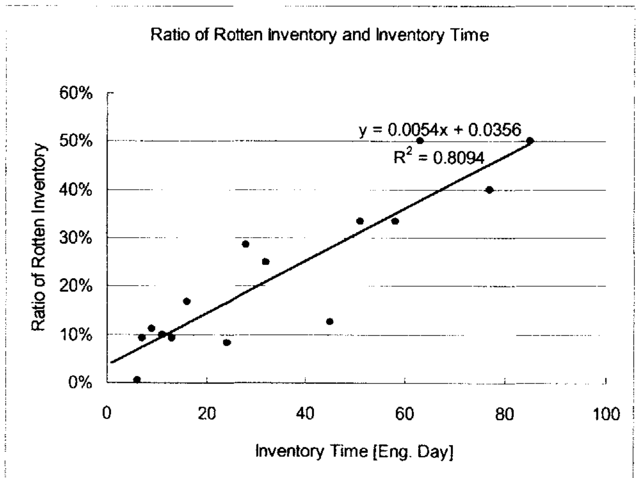


Figure A-8 Trend Line of Changes in Ratio of Rotten Inventory with Time (Project A)

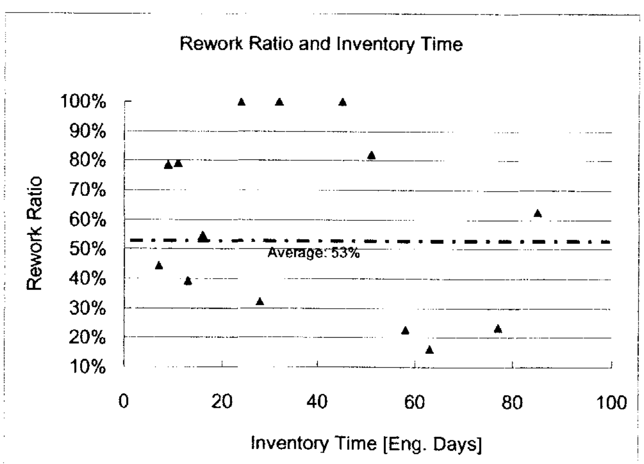


Figure A-9 Changes in Rework Ratio with Time (Project A)

5. CONCLUSION

Overall

- The lean tools and processes developed in this research have proved to be able to identify problems both peculiar and common to the organizations.
- Therefore, these lean tools and processes can deliver to product development organizations information that leads to continuous improvement of their value-creating processes.

Nine Waste Indicators

- The nine waste indicators were sufficient for identifying and measuring waste in product development processes.
- Among the nine waste indicators, three waste indicators, over processing, rework, and defective information were more significant than the others, implying the possibility of reducing the number of waste indicators.

Rework

- It has been shown that time per one occurrence of rework exponentially increases as time spent on the project increases.

Inventory of Information

- Inventory of information was prevalent in product development processes.
- Analysis of inventory of information has revealed that the development processes of the investigated projects have turned out more or less similar to the unsynchronized manufacturing processes several decades ago.
- In one of the investigated projects, information got rotten at the rate of 6% a month. This indicates information inventoried for a month causes additional engineering work by 6%.

Root-Cause Analysis Diagram

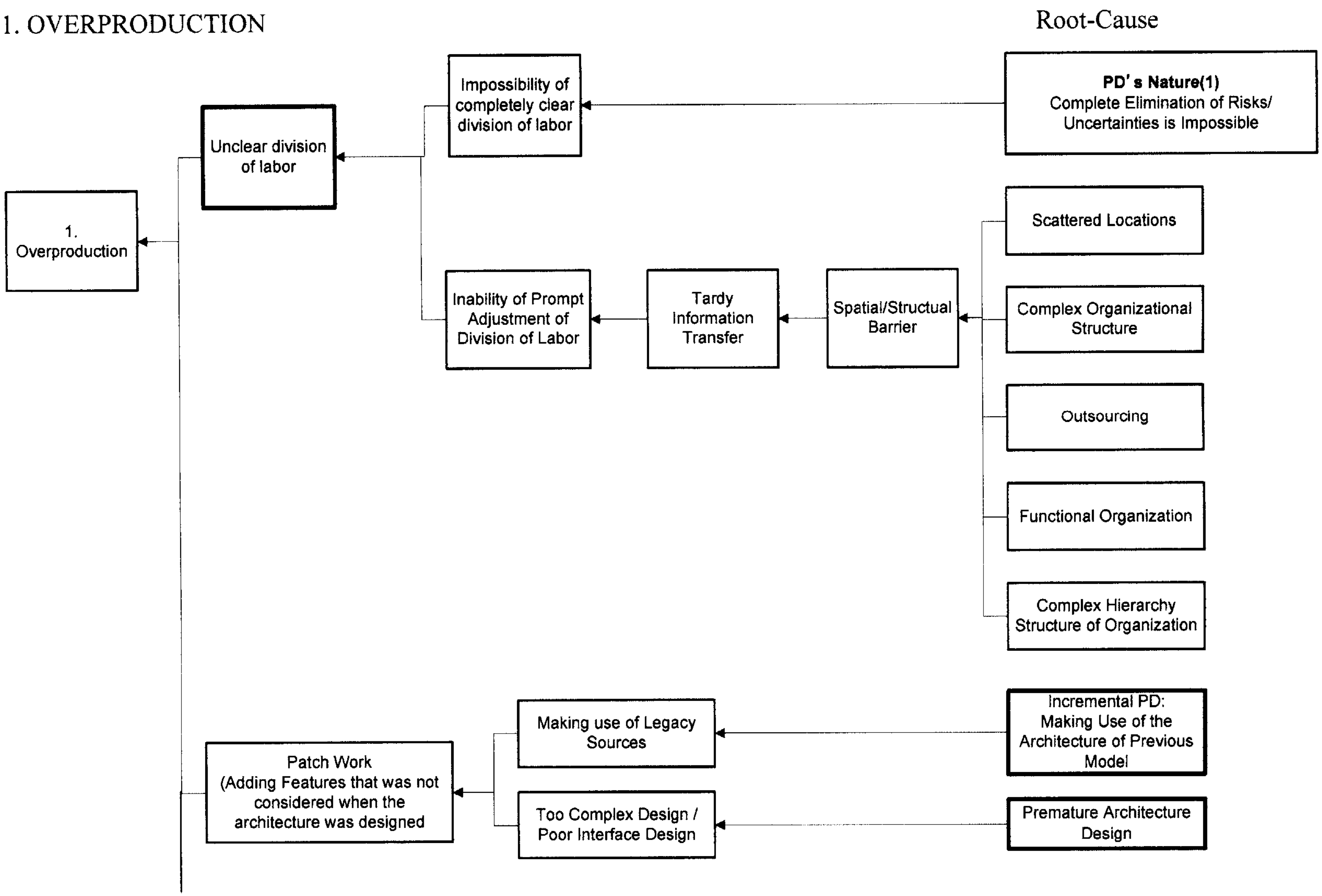
- The root-cause analysis diagram was useful for identifying typical root-causes for waste.

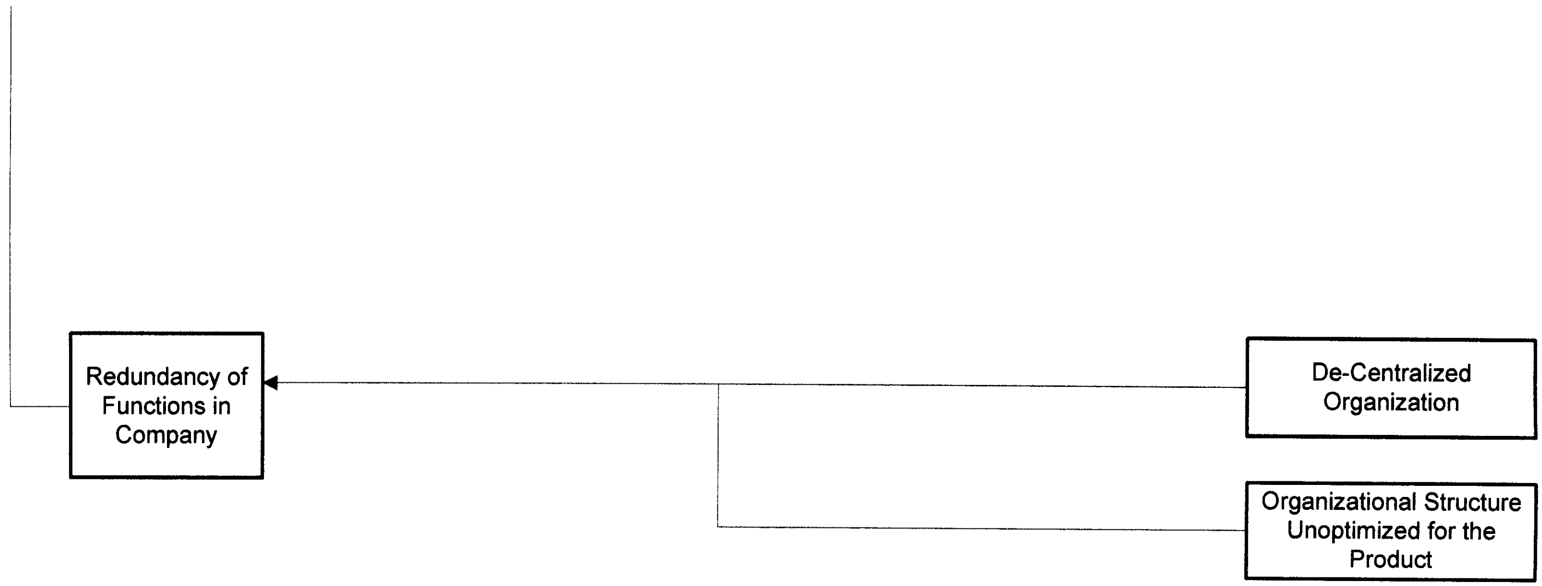
Value Stream Mapping for Quantitative Analysis

- Value Stream Mapping Optimized for Quantitative Analysis was applicable for measuring waste using the waste indicators.

APPENDIX II ROOT-CAUSE ANALYSIS DIAGRAM

1. OVERPRODUCTION

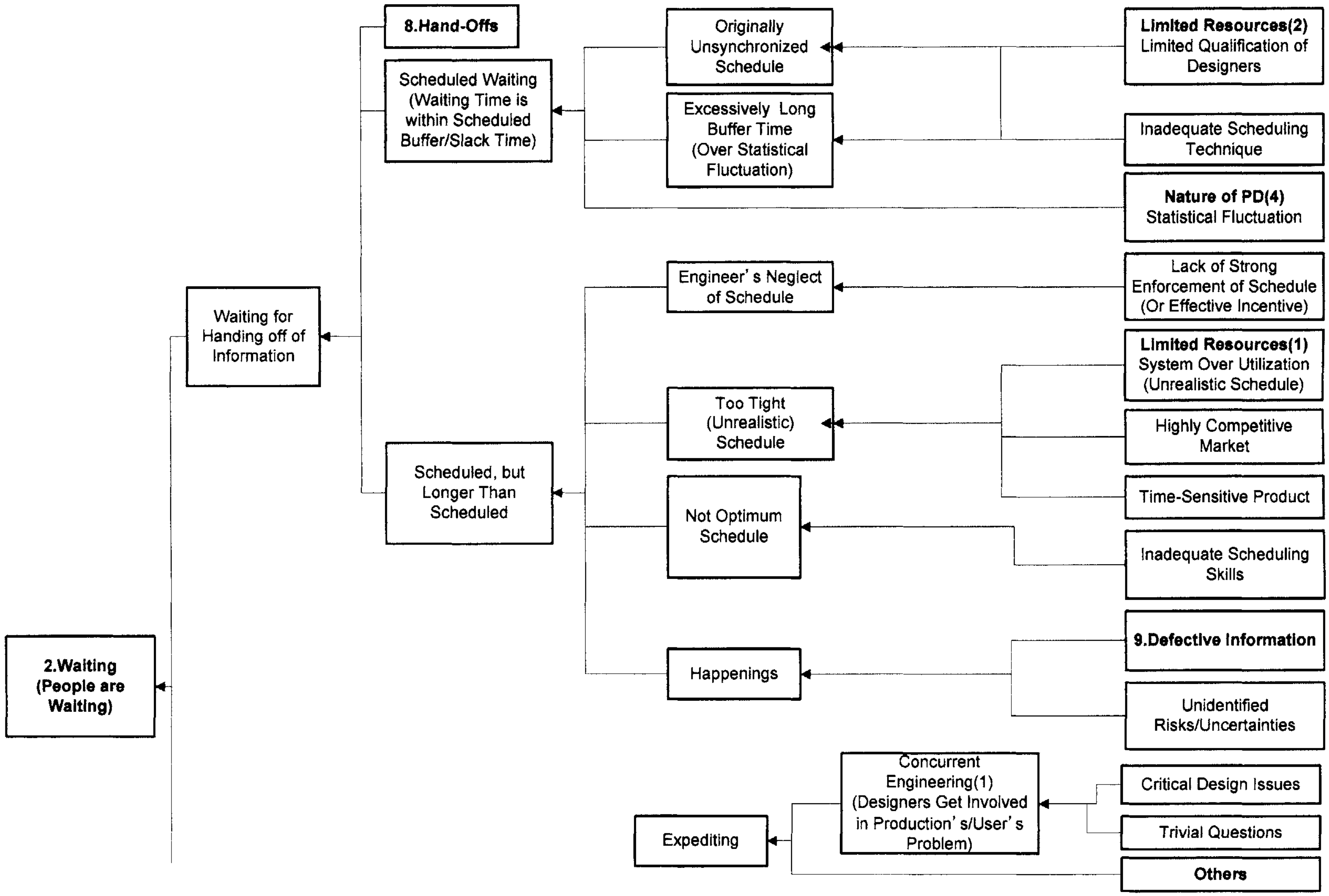


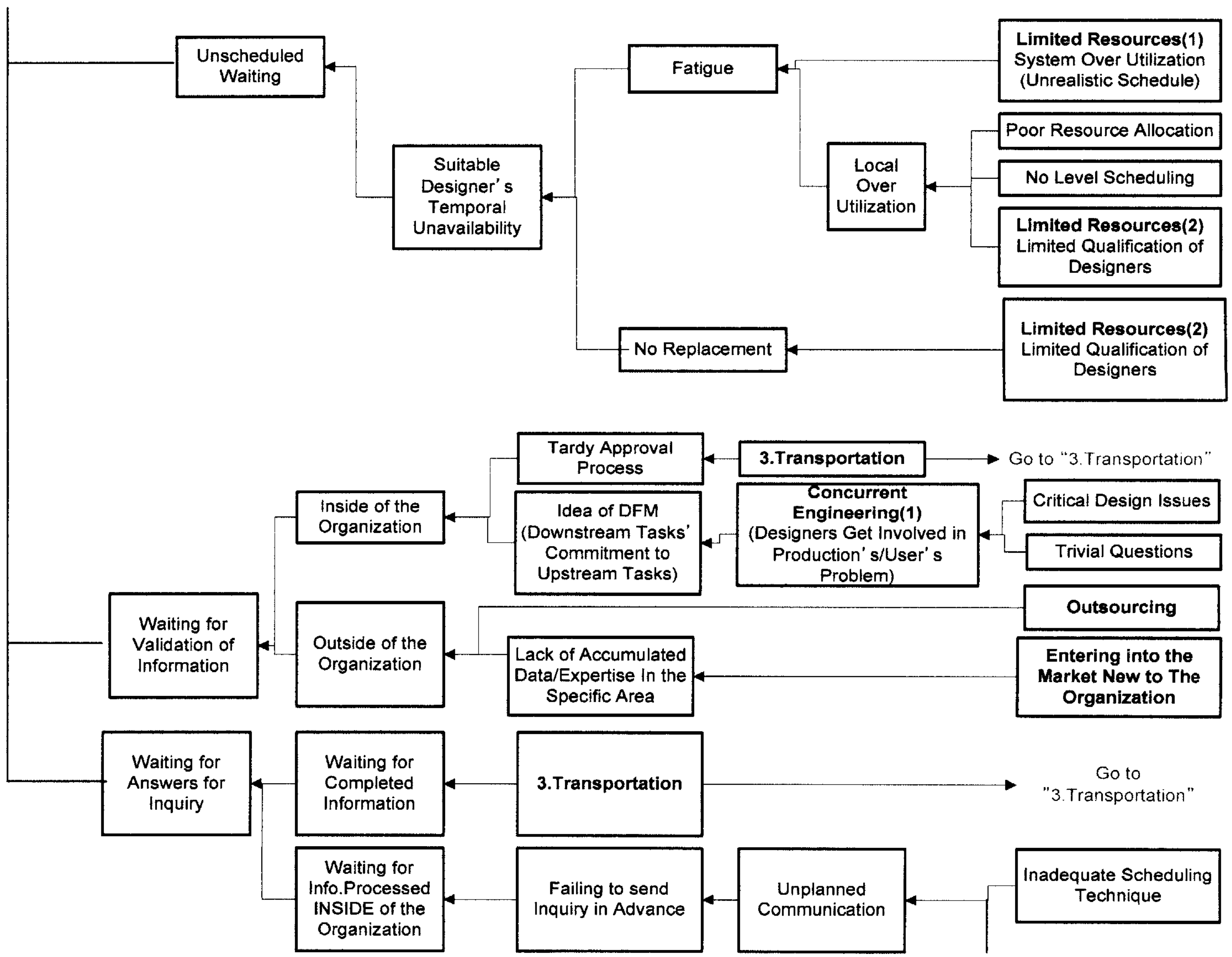


Redundancy of Functions in Company
De-Centralized Organization
Organizational Structure Unoptimized for the Product

