The Impact of the Geographic Distribution of Design Engineers on the Pace of Engineering Development

by
David A. Schiller

Submitted to the System Design & Management Program
In Partial Fulfillment of the Requirements for the Degree of

Master of Science in Engineering and Management

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2006

Signature of Author ______________________________

David A. Schiller
System Design and Management Program
June 2006

Certified By ______________________________

Dr. Ricardo Valerdi
Thesis Supervisor
Research Associate, Lean Aerospace Initiative

Certified By ______________________________

Patrick Hale
Director
System Design and Management Program

The Impact of the Geographic Distribution of Design Engineers on the Pace of Engineering Development

By

David A. Schiller

Submitted to the System Design & Management Program
On May 12, 2006 in Partial Fulfillment of the
Requirements for the Degree of Master of Science in
Engineering and Management

ABSTRACT

The increasing use of digital design tools and broadband information networks is creating an environment that permits the geographic distribution of design engineers. In order to successfully distributed engineering the consequences need to be understood. Through the examination of records of project execution, this thesis investigates whether the decision to geographically distribute engineers has a measurable impact on the pace of engineering development. A task-based Design Structure Matrix (DSM) was developed and showed that the projects studied were developed using a highly integral process. It is hypothesized the unanticipated consequences of distributing engineers geographically will slow the pace of engineering development to such an extent that costs incurred in protracted engineering development outweigh the benefits.

Three findings result from of this study. First, the geographic distribution of design engineers proved to have a negative affect on schedule performance causing distributed projects to overrun their schedules by more than twice as much as localized projects. Second, the development process for the systems studied was found to be highly iterative rather than adhering to the anticipated waterfall model espoused by the process documentation. Third, the level of task aggregation used to study this phenomenon affects the ability to identify the impact of distributed engineering.

Thesis Supervisor: Dr. Ricardo Valerdi
Title: Research Associate
Lean Aerospace Initiative

ACKNOWLEDGMENTS

I will always be grateful to my sponsor company for the opportunity to come to MIT. Particularly, I owe a debt of gratitude to George Perkins for his support throughout my career and to George Reynolds for his support while I've been at MIT. I also need to thank William H. Beck III for his enduring mentorship and friendship. My true education as a mechanical engineer happened through interactions with Bill. A number of people contributed to this work by taking the time to be interviewed and talk with me: Viguen Ter-Minassian, Michael Liu, Roy Kohn, Mike Merryman, Paul Kemlage, and Dr. John Lenard.

I've had the fortune of making and re-establishing some great friendships at MIT and through the SDM program. I will miss all of you when I leave. This thesis was expedited and enhanced because of the wisdom of Erica Fuchs and Matt LaMantia's kindred sense of how to battle procrastination.

I feel a pull of nostalgia for MIT. Jason Pinto and I have long conjectured that there are students that love their time here and students that endure their time here. I have loved almost every minute. I have found no other place in the world like MIT. I watch with pride and respect as my friends from this place move on to positions of responsibility and technology leadership throughout the world. Long may we roll.

Now, as always, I am overwhelmed with gratitude and love for my family and the example they have set for me. They have supported me without fail my entire life. They have inspired me to persist toward my dreams and focus on what is most important. I hope my parents never underestimate the value of their example to their children. Thank you. I love you.

per ardua ad astra

TABLE OF CONTENTS

TABLE OF FIGURES

CHAPTER 1: INTRODUCTION & MOTIVATION

The topic for this thesis was formed in my mind by on the job exposure to progress-hampering decisions to geographically distribute engineering development. One particular example I witnessed involved the development of an electronics chassis which was divided between engineering facilities on the east and west coasts of the United States – locations separated by over 3,000 miles and three time zones. Responsibility for the design of the chassis itself was given to engineers on the west coast while the design of the boards which populated that chassis was relegated to engineers on the east coast. Engineers involved with this project were baffled by the division of labor. Substantial iteration and information exchange occurs between chassis designers and board designers as the interfaces between the circuit cards and a chassis evolve. Those involved with this project have horror stories to tell about the difficulties of communicating with colleagues on the opposite coast.

A traditional understanding of the design process views chassis design and board design as two separated tasks. In schedules for the project the tasks are shown as being completed sequentially. This is a useful simplification for scheduling the engineering design process. However, the reality is that the engineering of these components happens simultaneously. Because of their interdependent nature there is no choice other than to engineer the components together. The simplification of thinking of tasks as sequential and simply dependent can mask important dynamics of information exchange. When engineers are co-located and can communicate informally these dynamics may not be important. However, when making decisions about how to divide or distribute engineering these dynamics can be crucial.