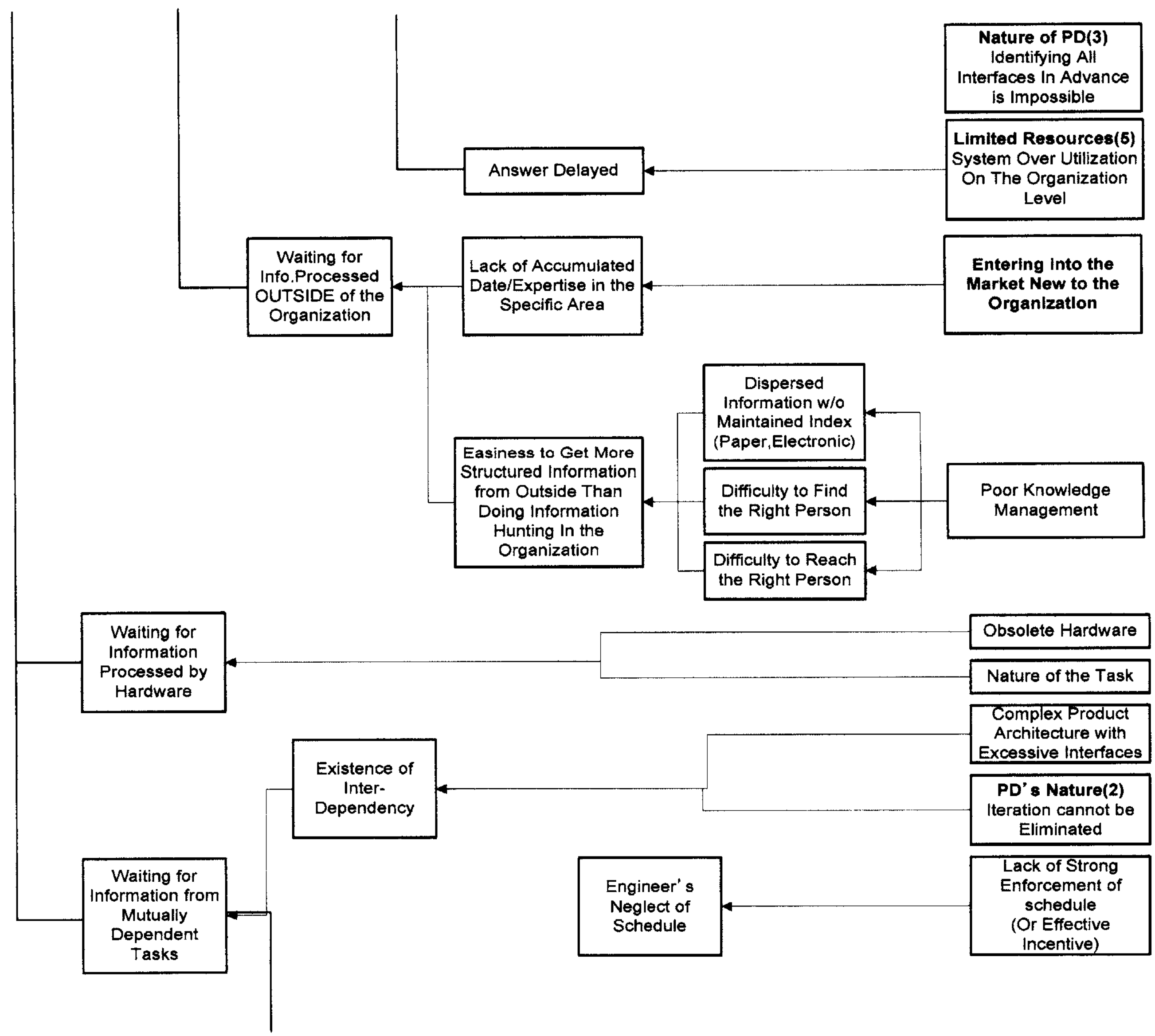
2. WAITING

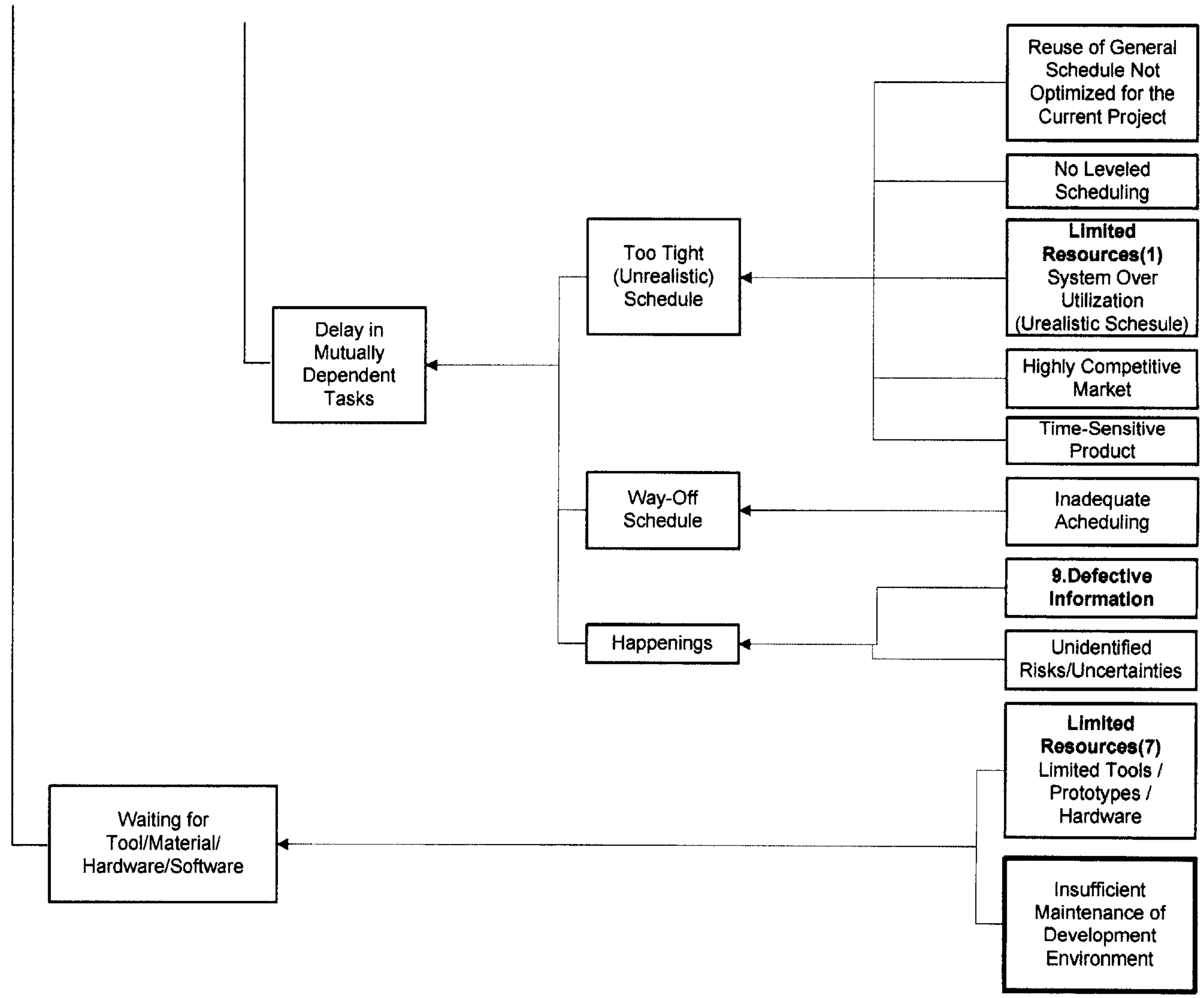
Root-Cause





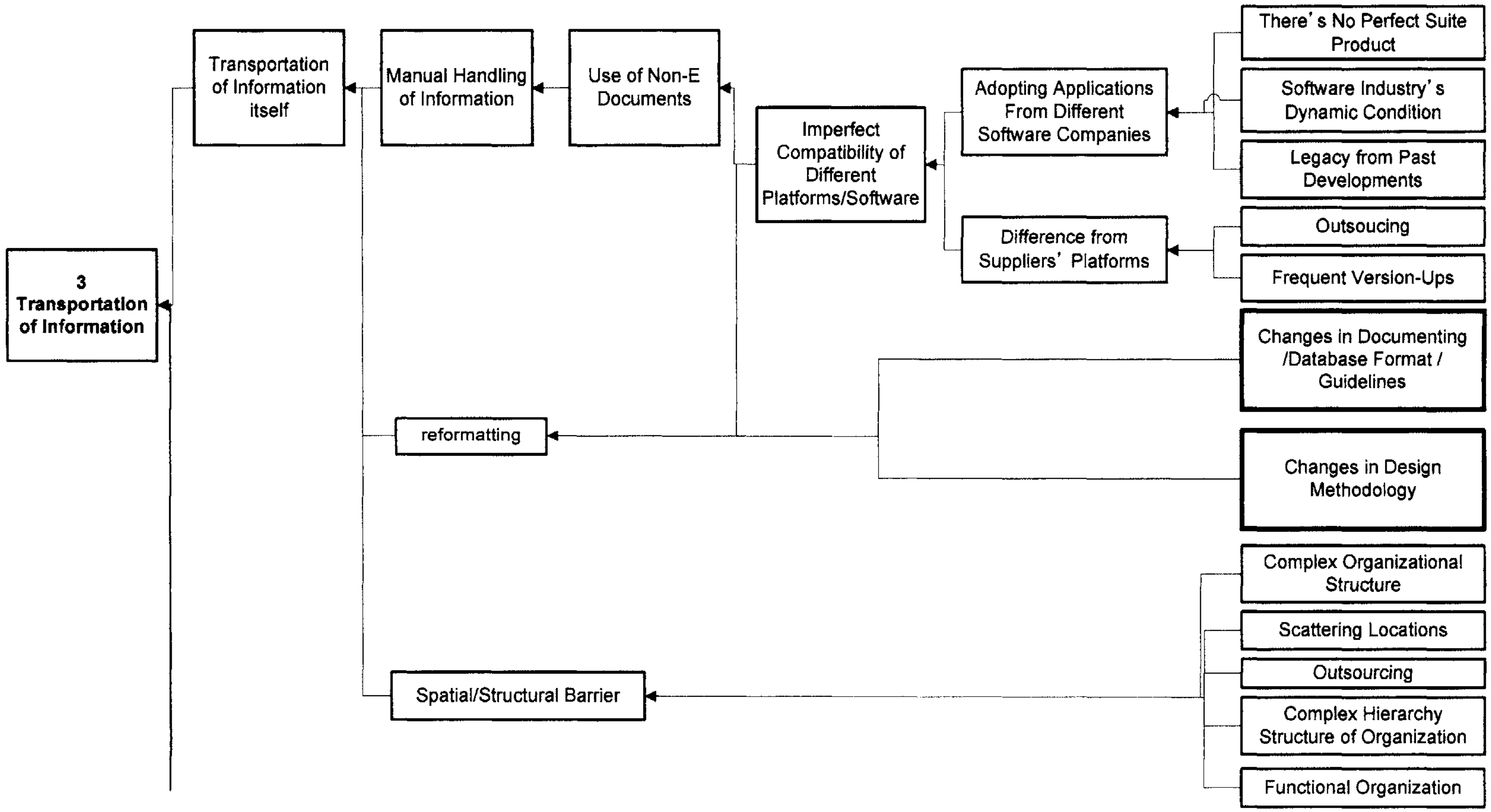
Unscheduled Waiting
Suitable Designer's Temporal Unavailability
Fatigue
Limited Resources(1) System Over Utilization (Unrealistic Schedule)
Poor Resource Allocation
Local Over Utilization
No Level Scheduling
Limited Resources(2) Limited Qualification of Designers
No Replacement
Limited Resources(2) Limited Qualification of Designers
Tardy Approval Process
3.Transportation
Go to "3.Transportation"
Inside of the Organization
Idea of DFM (Downstream Tasks' Commitment to Upstream Tasks)
Concurrent Engineering(1) (Designers Get Involved in Production's/User's Problem)
Critical Design Issues
Trivial Questions
Outsourcing
Waiting for Validation of Information
Outside of the Organization
Lack of Accumulated Data/Expertise In the Specific Area
Entering into the Market New to The Organization
Waiting for Answers for Inquiry
Waiting for Completed Information
3.Transportation
Go to "3.Transportation"
Waiting for Info.Processed INSIDE of the Organization
Failing to send Inquiry in Advance
Unplanned Communication
Inadequate Scheduling Technique

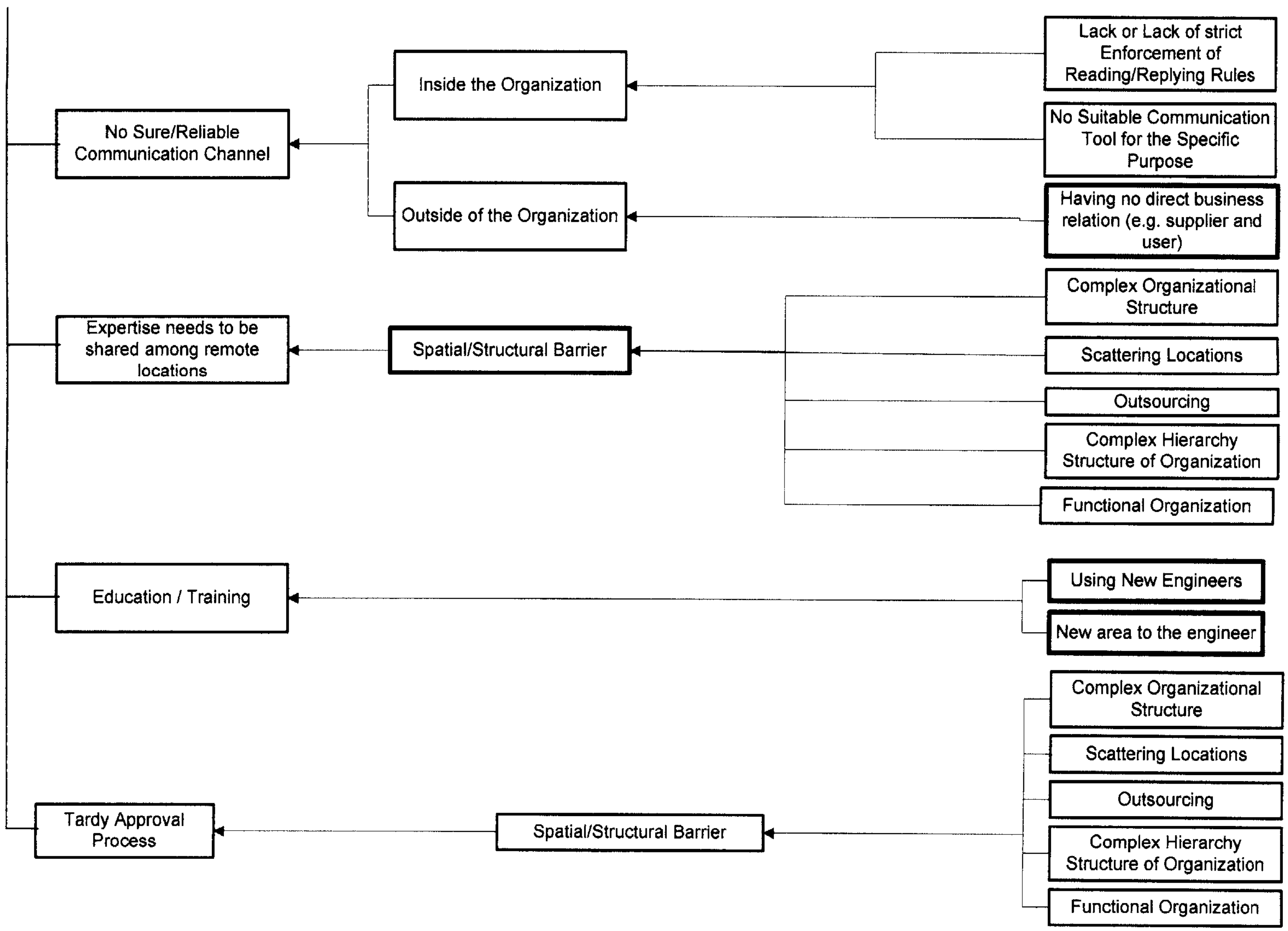

Nature of PD(3) Identifying All Interfaces In Advance is Impossible
Answer Delayed
Limited Resources(5) System Over Utilization On The Organization Level
Waiting for Info.Processed OUTSIDE of the Organization
Lack of Accumulated Date/Expertise in the Specific Area
Entering into the Market New to the Organization
Dispersed Information w/o Maintained Index (Paper,Electronic)
Easiness to Get More Structured Information from Outside Than Doing Information Hunting In the Organization
Difficulty to Find the Right Person
Poor Knowledge Management
Difficulty to Reach the Right Person
Waiting for Information Processed by Hardware
Obsolete Hardware
Nature of the Task
Complex Product Architecture with Excessive Interfaces
Existence of Inter-Dependency
PD's Nature(2) Iteration cannot be Eliminated
Waiting for Information from Mutually Dependent Tasks
Engineer's Neglect of Schedule
Lack of Strong Enforcement of schedule (Or Effective Incentive)


Reuse of General Schedule Not Optimized for the Current Project
No Leveled Scheduling
Limited Resources(1) System Over Utilization (Urealistic Schesule)
Too Tight (Unrealistic) Schedule
Delay in Mutually Dependent Tasks
Highly Competitive Market
Time-Sensitive Product
Way-Off Schedule
Inadequate Acheduling
9.Defective Information
Happenings
Unidentified Risks/Uncertainties
Limited Resources(7) Limited Tools / Prototypes / Hardware
Waiting for Tool/Material/ Hardware/Software
Insufficient Maintenance of Development Environment

3. TRANSPORTATION

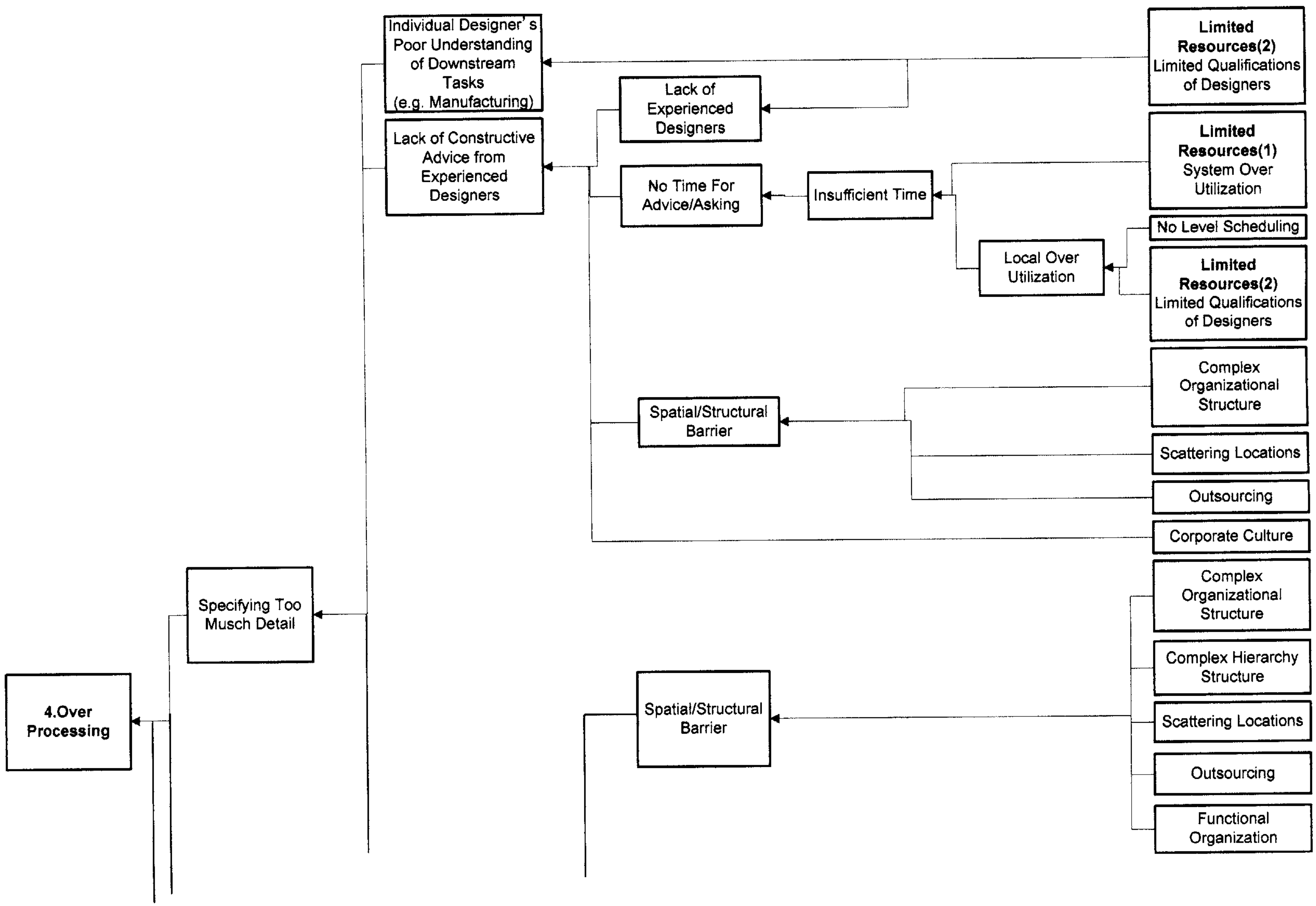
Root-Cause

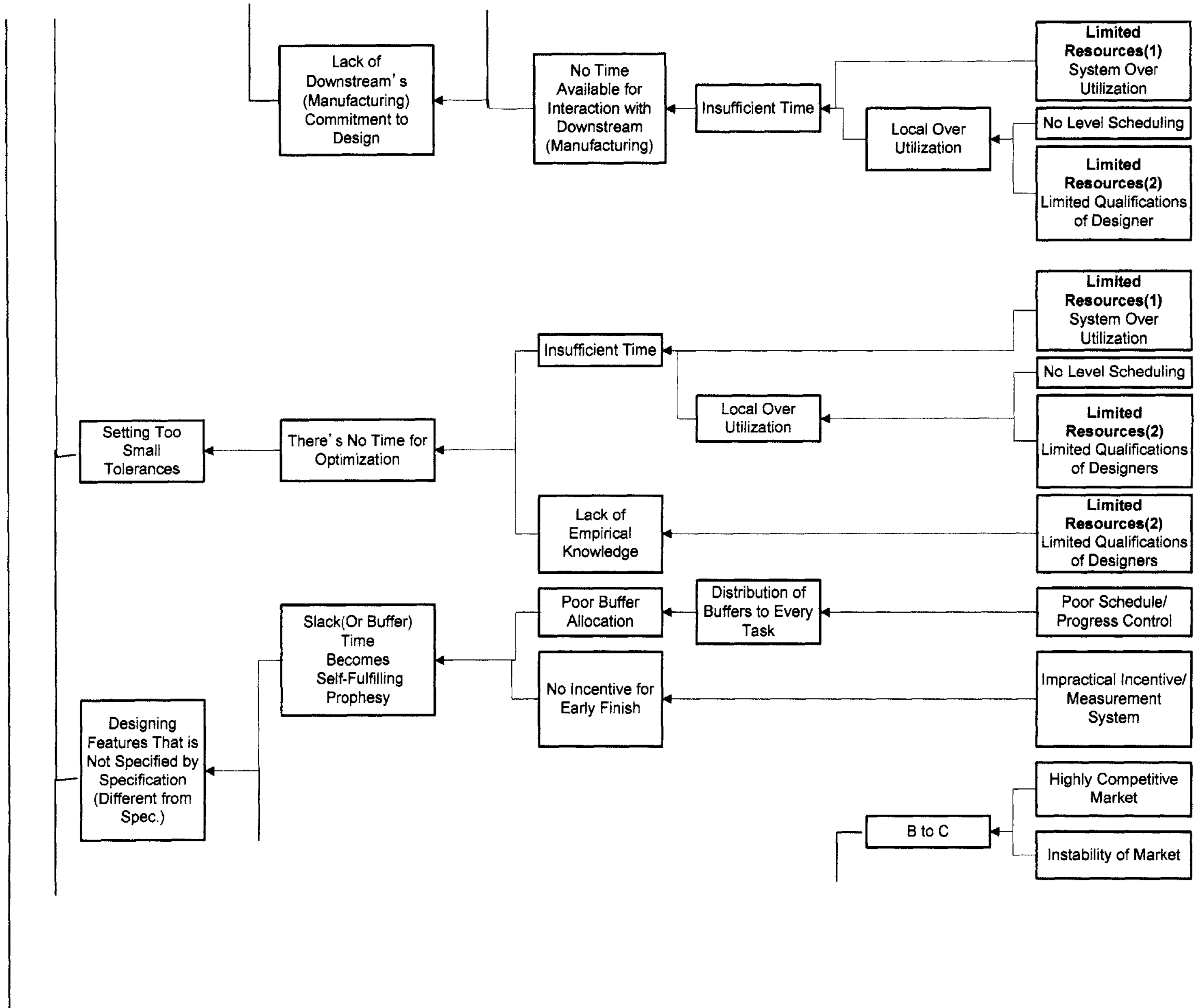


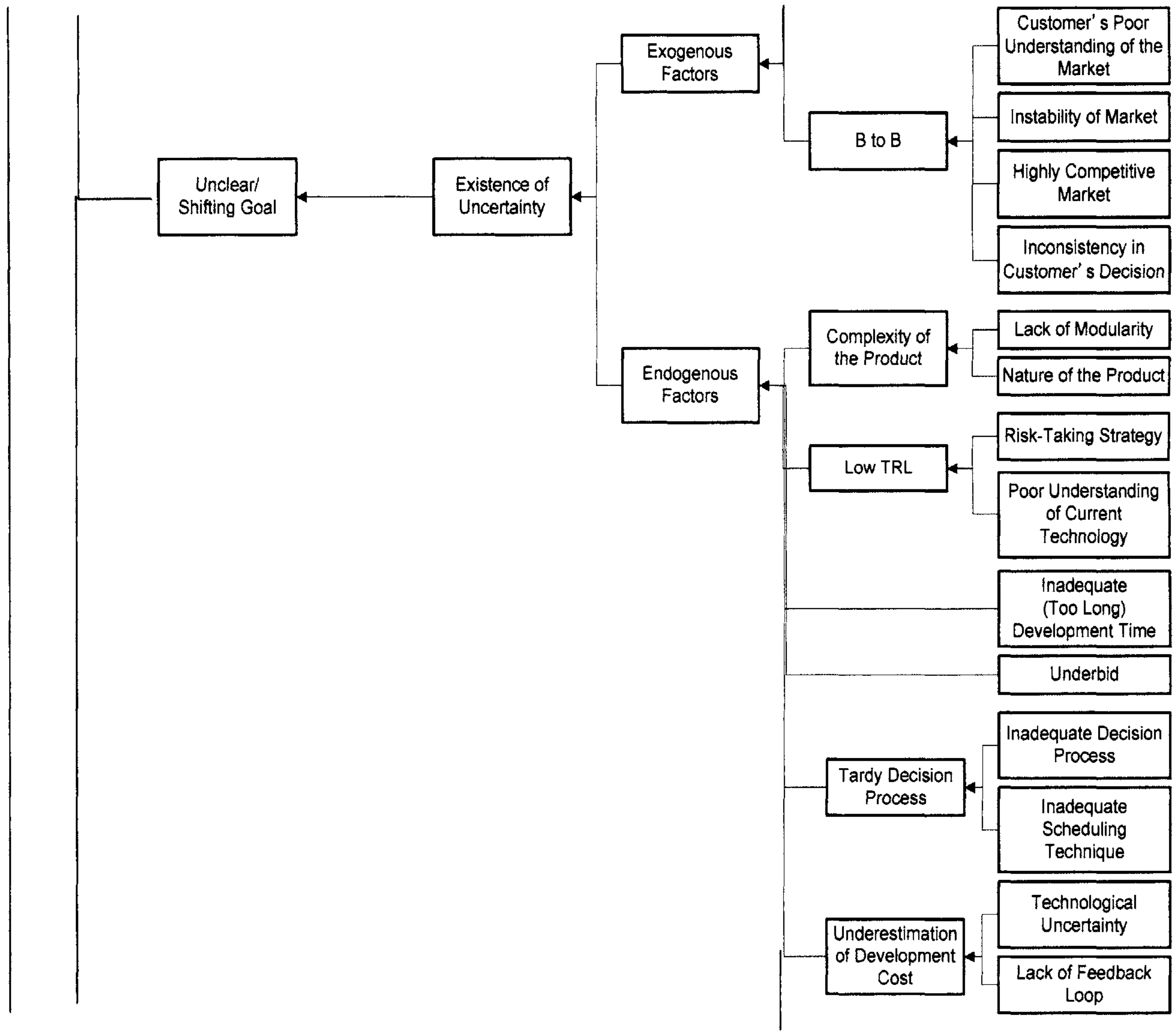

No Sure/Reliable Communication Channel
Inside the Organization
Outside of the Organization
Lack or Lack of strict Enforcement of Reading/Replying Rules
No Suitable Communication Tool for the Specific Purpose
Having no direct business relation (e.g. supplier and user)
Expertise needs to be shared among remote locations
Spatial/Structural Barrier
Complex Organizational Structure
Scattering Locations
Outsourcing
Complex Hierarchy Structure of Organization
Functional Organization
Education / Training
Using New Engineers
New area to the engineer
Tardy Approval Process
Spatial/Structural Barrier
Complex Organizational Structure
Scattering Locations
Outsourcing
Complex Hierarchy Structure of Organization
Functional Organization