The project described above was over budget and behind schedule when it was completed. Even in the presence of the standard the impediments to smooth engineering development – i.e. undiscovered rework, schedule and scope changes, shifting management priorities – the counterintuitive division of labor on this project was blamed by the engineers for the bulk of the problems. This thesis attempts to measure the impact of the distribution of engineers on the pace of engineering development. Quantifying the phenomenon will enable better informed decisions about distribution to be made.

There are many reasons why Engineering Development may be geographically distributed; schedule pressures, work load balancing, and cost savings must be weighed in decisions about where to locate engineering. Friedman [15] and others argue that the time has come for the global distribution of engineering development. The ubiquity of internet access combined with tools to aid in the remote collaboration of engineers has created an environment where it is possible to imagine remote collaboration where before none was possible. Geographically distributed teams are gaining in prevalence and discussion of how to manage them is growing. [10, 14, 17, 20] While modular designs may be simpler to geographically distribute, managers of monolithic projects must make difficult decisions about how to distribute engineering. As described by Tom Allen [1], the bandwidth of remote communication technologies is lower than the bandwidth of face to face communication. Certain types of ideas are best communicated when people are face to face. It is important to understand the impact on the pace of development before distribution decisions are made.

RESEARCH QUESTION & HYPOTHESIS

Through the examination of records of project execution, this thesis investigates whether the decision to geographically distribute engineers has a significant impact on the pace of engineering development. The specific research question is:

> *What is the schedule impact of geographically distributing engineering of design-intensive products with integral architectures when compared with localized engineering.*

This research question is studied with data collected from the engineering development of electronic circuit cards (ECAs) for space flight applications. A task-based Design Structure Matrix (DSM) was constructed to map the information relationships between the engineering development tasks. The DSM showed that the development process for ECAs is highly iterative and departs from the anticipated waterfall development model espoused by engineering process documentation. Examining the DSM shows that ECAs have an integral architecture and development process. The design tasks on the path of ECA development are interdependent and cannot be accomplished without information exchange between engineers. There are no clean interfaces which can be leveraged to develop the product in a modular fashion.

Based on the nature of information exchange between development engineers working on the design of these hardware systems, I hypothesize that:

> *The unanticipated consequences of distributing engineers geographically will slow the pace of engineering development to such an extent that costs incurred in protracted engineering development outweigh the benefits.*

Design issues which must be resolved through the collaboration of multiple engineers will proceed more slowly when using teleconferences and email because these media lack the richness of face to face communication. [1] While email and teleconferencing enable remote communication, they are poorly suited to discussions of complex spatial ideas and are more error prone and slower to provide feedback than face to face conversation.

Data was collected and analyzed at two levels of detail. The first, and more finely detailed study used low-level records of the occurrence of design updates for 25 design projects. The second level of detail was more coarse and used higher-level data typically available to program managers. There were six designs for this part of the study and they were a subset 25 design projects studied in fine detail. Review of the detailed measures of project progress – specifically the rate of circuit board design updates – does not show any significant impact of the geographic distribution of engineers. At a macro level, data retrieved from project schedules show a marked difference between the two groups. Distributed development takes substantially longer as a fraction of the time estimated to complete a project than localized development. That the impact of distribution is seen in the macro level data but not in the data of finer scale engineering indicates that the level of task aggregation used to study this phenomenon impacts the ability to identify trends.

BACKGROUND & SCOPE OF WORK

Incentives to Geographically Distribute Engineering Development

Even if we stipulate that the geographic distribution of design engineers slows the pace of engineering development, there may still be reasons why firms would pursue this practice. There may be economic advantages to the use of labor from a certain geographic location that offsets the slower pace of development. A company may replicate engineering competencies in various geographic locations through acquisition or growth. Engineers with overlapping skills in separate locations may encourage geographic distribution through efforts to level the engineering workload between locations.

There may also be intangible incentives to distribute engineering tasks having to do with synergies with other projects or the individual expertise of engineers. Geographically distributing development projects may expose engineering offices to practices and procedures from other regions. Distributing engineering development could be an instrument of disseminating corporate culture.

It is possible to imagine a host of rational incentives for geographically distributing engineering development. In some instances it may be a necessity of operating the business. One goal of this thesis is to make some data based observations about the unforeseen consequences of distributed engineering development.