4 OVER PROCESSING

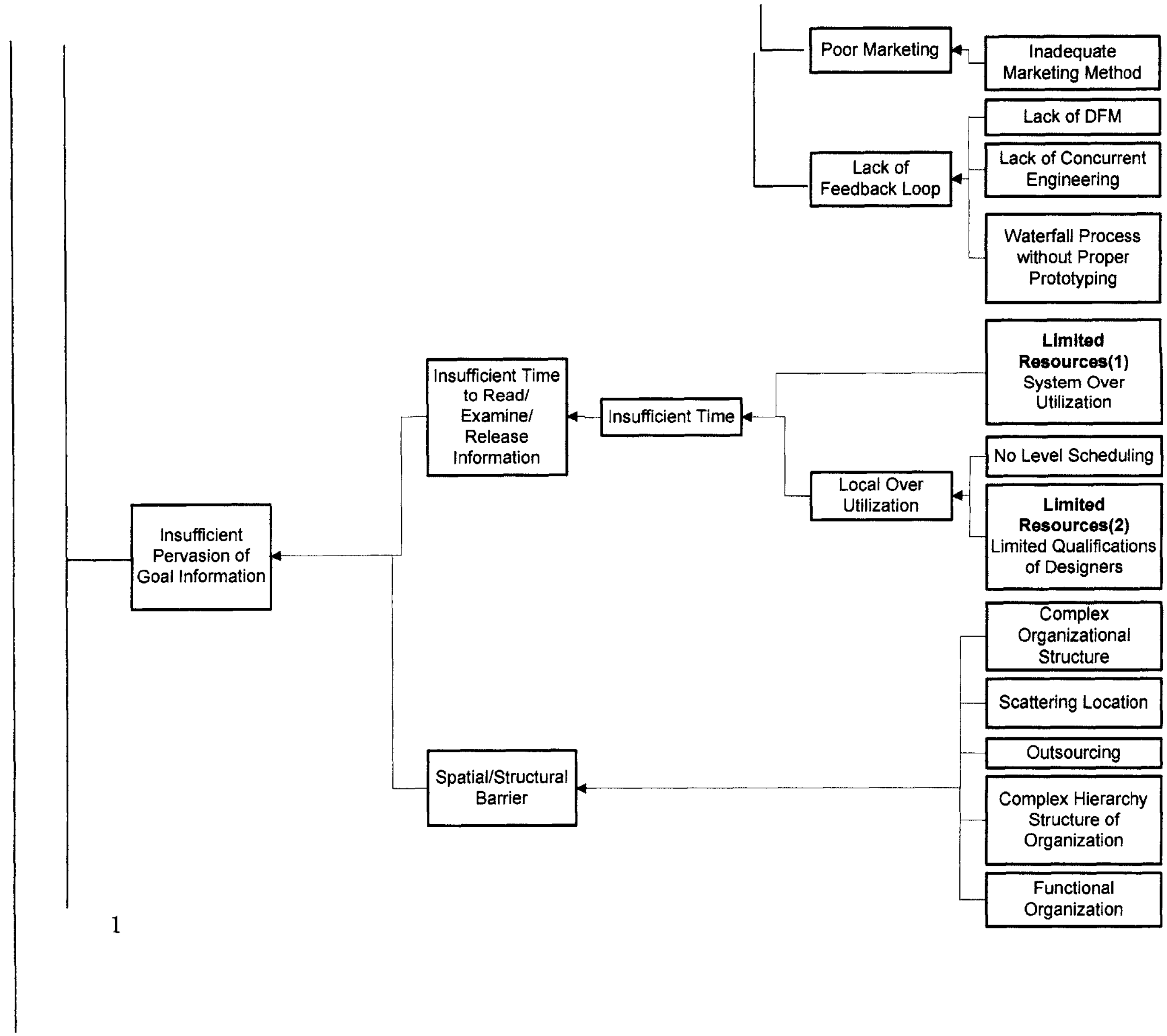
Root-Cause



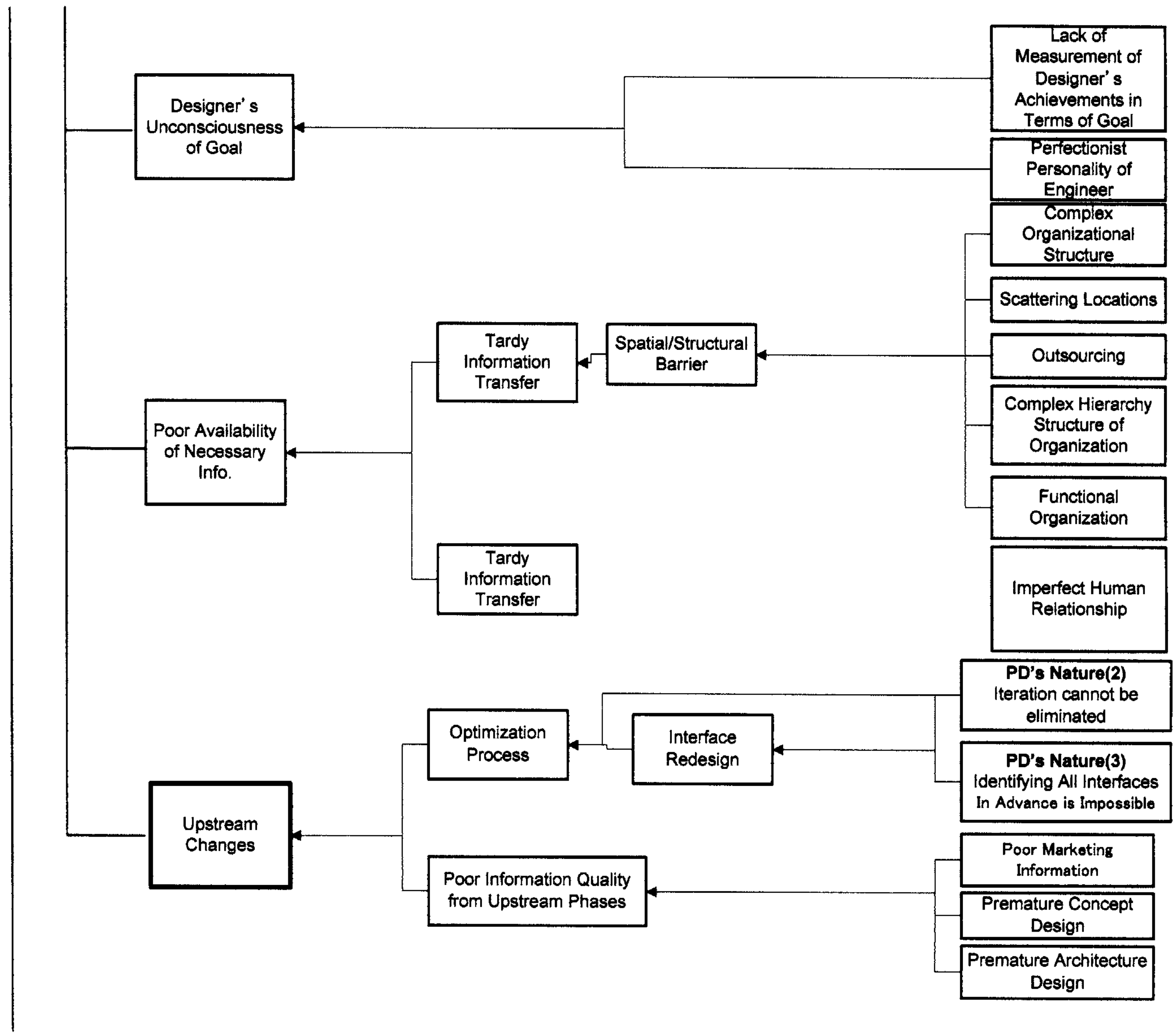

Lack of Downstream's (Manufacturing) Commitment to Design
No Time Available for Interaction with Downstream (Manufacturing)
Insufficient Time
Local Over Utilization
Limited Resources(1) System Over Utilization
No Level Scheduling
Limited Resources(2) Limited Qualifications of Designer
Setting Too Small Tolerances
There's No Time for Optimization
Insufficient Time
Local Over Utilization
Limited Resources(1) System Over Utilization
No Level Scheduling
Limited Resources(2) Limited Qualifications of Designers
Lack of Empirical Knowledge
Limited Resources(2) Limited Qualifications of Designers
Designing Features That is Not Specified by Specification (Different from Spec.)
Slack(Or Buffer) Time Becomes Self-Fulfilling Prophesy
Poor Buffer Allocation
Distribution of Buffers to Every Task
Poor Schedule/ Progress Control
No Incentive for Early Finish
Impractical Incentive/ Measurement System
B to C
Highly Competitive Market
Instability of Market

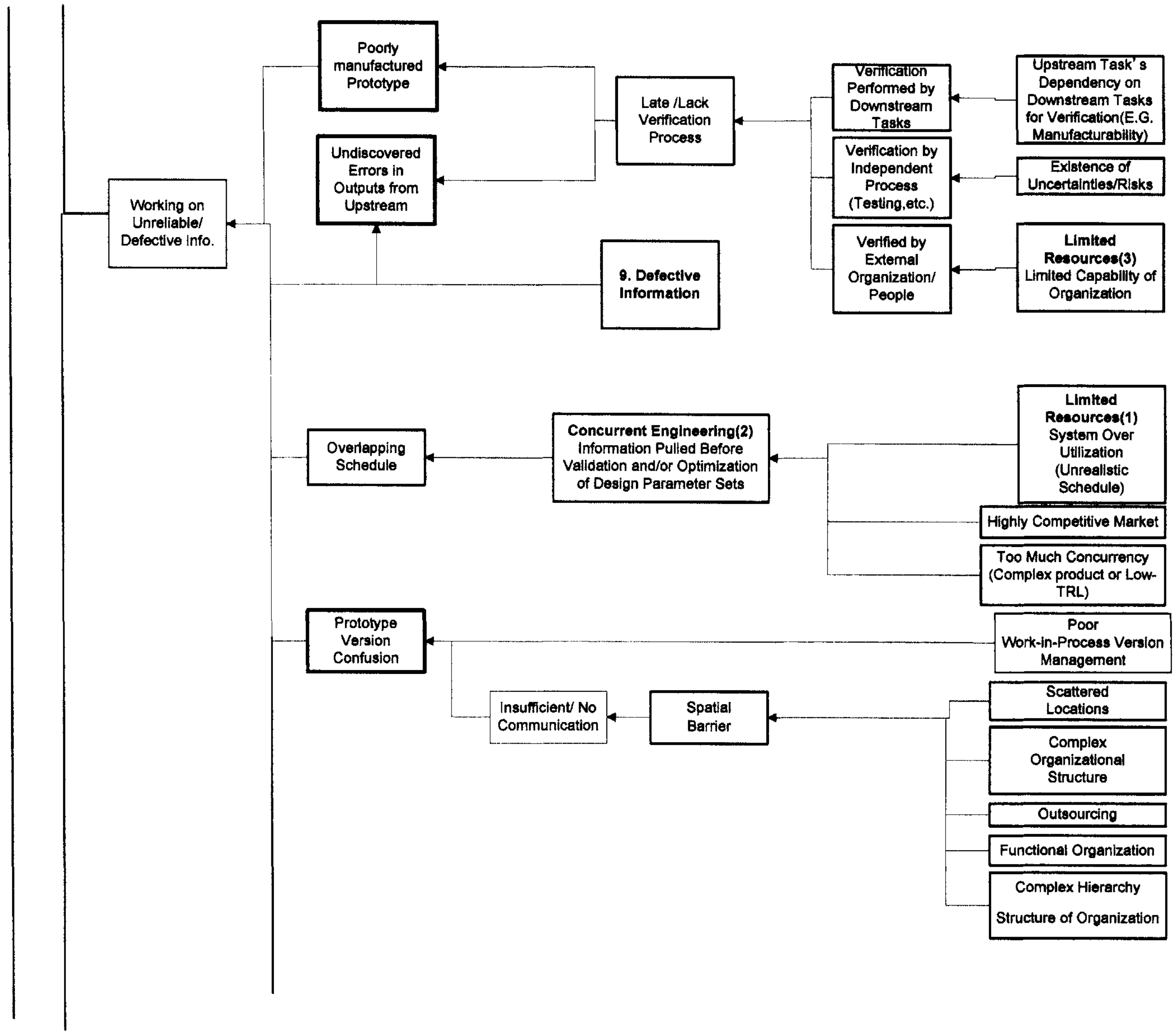


Unclear/ Shifting Goal
Existence of Uncertainty
Exogenous Factors
B to B
Customer' s Poor Understanding of the Market
Instability of Market
Highly Competitive Market
Inconsistency in Customer' s Decision
Endogenous Factors
Complexity of the Product
Lack of Modularity
Nature of the Product
Low TRL
Risk-Taking Strategy
Poor Understanding of Current Technology
Inadequate (Too Long) Development Time
Underbid
Tardy Decision Process
Inadequate Decision Process
Inadequate Scheduling Technique
Underestimation of Development Cost
Technological Uncertainty
Lack of Feedback Loop

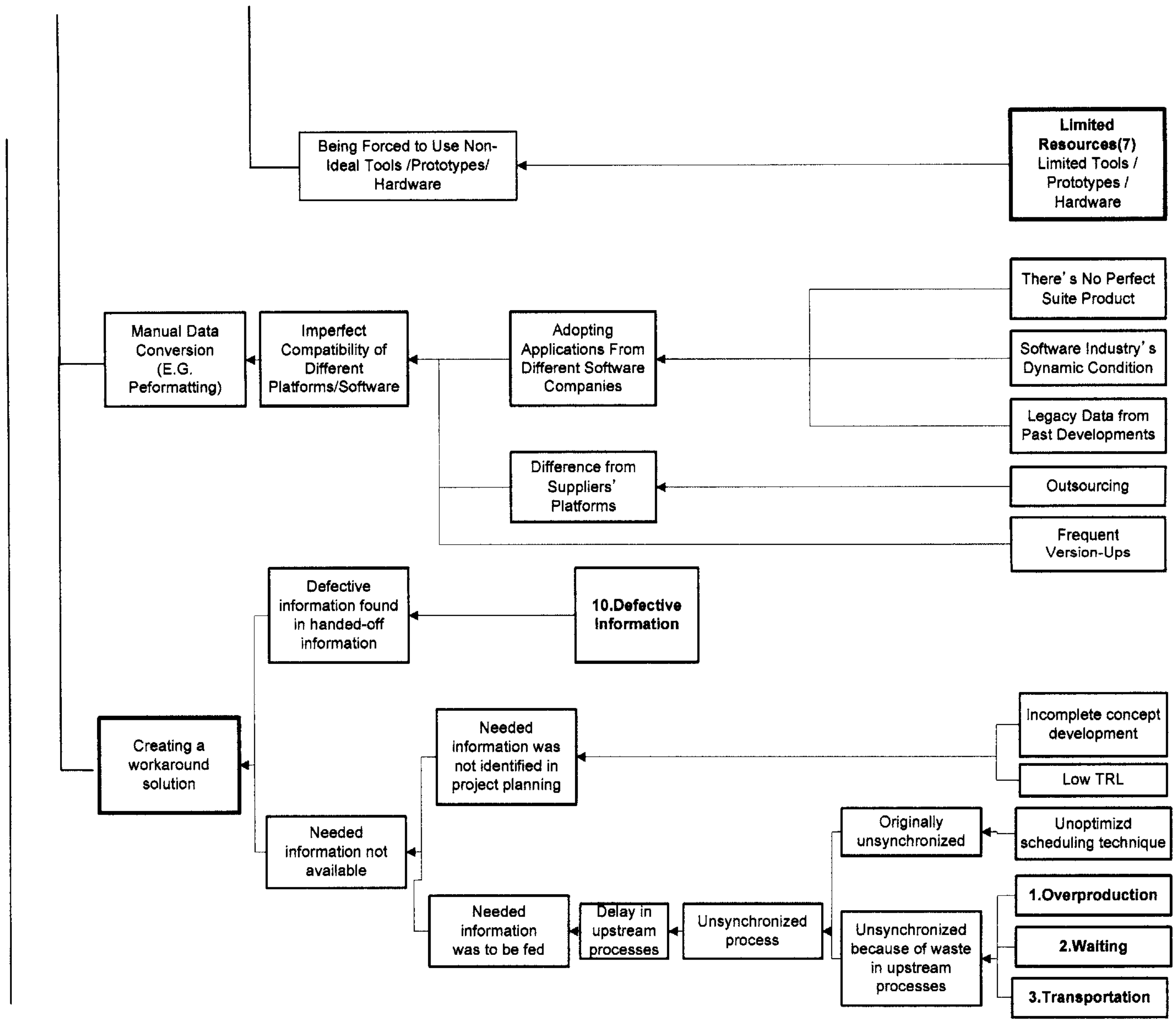


Poor Marketing
Inadequate Marketing Method
Lack of DFM
Lack of Feedback Loop
Lack of Concurrent Engineering
Waterfall Process without Proper Prototyping
Limited Resources(1) System Over Utilization
Insufficient Time to Read/ Examine/ Release Information
Insufficient Time
No Level Scheduling
Local Over Utilization
Limited Resources(2) Limited Qualifications of Designers
Insufficient Pervasion of Goal Information
Complex Organizational Structure
Scattering Location
Outsourcing
Spatial/Structural Barrier
Complex Hierarchy Structure of Organization
Functional Organization
1

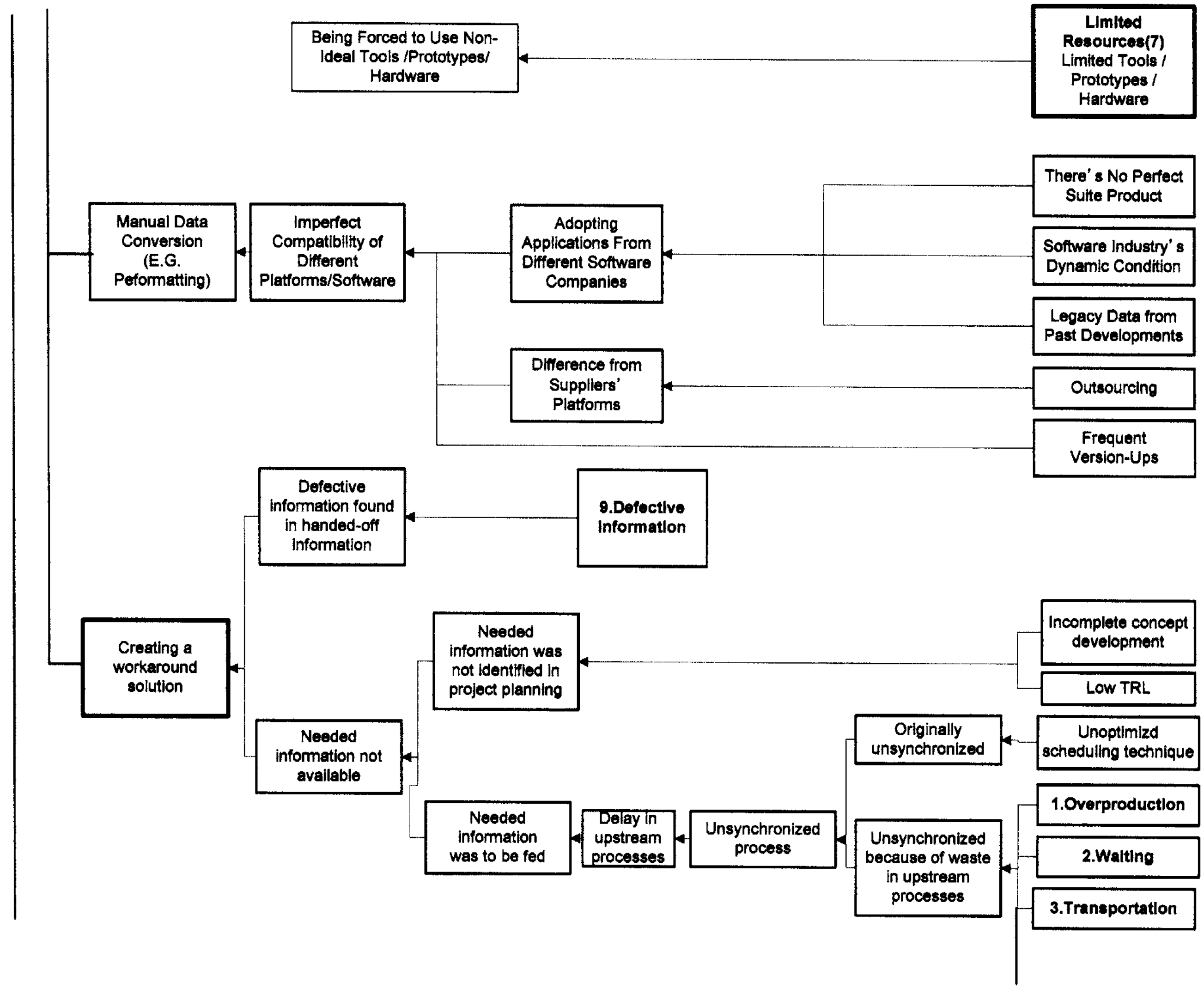

Designer's Unconsciousness of Goal
Lack of Measurement of Designer's Achievements in Terms of Goal
Perfectionist Personality of Engineer
Complex Organizational Structure
Scattering Locations
Tardy Information Transfer
Spatial/Structural Barrier
Outsourcing
Poor Availability of Necessary Info.
Complex Hierarchy Structure of Organization
Functional Organization
Tardy Information Transfer
Imperfect Human Relationship
PD's Nature(2) Iteration cannot be eliminated
Optimization Process
Interface Redesign
PD's Nature(3) Identifying All Interfaces In Advance is Impossible
Upstream Changes
Poor Marketing Information
Poor Information Quality from Upstream Phases
Premature Concept Design
Premature Architecture Design

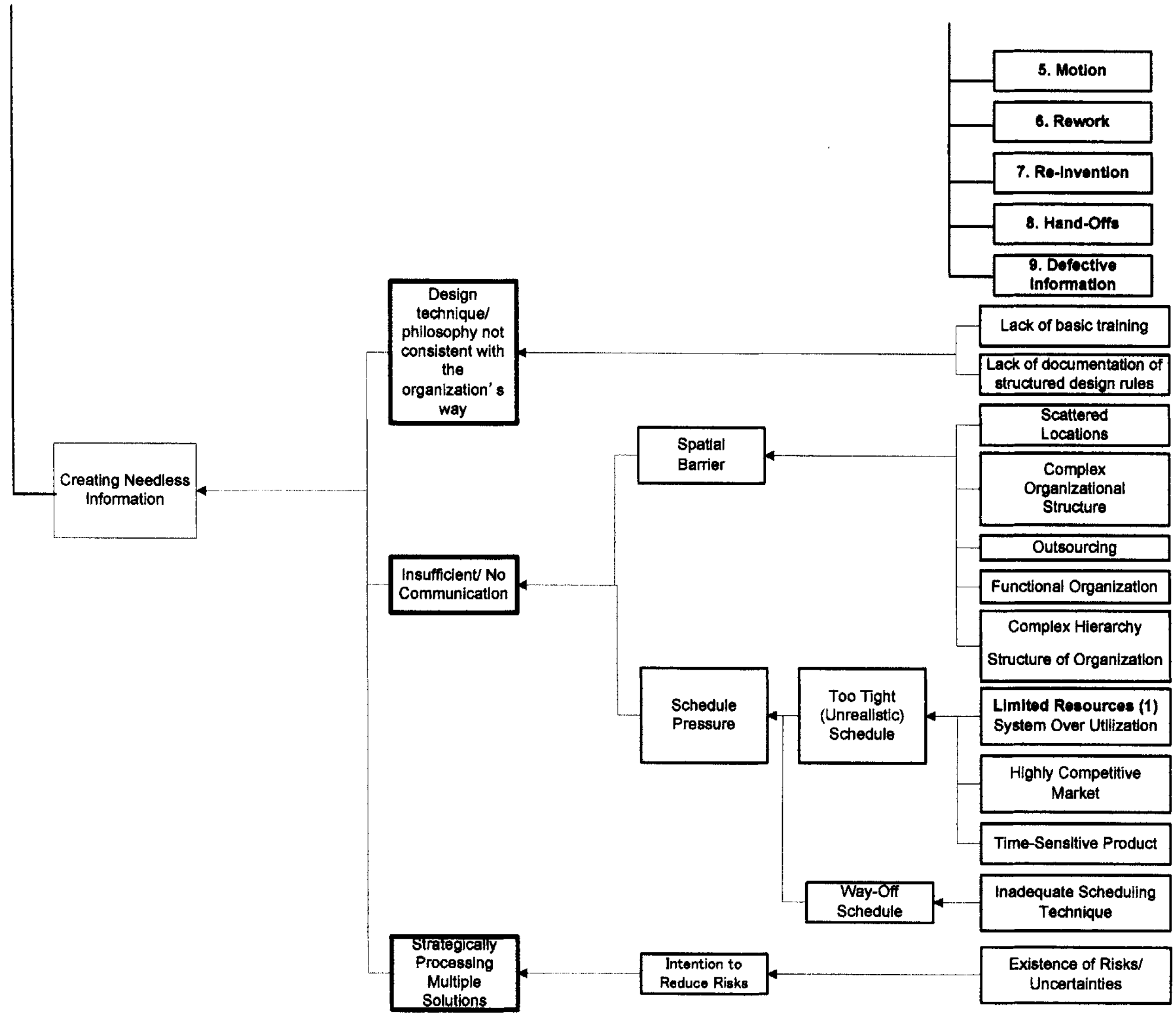


Poorly manufactured Prototype
Undiscovered Errors in Outputs from Upstream
Late /Lack Verification Process
Verification Performed by Downstream Tasks
Upstream Task's Dependency on Downstream Tasks for Verification(E.G. Manufacturability)
Verification by Independent Process (Testing,etc.)
Existence of Uncertainties/Risks
Verified by External Organization/ People
Limited Resources(3) Limited Capability of Organization
Working on Unreliable/ Defective Info.
9. Defective Information
Overlapping Schedule
Concurrent Engineering(2) Information Pulled Before Validation and/or Optimization of Design Parameter Sets
Limited Resources(1) System Over Utilization (Unrealistic Schedule)
Highly Competitive Market
Too Much Concurrency (Complex product or Low-TRL)
Prototype Version Confusion
Poor Work-In-Process Version Management
Insufficient/ No Communication
Spatial Barrier
Scattered Locations
Complex Organizational Structure
Outsourcing
Functional Organization
Complex Hierarchy
Structure of Organization

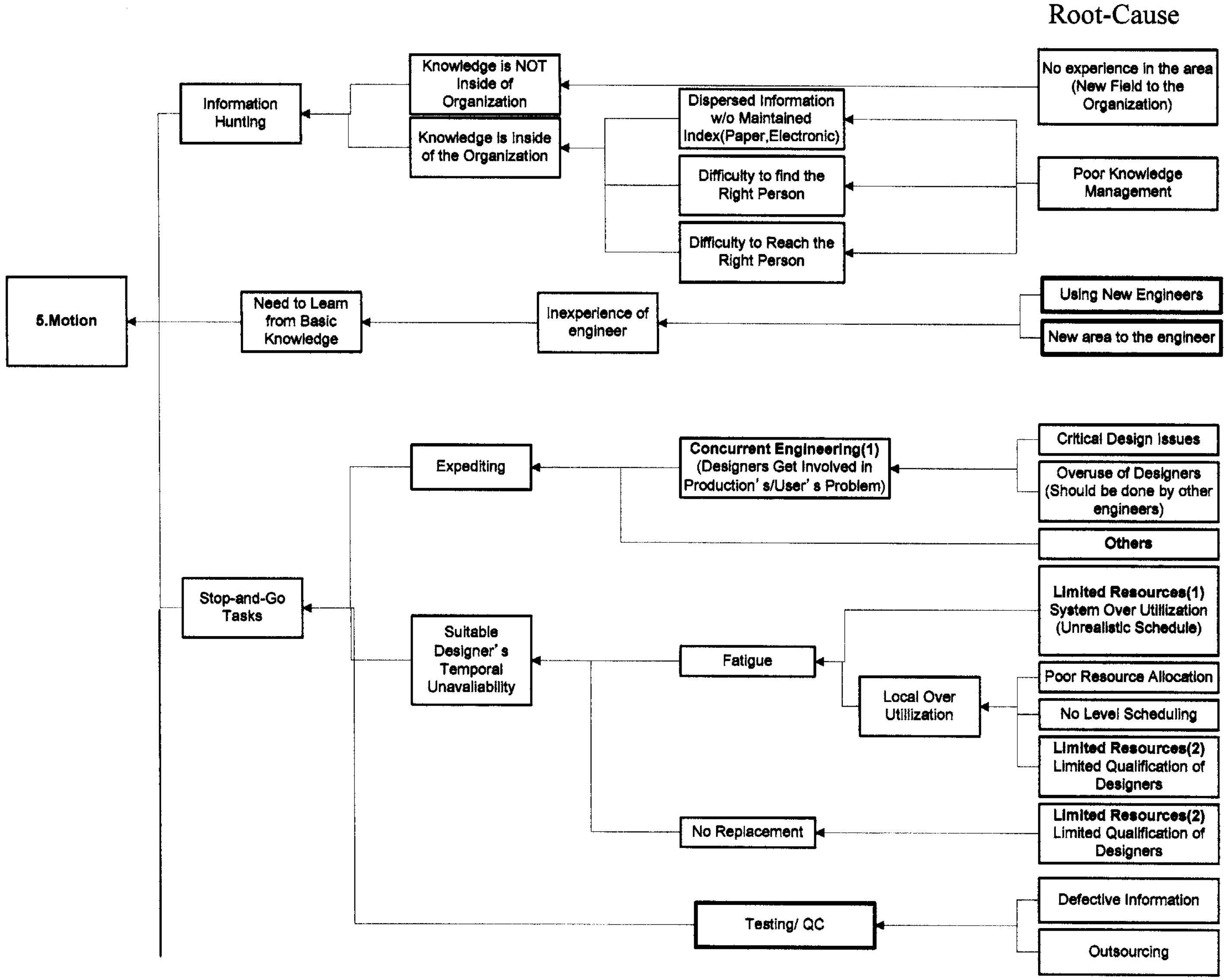


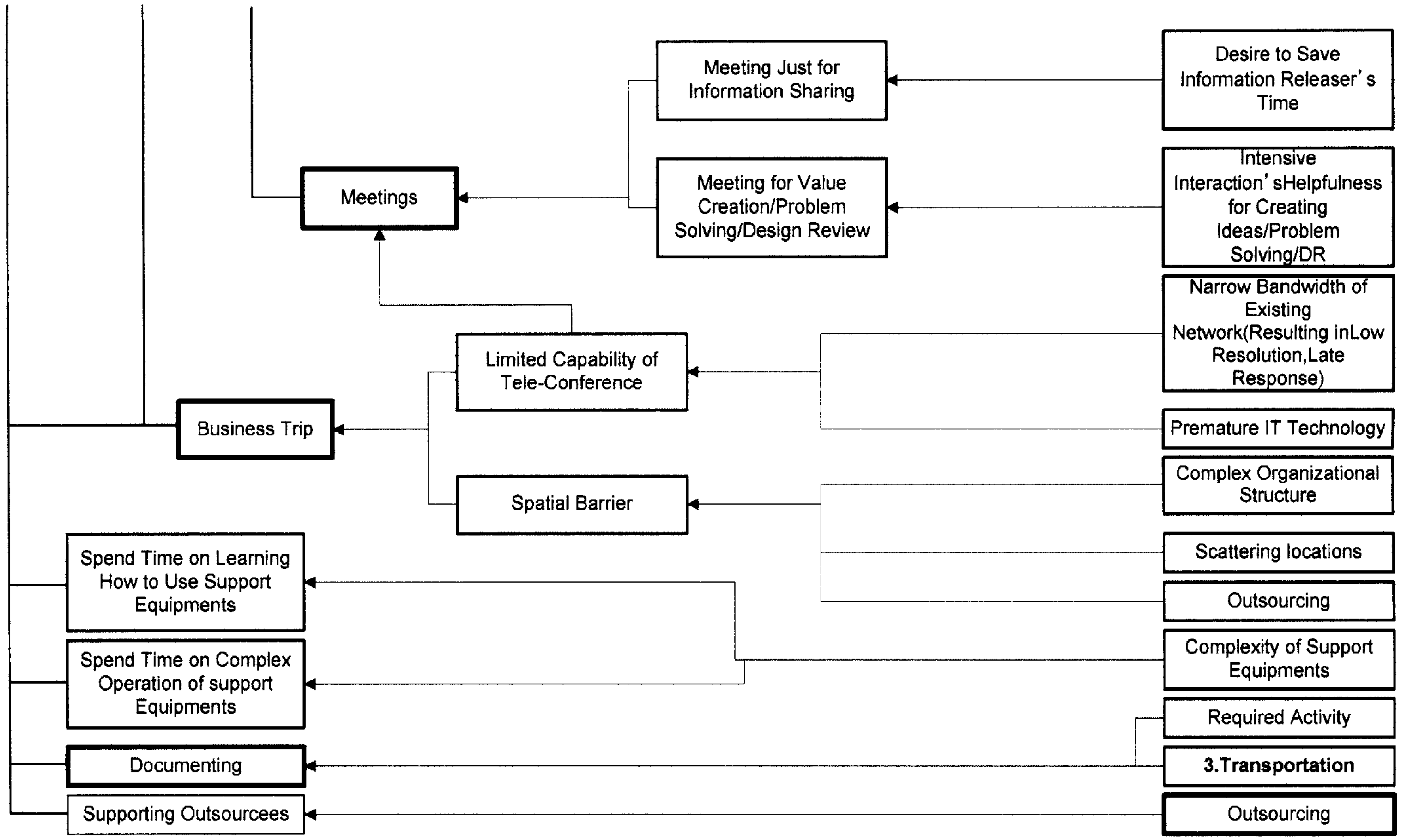
Being Forced to Use Non-Ideal Tools /Prototypes/ Hardware
Limited Resources(7) Limited Tools / Prototypes / Hardware
There's No Perfect Suite Product
Manual Data Conversion (E.G. Peformatting)
Imperfect Compatibility of Different Platforms/Software
Adopting Applications From Different Software Companies
Software Industry's Dynamic Condition
Legacy Data from Past Developments
Difference from Suppliers' Platforms
Outsourcing
Frequent Version-Ups
Defective information found in handed-off information
10.Defective Information
Incomplete concept development
Creating a workaround solution
Needed information was not identified in project planning
Low TRL
Originally unsynchronized
Unoptimizd scheduling technique
Needed information not available
Needed information was to be fed
Delay in upstream processes
Unsynchronized process
Unsynchronized because of waste in upstream processes
1.Overproduction
2.Waiting
3.Transportation


Being Forced to Use Non-Ideal Tools /Prototypes/ Hardware
Limited Resources(7) Limited Tools / Prototypes / Hardware
There's No Perfect Suite Product
Manual Data Conversion (E.G. Peformatting)
Imperfect Compatibility of Different Platforms/Software
Adopting Applications From Different Software Companies
Software Industry's Dynamic Condition
Legacy Data from Past Developments
Difference from Suppliers' Platforms
Outsourcing
Frequent Version-Ups
Defective information found in handed-off information
9.Defective Information
Creating a workaround solution
Needed information was not identified in project planning
Incomplete concept development
Low TRL
Originally unsynchronized
Unoptimizd scheduling technique
Needed information not available
Needed information was to be fed
Delay in upstream processes
Unsynchronized process
Unsynchronized because of waste in upstream processes
1.Overproduction
2.Waiting
3.Transportation


5. Motion
6. Rework
7. Re-Invention
8. Hand-Offs
9. Defective Information
Creating Needless Information
Design technique/ philosophy not consistent with the organization's way
Lack of basic training
Lack of documentation of structured design rules
Spatial Barrier
Scattered Locations
Complex Organizational Structure
Outsourcing
Functional Organization
Complex Hierarchy Structure of Organization
Insufficient/ No Communication
Schedule Pressure
Too Tight (Unrealistic) Schedule
Limited Resources (1) System Over Utilization
Highly Competitive Market
Time-Sensitive Product
Way-Off Schedule
Inadequate Scheduling Technique
Strategically Processing Multiple Solutions
Intention to Reduce Risks
Existence of Risks/ Uncertainties

5 MOTION

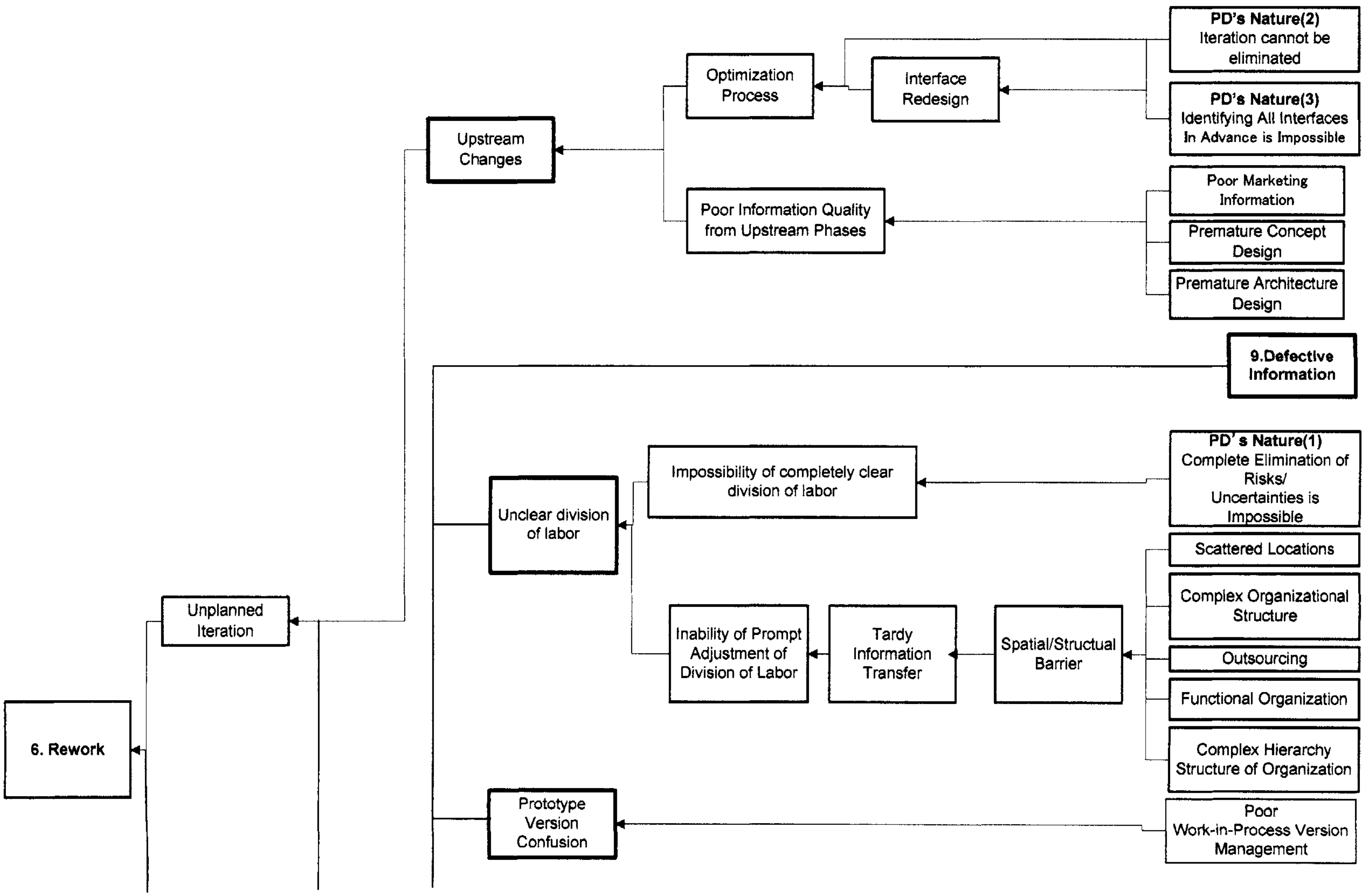

Root-Cause
5.Motion
Information Hunting
Knowledge is NOT inside of Organization
Knowledge is inside of the Organization
Dispersed Information w/o Maintained Index(Paper,Electronic)
Difficulty to find the Right Person
Difficulty to Reach the Right Person
No experience in the area (New Field to the Organization)
Poor Knowledge Management
Need to Learn from Basic Knowledge
Inexperience of engineer
Using New Engineers
New area to the engineer
Stop-and-Go Tasks
Expediting
Concurrent Engineering(1) (Designers Get Involved in Production' s/User' s Problem)
Critical Design Issues
Overuse of Designers (Should be done by other engineers)
Others
Suitable Designer' s Temporal Unavailability
Fatigue
Limited Resources(1) System Over Utilization (Unrealistic Schedule)
Local Over Utilization
Poor Resource Allocation
No Level Scheduling
Limited Resources(2) Limited Qualification of Designers
No Replacement
Limited Resources(2) Limited Qualification of Designers
Testing/ QC
Defective Information
Outsourcing