Defining Engineering Development

This thesis will focus on the Engineering Development (ED) portion of the product development process (PDP). In the widest sense of the term, the PDP describes all stages of a product's lifecycle. It includes all of the following: initial concept development, engineering design, manufacturing, product operation and maintenance, and disposal. All of these tasks are part of the whole system PDP. [8] This thesis studies only the Engineering Development portion of the product development process because this portion of the development process is the most communication intensive and therefore the most affected by geographic distribution.

Engineering Development will be taken to mean all of the steps which occur between the award of a contract for development of a piece of hardware and the release of engineering drawings to production. The steps which precede engineering development involve the gathering of user needs and the synthesis of a product concept. When engineering development is complete the product is ready for manufacture. These three steps are illustrated in Figure 1–1.

The boundaries of engineering development are not always clear cut. In aerospace projects the nature of firm and customer interaction is somewhat reversed from what is typically observed in the development of consumer goods. Unlike consumer goods where firms develop product concepts in response to perceived user needs, in typical aerospace programs the initial concept development is performed by customer, sometimes in concert with the potential contractors and consultants. The initial concept is used to generate a request for proposals from potential contractors. The aerospace firm which wins the contract uses the preliminary product concept as a starting point to complete concept development and engineering. Because of this relationship, when a contractor begins an engineering project, the engineers are not beginning from a blank sheet of paper. In some sense, the concept for what is to be created has already been formulated and stated in the product specification. This does not mean that the architecture of the product is predetermined. The design process still evolves through a series of concept, preliminary, and detailed design steps. However, this relationship blurs the boundary of engineering development at the beginning of a project because responsibilities are shared between the customer and the contractor.

Similarly, at the end of engineering development there is a gray area between engineering and manufacturing. Modern engineering practice has embraced the notion that design engineers cannot divorce themselves from a project after the engineering is nominally complete [25] The line between engineering and manufacturing is blurred in two ways. First, manufacturing engineers are involved in the engineering development of a product as early in the process as is practical. Second, design engineers work closely with manufacturing to solve the problems which arise during manufacturing. While acknowledging the fact that engineering continues after manufacturing has begun, this thesis studies only the engineering that occurs before drawings are first released to manufacturing.

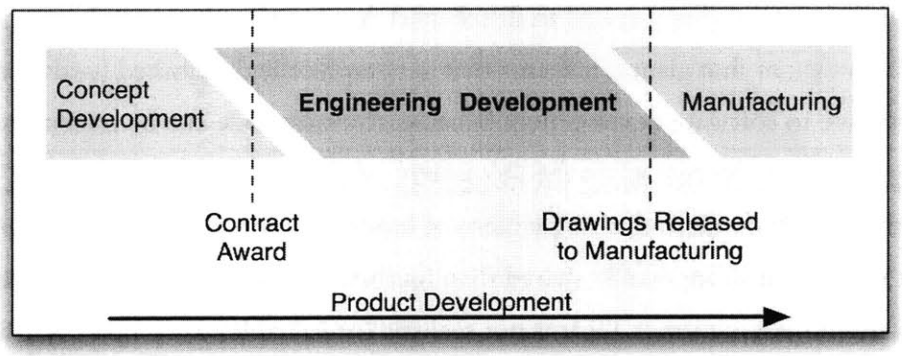


Figure 1–1: Diagram showing the relationship between engineering development and the adjacent stages of development. This thesis studies the engineering tasks between the award of a contract and the release of drawings to manufacturing.

The decision to limit the scope of research to engineering completed before the start of manufacturing was made so that the most information intensive portion of the development process was the focus of study. The sources of data used in this thesis break the product development process into engineering and manufacturing stages. Mirroring this break is process documentation with matching boundaries in the thesis facilitated the gathering of data. This thesis is not a study of the outsourcing or off-shoring of manufacturing, the products studied in the thesis were assembled in-house from pieces purchased or manufactured domestically or even locally. See Fuchs *et al.* [16] for a discussion of the impacts of outsourcing manufacturing on engineering development.

Hardware Development Compared and Contrasted with Software Development

This thesis studies electronic circuit card assemblies. These are electromechanical systems in the sense that they involve electrical and mechanical engineering. The primary function of these systems is to perform some electrical job. The proper functioning of the electrical circuitry is dependent not only on electrical engineering, but also on mechanical and software components. The bulk of the engineering development time is spent of the design of hardware. While software engineering is required as part of system development, the hardware engineering throttles the pace of development. Software is rarely on the critical path. The ECAs studied here were approximately 95% hardware effort and 5% software effort on average.