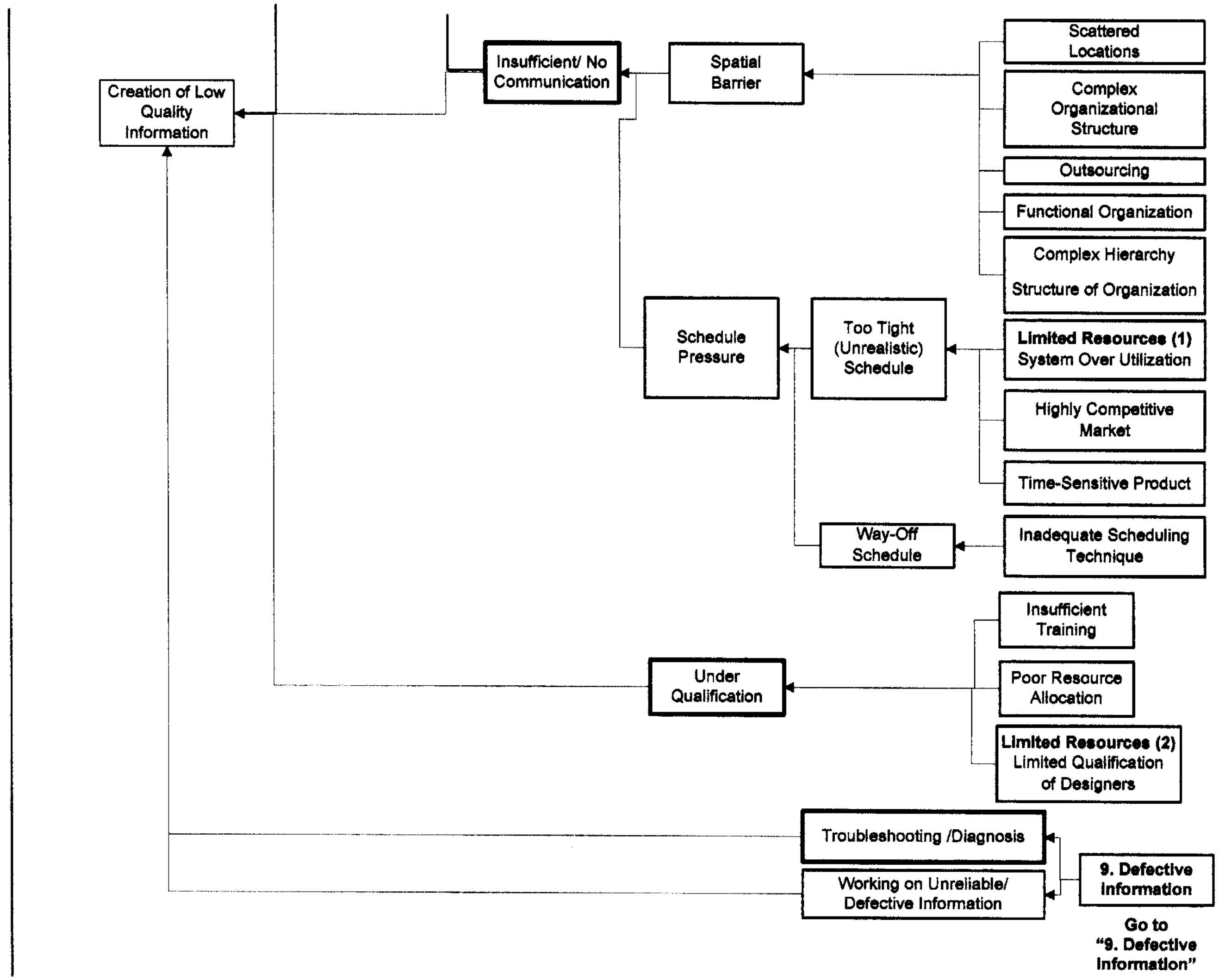


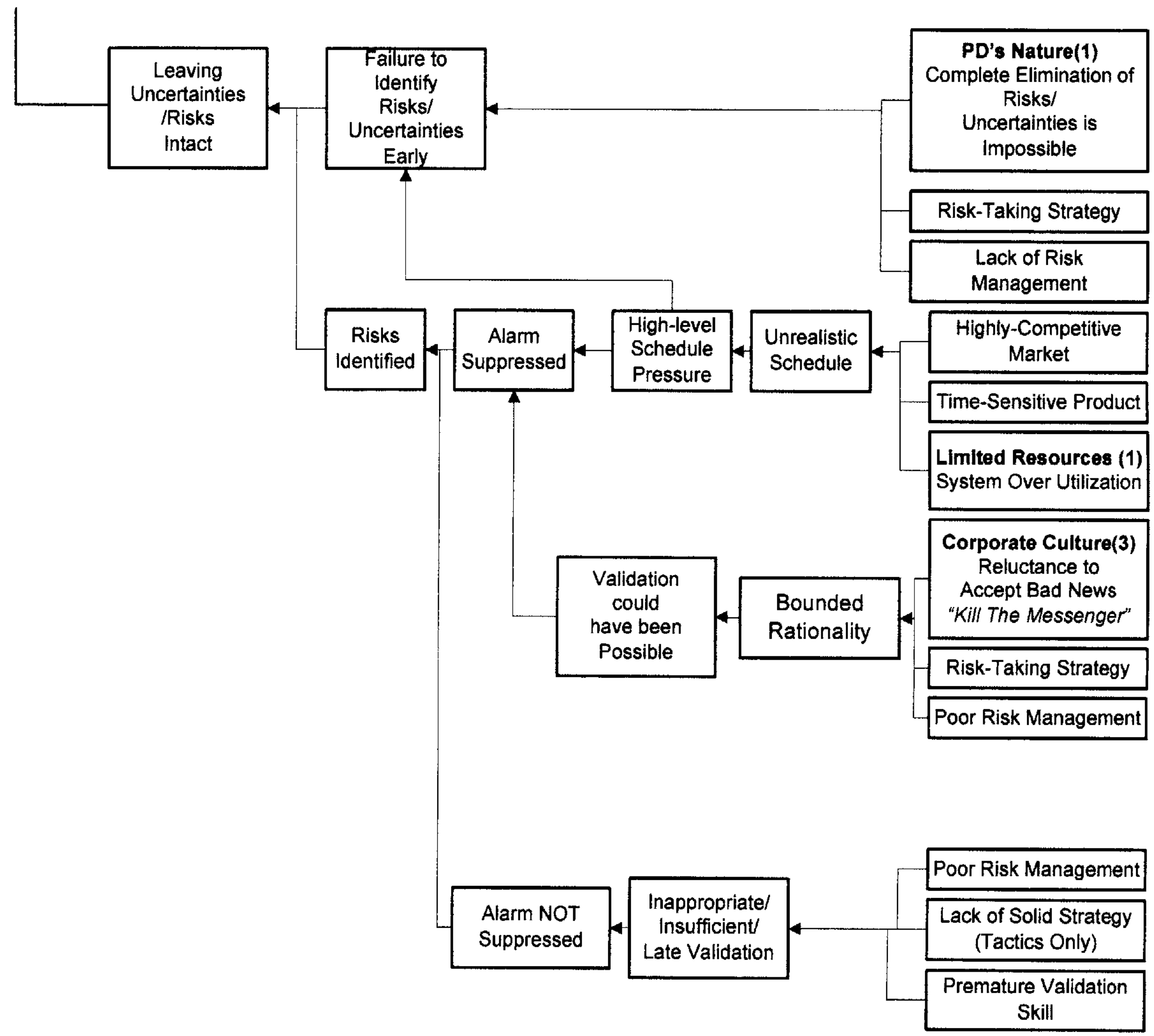
Meeting Just for Information Sharing
Desire to Save Information Releaser' s Time
Meetings
Meeting for Value Creation/Problem Solving/Design Review
Intensive Interaction' sHelpfulness for Creating Ideas/Problem Solving/DR
Narrow Bandwidth of Existing Network(Resulting inLow Resolution,Late Response)
Limited Capability of Tele-Conference
Business Trip
Premature IT Technology
Complex Organizational Structure
Spatial Barrier
Scattering locations
Spend Time on Learning How to Use Support Equipments
Outsourcing
Complexity of Support Equipments
Spend Time on Complex Operation of support Equipments
Required Activity
Documenting
3.Transportation
Supporting Outsourcees
Outsourcing

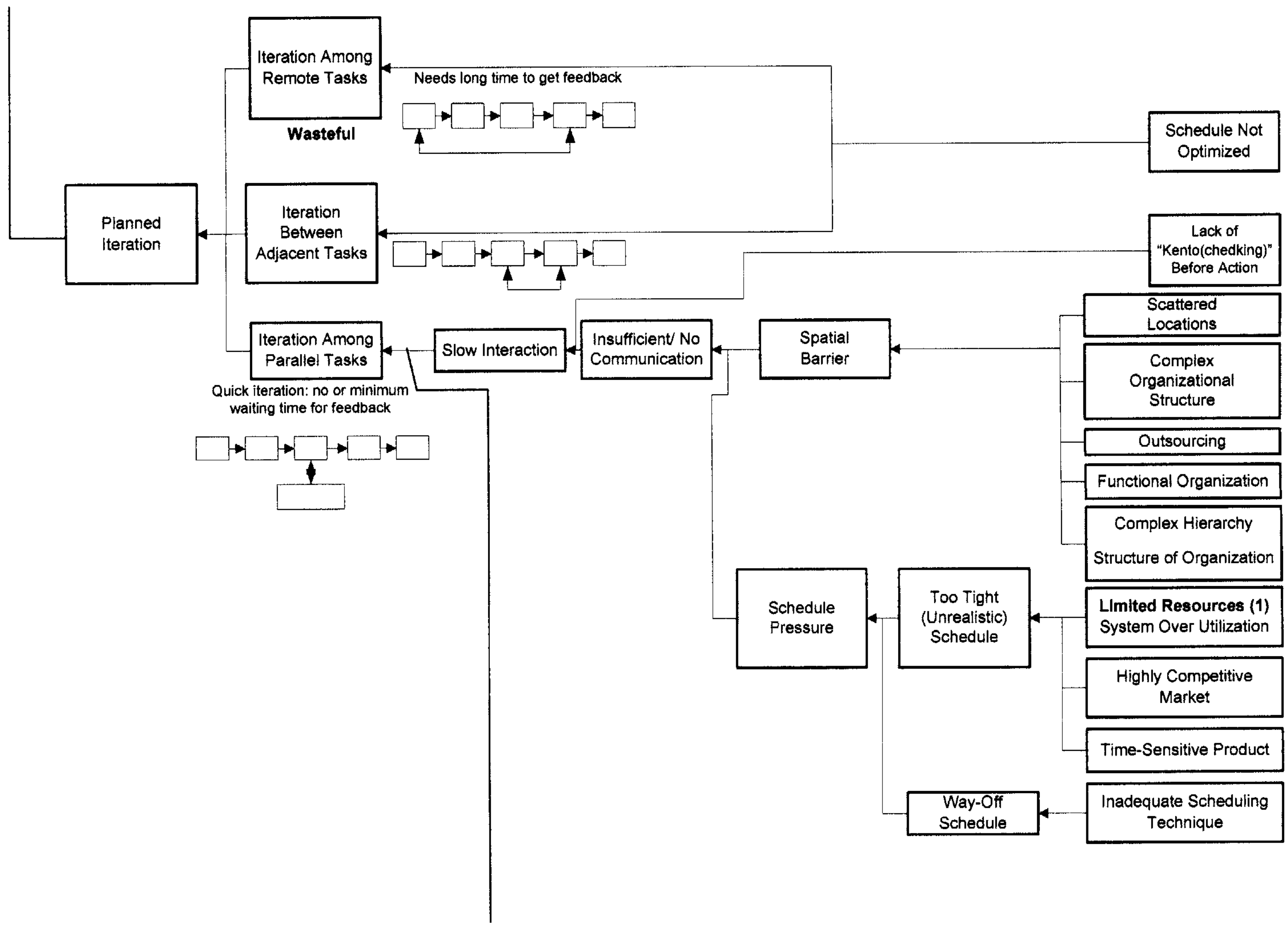
6 REWORK

Root-Cause

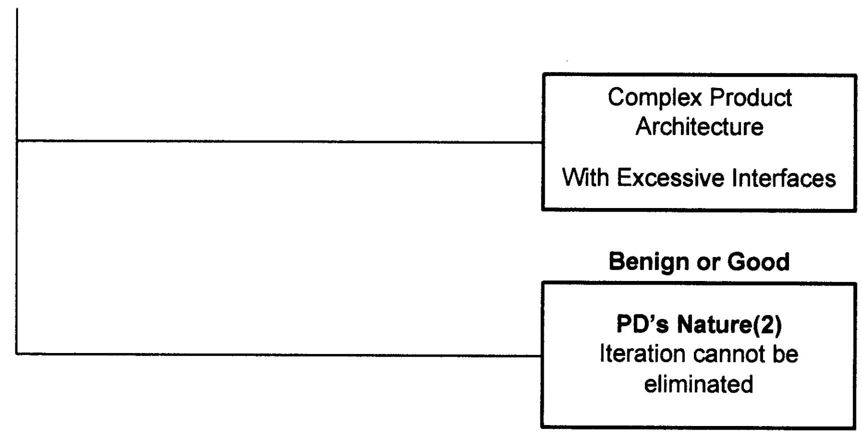



Creation of Low Quality Information
Insufficient/ No Communication
Spatial Barrier
Scattered Locations
Complex Organizational Structure
Outsourcing
Functional Organization
Complex Hierarchy Structure of Organization
Schedule Pressure
Too Tight (Unrealistic) Schedule
Limited Resources (1) System Over Utilization
Highly Competitive Market
Time-Sensitive Product
Way-Off Schedule
Inadequate Scheduling Technique
Insufficient Training
Under Qualification
Poor Resource Allocation
Limited Resources (2) Limited Qualification of Designers
Troubleshooting /Diagnosis
Working on Unreliable/ Defective Information
9. Defective Information
Go to "9. Defective Information"


Leaving Uncertainties /Risks Intact
Failure to Identify Risks/ Uncertainties Early
PD's Nature(1) Complete Elimination of Risks/ Uncertainties is Impossible
Risk-Taking Strategy
Lack of Risk Management
Risks Identified
Alarm Suppressed
High-level Schedule Pressure
Unrealistic Schedule
Highly-Competitive Market
Time-Sensitive Product
Limited Resources (1) System Over Utilization
Validation could have been Possible
Bounded Rationality
Corporate Culture(3) Reluctance to Accept Bad News "Kill The Messenger"
Risk-Taking Strategy
Poor Risk Management
Alarm NOT Suppressed
Inappropriate/ Insufficient/ Late Validation
Poor Risk Management
Lack of Solid Strategy (Tactics Only)
Premature Validation Skill

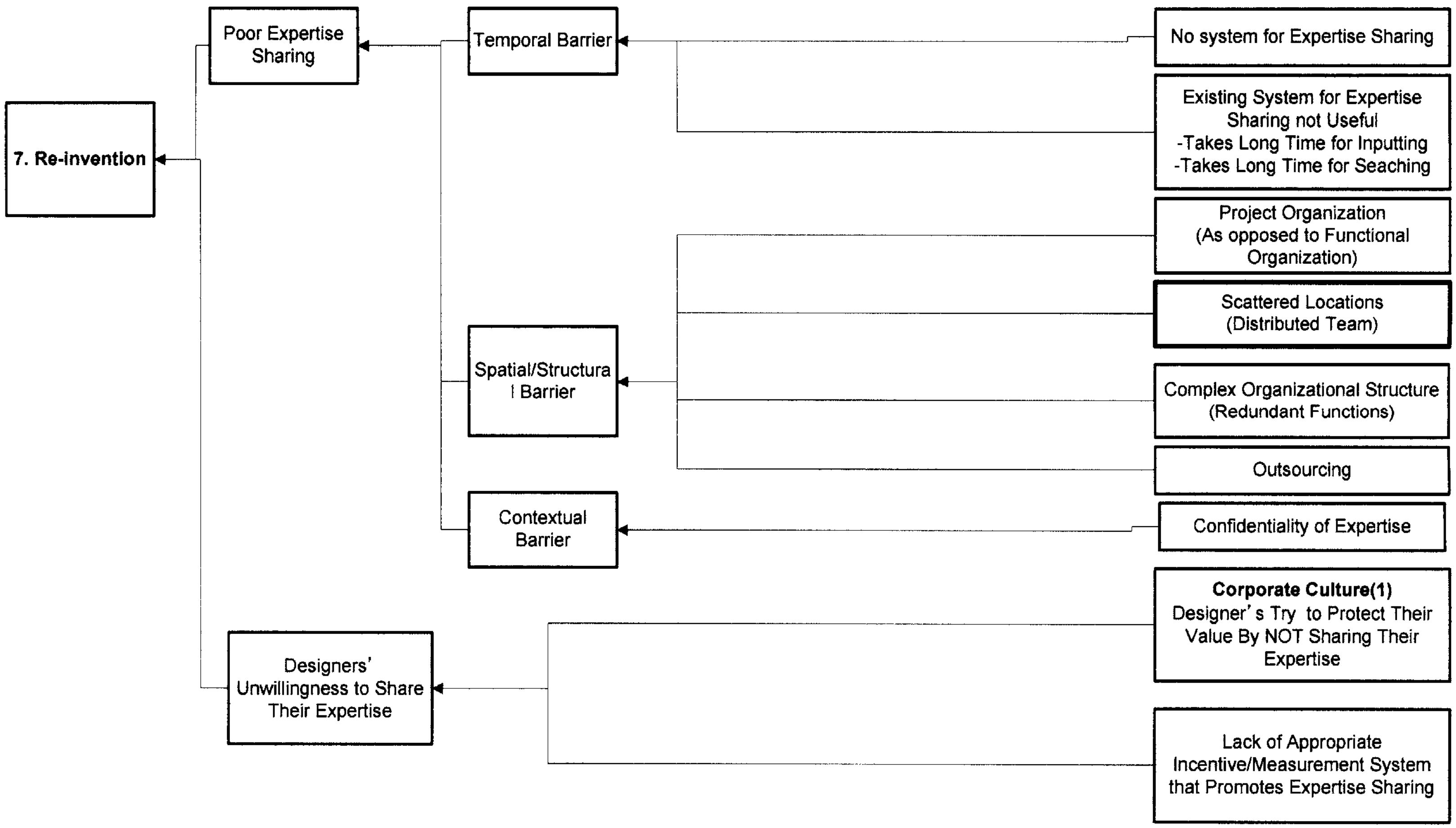


Iteration Among Remote Tasks
Needs long time to get feedback
Wasteful
Schedule Not Optimized
Planned Iteration
Iteration Between Adjacent Tasks
Lack of "Kento(chedking)" Before Action
Scattered Locations
Iteration Among Parallel Tasks
Slow Interaction
Insufficient/ No Communication
Spatial Barrier
Complex Organizational Structure
Quick iteration: no or minimum waiting time for feedback
Outsourcing
Functional Organization
Complex Hierarchy
Structure of Organization
Schedule Pressure
Too Tight (Unrealistic) Schedule
Limited Resources (1) System Over Utilization
Highly Competitive Market
Time-Sensitive Product
Way-Off Schedule
Inadequate Scheduling Technique

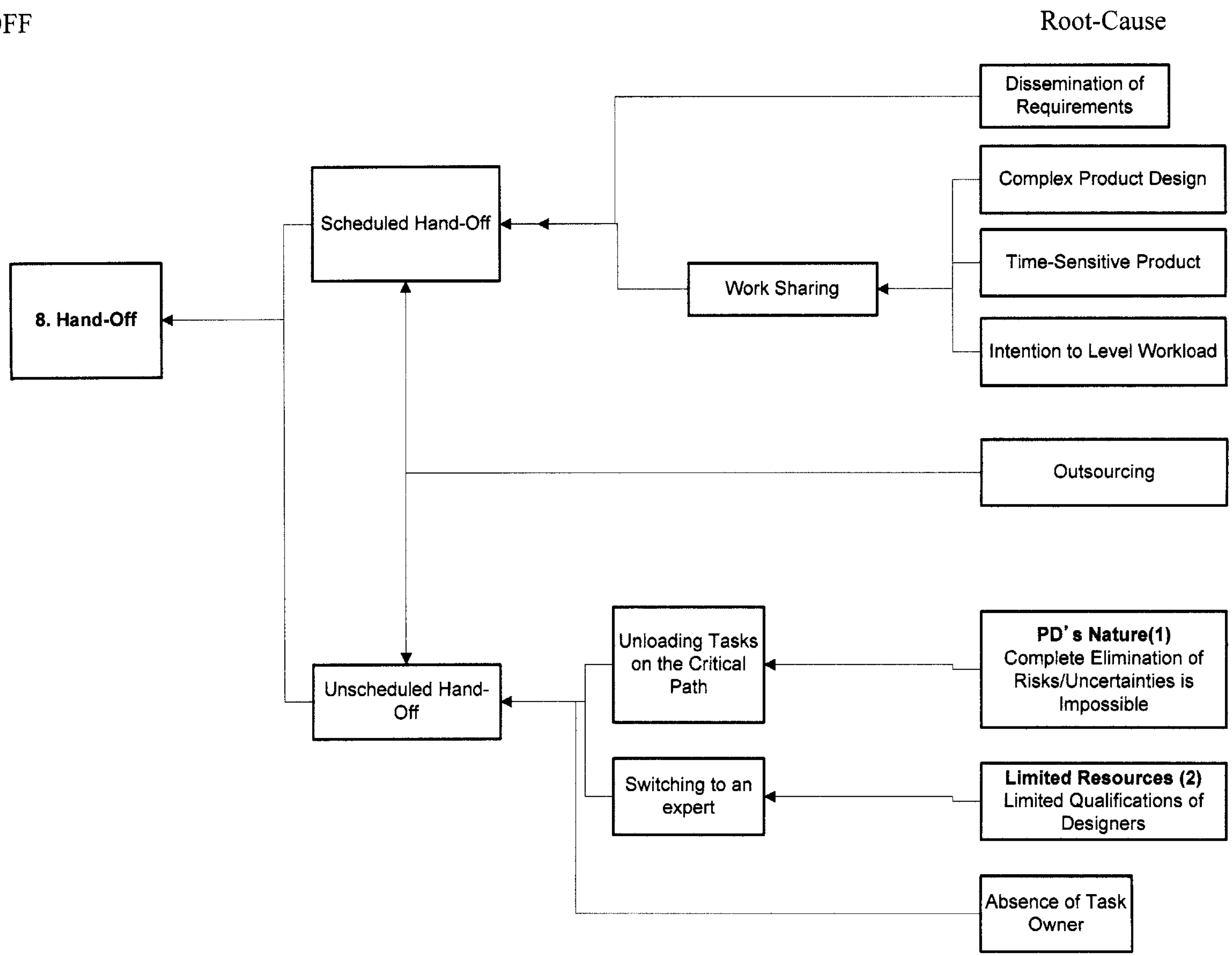

Complex Product Architecture
With Excessive Interfaces
Benign or Good
PD's Nature(2)
Iteration cannot be eliminated

7. RE-INVENTION

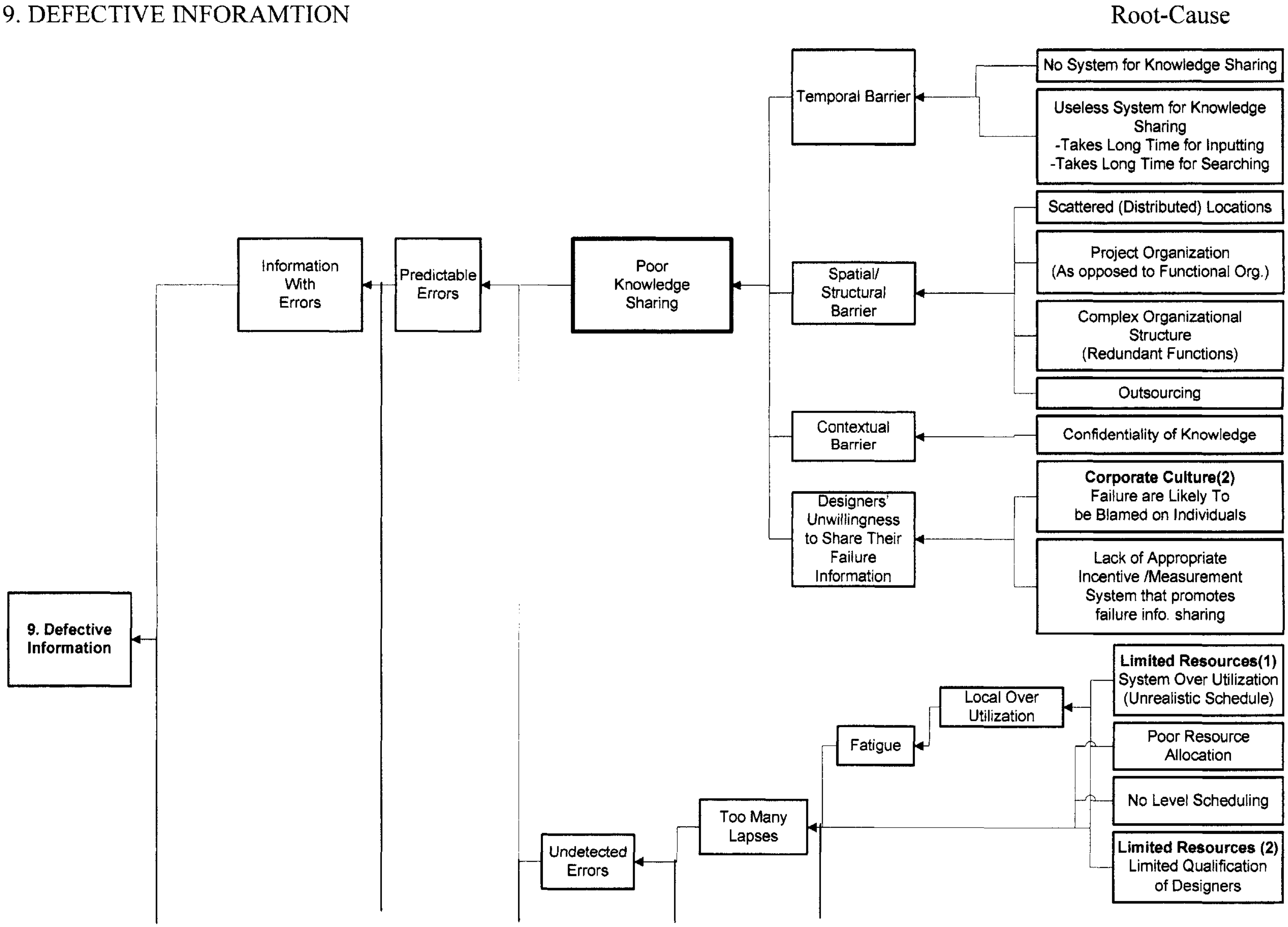
Root-Cause

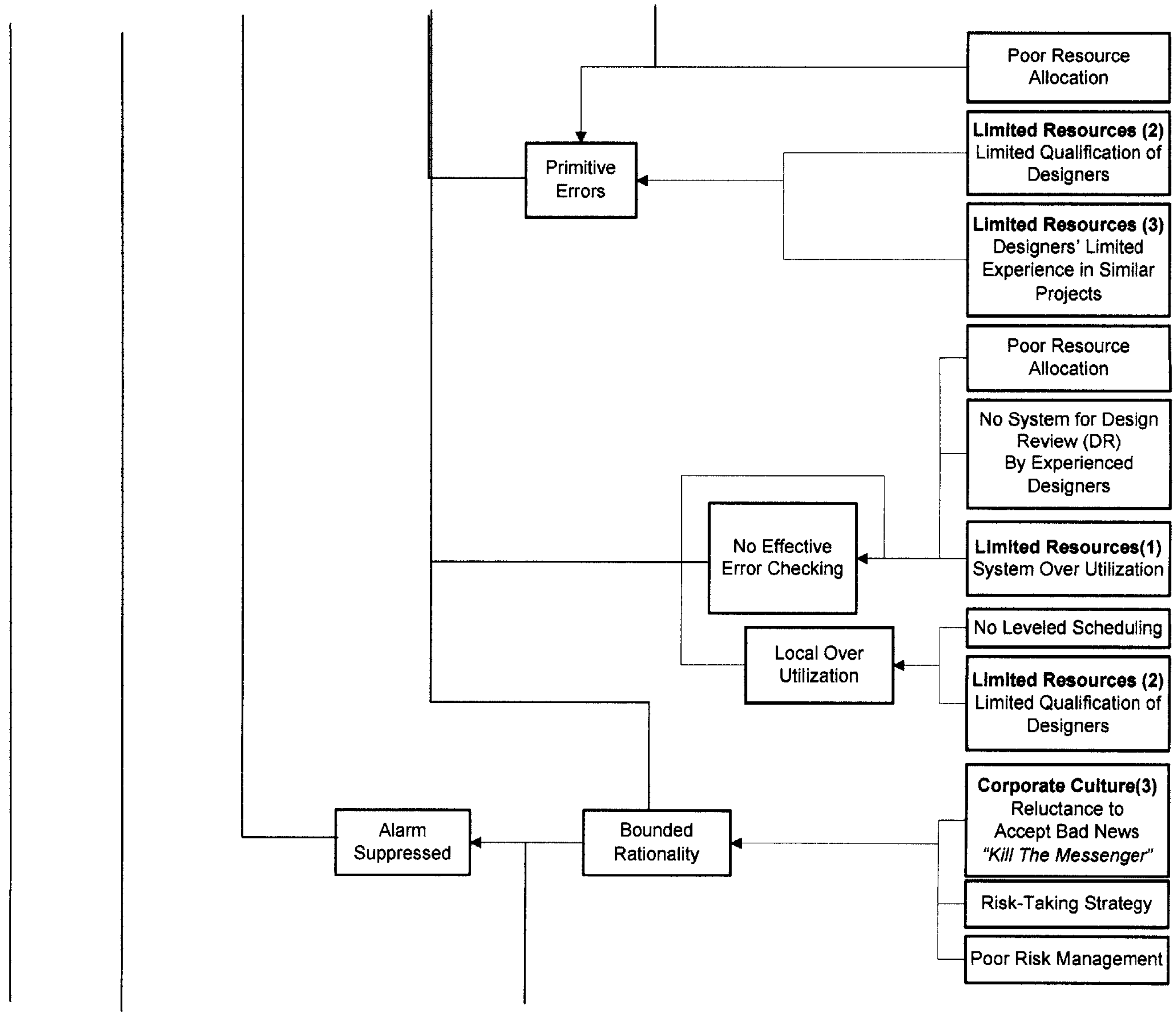


8. HAND-OFF

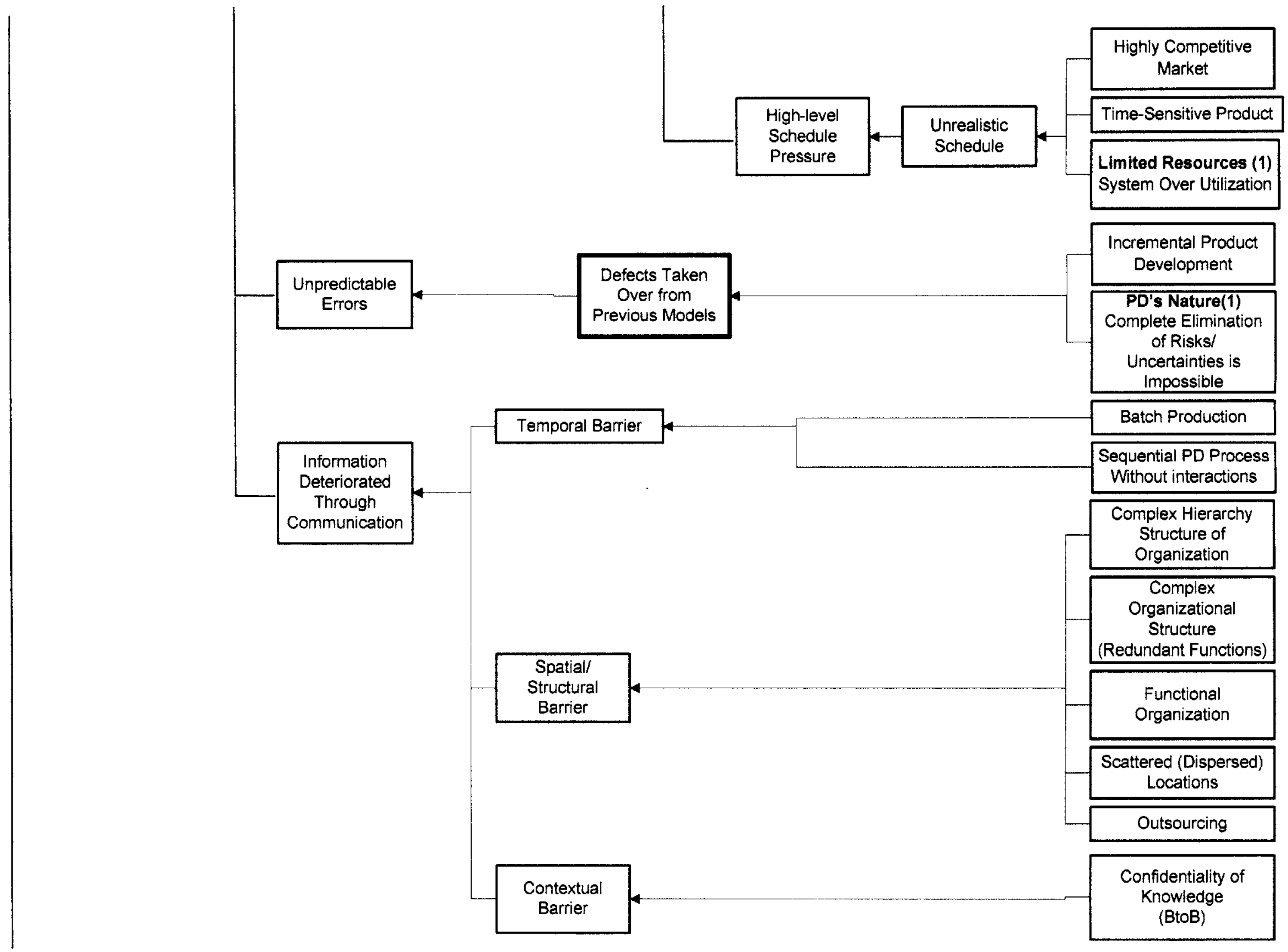

Root-Cause
8. Hand-Off
Scheduled Hand-Off
Unscheduled Hand-Off
Work Sharing
Dissemination of Requirements
Complex Product Design
Time-Sensitive Product
Intention to Level Workload
Outsourcing
Unloading Tasks on the Critical Path
Switching to an expert
PD's Nature(1) Complete Elimination of Risks/Uncertainties is Impossible
Limited Resources (2) Limited Qualifications of Designers
Absence of Task Owner