It is interesting to compare the ECAs studied here to software development projects because of the large number of examples of distributed software development. Large-scale software development projects are being successfully developed using the open source software model. Typically open source software developers do not even work for the same company, much less interact face to face. [21] Software design seems ideally suited to distributed development because the product is not a physical good. In fact, at first glance, it seems that geographically distributed hardware development is hampered relative to software development because software code can be transmitted and shared geographically for almost no cost while the movement of physical goods is expensive. Furthermore while software is readily duplicated, a single piece of hardware cannot exist in two locations at once. The view that software is more readily shared than hardware mixes metaphors in a manner that is surprisingly common but incorrect. [3] It is not realistic (or valuable) to compare software development with the *manufacture* of physical goods. It is much more reasonable to compare software development with the *engineering development* portion of hardware development.

Viewed from this perspective, the prospect of distributed hardware development is not hampered by the expense of shipping or the impossibility of sharing physical goods. The majority of hardware development happens long before there is ever a physical artifact to worry about. For modern products the majority of the design is developed as digital files in one of many design tools. Physical parts are described in model files in CAD programs. Thermal analyses exist as digitized matrices describing temperatures throughout a part. The list goes on and on. Everything from structural analyses, to schematics of electrical circuits, to simulations of RF performance exist as readily transferable digital files. [4] The barrier to the geographic distribution of hardware development is not that the design cannot be readily shared or duplicated. The reasons are more subtle.

One difficulty in the development of hardware is the lack of interoperability between the various design tools. Not only are engineers separated into disciplines (mechanical, electrical, software), often the tools used by engineers are so specialized that an individual will spend years becoming proficient in one piece of the development process (board designers, packaging engineers, thermal or structural analysts). In practice, the mechanical engineer does not know how to operate the electrical engineer's schematic tools and the electrical engineer does not know how to operate the mechanical engineer's CAD tools. Therefore, just because an engineer is able to transmit a digital file does not mean that it can be understood by the recipient. Typically meta-communication about the ideas in the digital file takes place so that engineers of all disciplines can perform their jobs.

End-to-end product development of hardware systems involves many stages. However, it is only during product development that geographical distribution of participants can be achieved without incurring transportation costs. In contrast, once a product enters the manufacturing stages of development, the materials involved must be physically transported if development is to continue across locations. This is another reason the scope of this work is limited to engineering design.

Distributed Engineering Development Compared to Subcontracting

The degree to which geographic distribution will impact the pace of engineering development depends on the degree to which a given product architecture is modular. [23] If the project can be divided into subsystems with well established and clean interfaces then the problem of communication is limited to documenting and controlling the interfaces. This is a key characteristic of a modular architecture which makes products with this style of architecture amenable to distributed development. [2, 21] Modular architectures are the norm in systems which subcontract some portion of engineering development. When engineering development is subcontracted interfaces must be defined in advance of the subcontracted system being fully designed. A trade-off is made when specifying these interfaces in that they place constraints on the system. This may result in suboptimal performance.

Where modularization is possible and desirable it is often useful. Modularizing facilitates the decomposition of systems into manageable pieces which is one tool for dealing with the complexity of large systems. However, some systems are not amenable to modularization. [26] Further, there is research which suggests that too much modularity may limit a firm's flexibility. [7, 13]

For products with an integral architecture it is often not possible to subcontract engineering projects because interfaces can not be cleanly defined. In these instances all portions of a design are in flux until the entire design is completed. The ECAs studied for this thesis have an integral architecture and as a consequence an integral design process. Because the design is integral it is not possible to cleanly separate a portion of the design process. Any work which involves engineers distributed from the rest of the team will require extensive communication between the two groups.

DESIGN STRUCTURE MATRICES IN THE LITERATURE

The DSM developed for this thesis builds on published techniques for the study and analysis of engineering processes. [5, 6, 9, 11, 12, 18, 19] The DSM which resulted from the analysis of the ECA development process shares many common characteristics with previously developed DSMs. [22, 27-29]

CHAPTER 2: CASE STUDY SELECTION AND DESCRIPTION

All data used in this thesis was collected from the records of a single aerospace firm. The firm employs thousands of engineers that work on a range of products. The product area of focus for this thesis is electronic systems produced for spaceflight applications. This type of engineering is commonly referred to as electronics packaging. The rigorous environmental demands placed on electronics hardware by the space environment necessitate detailed engineering analysis of every aspect of a design. In addition to ensuring that the electrical functionality of the device meet the requirements of the application, it also necessary to choose materials that are compatible with the high radiation and vacuum environment. The design must account for the vibration and shock loads endured during launch, the on-orbit temperature extremes, and the lack of convection cooling to carry away heat. Further, because of the high cost of lifting hardware to orbit, the customer places commensurately high reliability requirements on the design.

SELECTING ECAS AS THE UNIT OF STUDY

The unit of study ultimately selected for this thesis is the Electronic Card Assembly (ECA). The following section describes the process by which this selection was made.

Four criteria were established to evaluate candidate projects for study. The first of these was that the project had to be sufficiently large to support distributed engineering. Systems which are too simple do not involve enough engineers to warrant distributed engineering. In studying the development of hardware for spaceflight applications it was observed that the engineering rigor involved emphasizes the multidisciplinary nature of engineering development. Even if the delivered electrical function of an ECA is straightforward and only one engineer is required for that portion of the design, the talents of engineers from various disciplines – electrical engineering, mechanical engineering, thermal-structural analysis, materials engineering, software engineering – must be leveraged to design a successful product. Because of the disparate engineering disciplines required for each design, it is the case that even relatively simple projects require a sizeable team of engineers. ECAs are typically developed by a team that ranges in size from 10 to 15 people. This team size is sufficient for geographically distributed development. Simply stated, this first criterion was that the project cannot be too small.

The second criterion was that the project cannot be too large. When engineering projects grow beyond a certain size their development process becomes so complex that the process is effectively never repeated. This requirement prevented the use of the entire electronics chassis as the unit of study for this thesis. The difficulty with electronics chassis is that each one is in some way different from the next. Each design has distinct challenges which alter the path and substance of the design process. The result of this heterogeneity is that any process description which is granular enough to be used for all chassis is too granular to be useful for analyzing the progress of any particular design. The process steps are so high-level that the interesting dynamics are masked.

The ECAs studied for this thesis performed a variety of functions. A sampling of the delivered functionality includes controlling the deployment of spacecraft solar cells, controlling the detonation of the explosive bolts used to separate sections of a spacecraft, battery monitoring and control, solar array power management, and command and data handling, to name a few. Even with the widely varying delivered functionality of these ECAs, the design procedure for any single ECA is largely the same as the generic procedure. Engineering analyses performed during design drive the engineering solution down different paths and the final designs of the ECAs are distinct. However, the steps that are taken and the order in which they are performed are largely the same for every ECA.

The third criterion for selecting projects was that the development process in question had been executed enough times to make a reasonable study. ECAs satisfy this requirement. The large number of designs available for study is the result of two factors. First, ECAs are the modular unit of functionality for electronics design so there are many of them. Second, as mentioned above, the design process for each design is standardized. Experience with many ECA designs has advanced the engineering firm up the learning curve to the point where the process is consistent for each design.

The fourth and final criterion used to select projects for study was that they have been developed by both localized and distributed teams. Discussions with engineers revealed many projects which had been developed by distributed engineering teams. However, few of these projects had an analogous project developed by a localized team. The purpose of restricting this thesis to the study of projects which had been developed by both localized and distributed teams was to observe the impact of one variable while maintaining other parameters affecting the pace of design constant. This helped single out the impact of geographic distribution for study.

ECAs fulfill all four of the high-level criteria. An additional, less tangible reason for their selection was that I had tacit knowledge of the intricacies of the design process gained through experience designing and managing the design of these systems.

Figure 2–1 illustrates the relationship between ECAs and an electronics chassis. Typically an electronic chassis is a deliverable system. It is the device a customer interacts with. The ECAs are modules within the chassis which provide each required electrical function. Like the average person interacts with a personal computer, the aerospace customer typically handles and makes connections with the chassis as a whole. It is unusual for a customer to have cause to interact with or handle the circuit boards inside. In the same way most consumers never interact with the circuit boards inside of their personal computers.

If a contract requires that multiple functions be performed by the delivered system, it is common to find each function or group of functions allocated to an individual ECA. Each ECA is a design project unto itself.