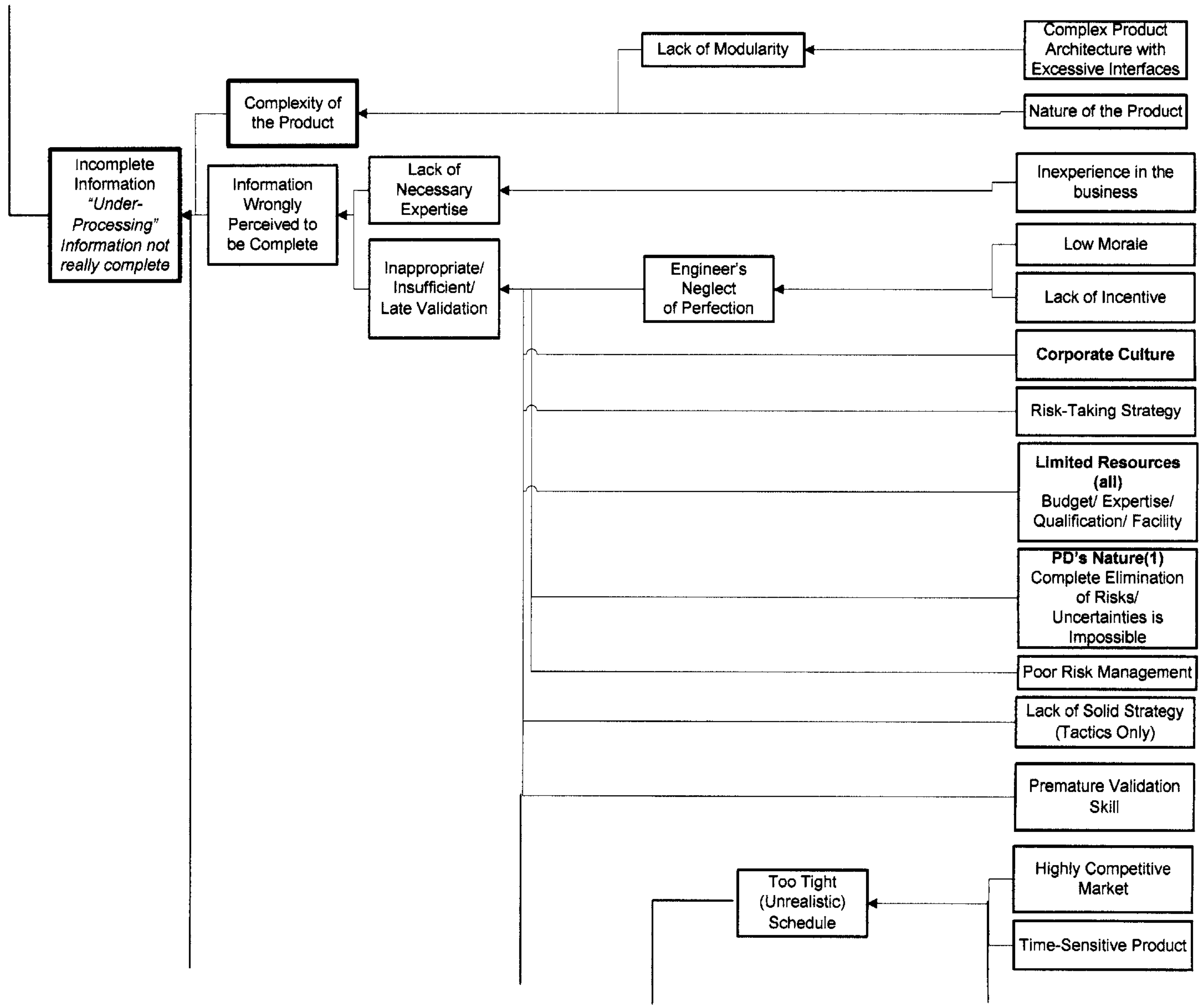
9. DEFECTIVE INFORAMTION



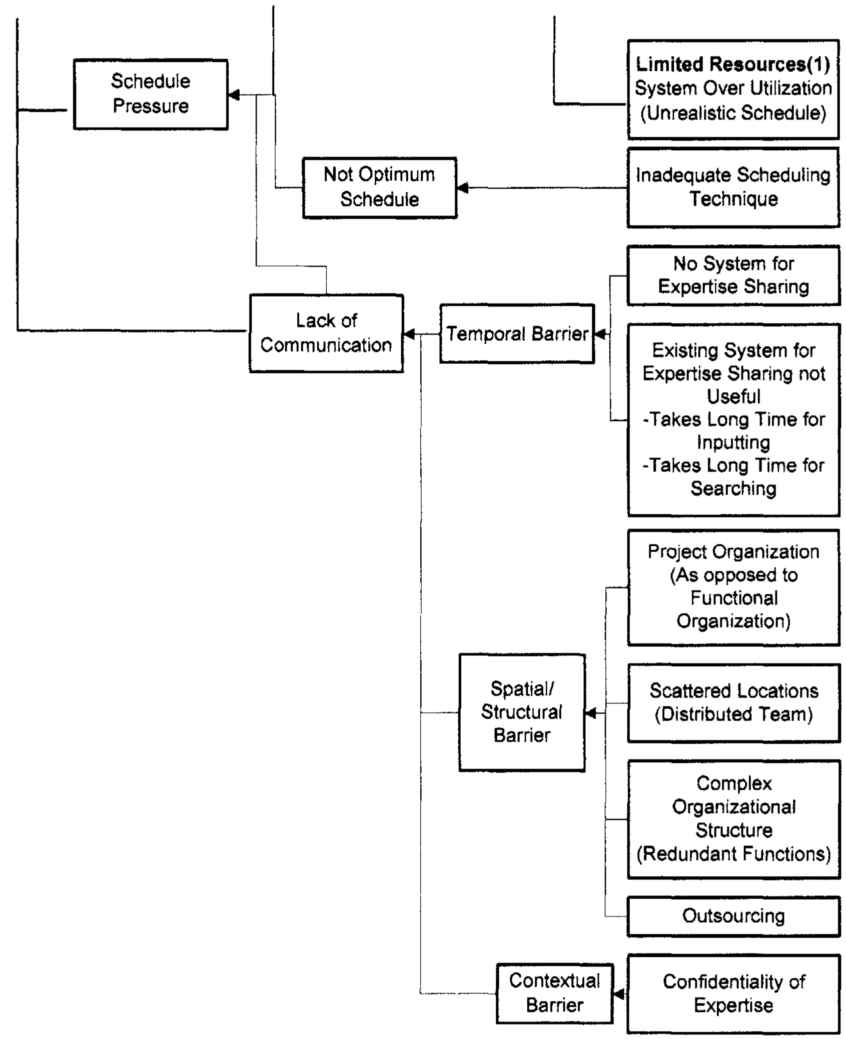


Poor Resource Allocation
Limited Resources (2) Limited Qualification of Designers
Limited Resources (3) Designers' Limited Experience in Similar Projects
Primitive Errors
Poor Resource Allocation
No System for Design Review (DR) By Experienced Designers
Limited Resources(1) System Over Utilization
No Effective Error Checking
No Leveled Scheduling
Limited Resources (2) Limited Qualification of Designers
Local Over Utilization
Corporate Culture(3) Reluctance to Accept Bad News "Kill The Messenger"
Risk-Taking Strategy
Poor Risk Management
Bounded Rationality
Alarm Suppressed

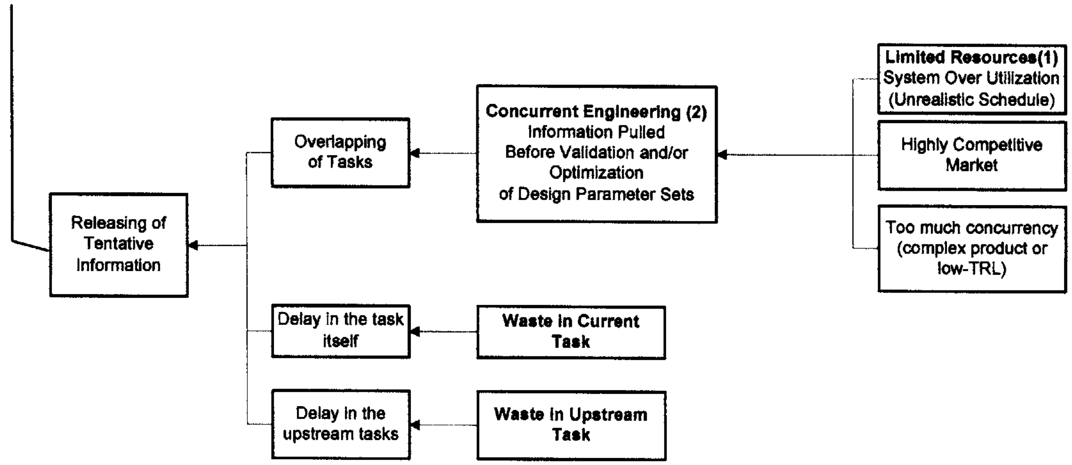

Highly Competitive Market
Time-Sensitive Product
Limited Resources (1) System Over Utilization
Unrealistic Schedule
High-level Schedule Pressure
Incremental Product Development
PD's Nature(1) Complete Elimination of Risks/ Uncertainties is Impossible
Defects Taken Over from Previous Models
Unpredictable Errors
Batch Production
Sequential PD Process Without interactions
Temporal Barrier
Information Deteriorated Through Communication
Complex Hierarchy Structure of Organization
Complex Organizational Structure (Redundant Functions)
Functional Organization
Scattered (Dispersed) Locations
Outsourcing
Spatial/ Structural Barrier
Confidentiality of Knowledge (BtoB)
Contextual Barrier



Complex Product Architecture with Excessive Interfaces
Lack of Modularity
Nature of the Product
Complexity of the Product
Incomplete Information "Under-Processing" Information not really complete
Information Wrongly Perceived to be Complete
Lack of Necessary Expertise
Inexperience in the business
Inappropriate/ Insufficient/ Late Validation
Engineer's Neglect of Perfection
Low Morale
Lack of Incentive
Corporate Culture
Risk-Taking Strategy
Limited Resources (all) Budget/ Expertise/ Qualification/ Facility
PD's Nature(1) Complete Elimination of Risks/ Uncertainties is Impossible
Poor Risk Management
Lack of Solid Strategy (Tactics Only)
Premature Validation Skill
Too Tight (Unrealistic) Schedule
Highly Competitive Market
Time-Sensitive Product



Schedule Pressure
Not Optimum Schedule
Lack of Communication
Temporal Barrier
Spatial/ Structural Barrier
Contextual Barrier
Limited Resources(1) System Over Utilization (Unrealistic Schedule)
Inadequate Scheduling Technique
No System for Expertise Sharing
Existing System for Expertise Sharing not Useful
-Takes Long Time for Inputting
-Takes Long Time for Searching
Project Organization (As opposed to Functional Organization)
Scattered Locations (Distributed Team)
Complex Organizational Structure (Redundant Functions)
Outsourcing
Confidentiality of Expertise



Releasing of Tentative Information
Overlapping of Tasks
Concurrent Engineering (2) Information Pulled Before Validation and/or Optimization of Design Parameter Sets
Limited Resources(1) System Over Utilization (Unrealistic Schedule)
Highly Competitive Market
Too much concurrency (complex product or low-TRL)
Delay in the task itself
Waste in Current Task
Delay in the upstream tasks
Waste in Upstream Task

BIBLIOGRAPHY

1. Allen, Thomas J. (1984), "Managing the Flow of Technology: Technology Transfer and the Dissemination of Technological Information within the R&D Organization," The MIT Press, Cambridge, MA.
2. Bauch, Christoph (2004), "Lean Product Development: Making Waste Trasparent," Diploma Thesis, Massachusetts Institute of Technology and Technical University of Munich.
3. Brooks, Frederick P. (1975), "The Mythical Man-Month," Addison-Wesley Pub. Co, Reading,
4. C.M. Creveling, J.L. Slutsky and D. Antis, Jr. (2000), "Design for Six Sigma in Technology and Product Development," Prentice Hall PTR, Upper Saddle River, N.J.
5. Fleisher, Mitchell and Jeffrey Liker (1997), "Concurrent Engineering Effectiveness : Integrating Product Development across Organizations," Hanser Gardner Publications, Cincinnati.
6. Forsberg, Kevin, Hal Mooz, and Howard Cotterman (1996), "Visualizing Project Management," John Wiley & Sons, Inc.
7. Graebsch, Martin (2005), "Information and Communication in Lean Product Development," Diploma Thesis, Massachusetts Institute of Technology and Technical University of Munich.
8. Goldratt, Eliyahu M. and Jeff Cox (1984), "The Goal," The North River Press, Great Barrington, MA.
9. Goldratt, Eliyahu M. (1997), "Critical Chain," The North River Press, Great Barrington, MA.
10. Kaplan, Robert S. and David P. Norton (1992), "The Balanced Scorecard – Measures that Drive Performance" Harvard Business Review, Jan-Feb 1992, Cambridge, MA.
11. Kruchten, Philippe (1999), "The Rational Unified Process," Addison-Wesley, Reading, MA.
12. Lyneis, James (1999), "System Dynamics for Market Forecasting and Structural Analysis," System Dynamics Review, Vol. 15-1, 2000
13. Lyneis, James (2000), "System Dynamics for Business Strategy: A Phased Approach," System Dynamics Review, Vol. 16-1, 2000
14. McManus, Hugh(2004), "Product Development Value Stream Mapping Manual, Beta Version," Lean Aerospace Initiative, MIT, Cambridge MA
15. Morgan, James M. (2002), High Performance Product Development: A Systems Approach to a Lean Product Development Process," PhD. Thesis in Industrial and Operations Engineering, The University of Michigan.
16. Murman, Earll, Thomas Allen, Kirkor Bozdogan, Joel Cutcher-Gershenfeld, Hugh McManus, Deborah Nightingale, Eric Rebentisch, Tom Shields, Fred Stahl, Myles Walton, Joyce

Warmkessel, Stanley Weiss, and Sheila Widnall (2002), "Lean enterprise value : insights from MIT's Lean Aerospace Initiative" Palgrave, New York

17. Ohno, Taiichi (1978), "Toyota Production System: Beyond Large-Scale Production," Productivity Press, Cambridge, MA.
18. Oppenheim, Bohdan W. (2004) "Lean Product Development Flow," System Engineering, Vol. 7-4, 2004.
19. Pich, Michael T., Christiph H. Loch, and Arnoud De Meyer (2002), "On Uncertainty, Ambiguity and Complexity in Project Management," Management Science, Vol. 48-8, 2002.
20. Repenning, Nelson P. (2001), "Understanding Fire Fighting in New Product Development," The Journal of Product Innovation Management, Vol. 18, 2001, pp. 285-300.
21. Shenhar Aaron J, Dov Dvir, Thomas Lechler, and Michael Poli (2003), "One Size Does Not Fit All"
22. Slack, Robert A. (1999), "The Application of Lean Principles to the Military aerospace Product Development Process," Master's Thesis in Engineering and Management, MIT, Cambridge, MA.
23. Soderloud, Jonas (2002), "Managing Complex development Projects: Arenas, Knowledge Processes and Time," R&D Management, Vol. 32-5, 2002.
24. Sterman, John D. (2000), "Business Dynamics" McGraw-Hill Higher Edication.
25. Stewart, Wendy E (2001)., "Balanced Scorecard for Projects," Project Management Journal, March, 2001.
26. Ward, Allen, Jeffrey K. Liker, John J. Christiano, and Sobek II, Durward K. (1995), "The Second Toyota Paradox: How Delaying Decisions Can Make Better Cars Faster," Sloan Management Review, Spring 1995, Cambridge, MA.
27. White, Diana and Joyce Fortune (2002), "Current practice in project management -- an empirical study," International Journal of Project Management, Vo. 20, 2002.
28. Womack, James P. and Daniel T. Jones, and Daniel Roos (1991), The Machine That Changed the World," Harper Perennial.
29. Womack, James P. and Daniel T. Jones (1996), "Lean Thinking," Simon & Schuster.