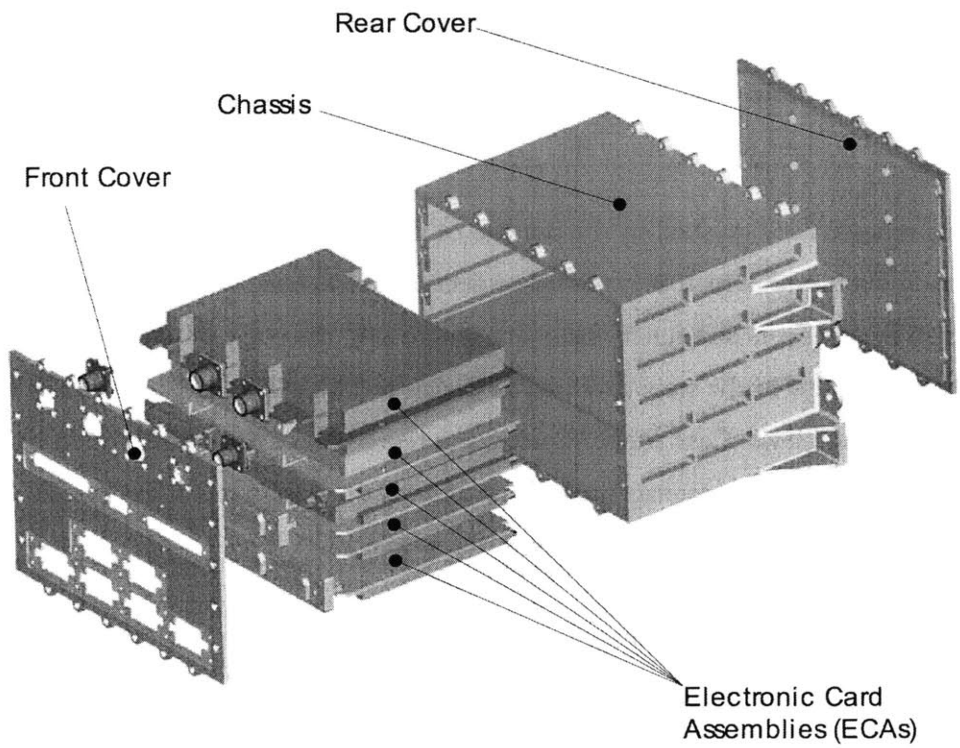


Figure 2–1: This exploded view of an electronics chassis shows the elements which make up a typical system. All of the elements taken together form a delivered product. Each ECA can in itself be considered an engineering project.

The ECAs themselves consists of distinct subsystems. The predominant subsystem is the circuit board which interconnects all of the discreet electrical components. The circuit board is attached to a heatsink which carries away the heat generated by the electrical components and provides a mechanical structure to support the circuit board. The heatsink typically incorporates features which are used to secure the card in the chassis. In addition to discreet electrical components which are soldered to the circuit board there are also electrical connectors which have an obvious electrical function but which must also be integrated into the mechanical design. The connectors comprise an important electrical *and* mechanical interface. Finally, some ECAs incorporate processing units which require software as part of the total design.

DATA SETS AVAILABLE FOR STUDY

The data documenting ECA development comes from two sources and is separated into two data sets. The first source is detailed records of design iterations kept by the lead designer in a circuit board design group. These data are a record of the day to day events which comprise a portion of the engineering of 25 ECAs. The data extracted from these records constitute case study #1. The second source of data for this thesis is a collection of project schedules used to track the progress of six ECAs. Data extracted from these schedules make up case study #2. The six ECAs in case study #2 are a subset of the designs in case study #1.

The two data sets are distinguished by the portion of the ECA engineering process they document and the level of detail with which they record project progress. Case study #1 is made up of data that would be used by engineers and functional managers to take the pulse of day to day project progress. It documents only a few of the steps required to design an ECA. The data is low-level and recorded at the engineering task level. Case study #2 uses data recorded for the consumption of individuals whose perspective is program management. It condenses the design of each entire ECA into a single metric. This data is high-level and is recorded at the whole-ECA level.

Description of Case Study #1

Data for the first case study is taken from the design records of a group of circuit board designers. An ECA's primary function is to perform some electrical task and the completed circuit board is the assembly that fulfills that role. The circuit board not only serves as the mounting surface for a majority of the electrical components, the traces inside the circuit board form the electrical interconnection between the components. Consequently, circuit board designers have a central role in the ECA development process.

The nominal sequence of steps executed by the circuit board design group is shown in Figure 2–2. For each step, the lead designer in the group made a record noting the occurrence and the date every time a design iteration was completed. The resulting record contains information about the number and frequency of design iterations for each step in the circuit board design process. Records kept for component placement (step 3) are of particular interest for this thesis. For each placement, data is first transferred from circuit board design to mechanical and electrical engineering. The electrical and mechanical engineers subsequently review the placement and give comments back to the board designer. Coordination between these three groups is required to finalize the placement.

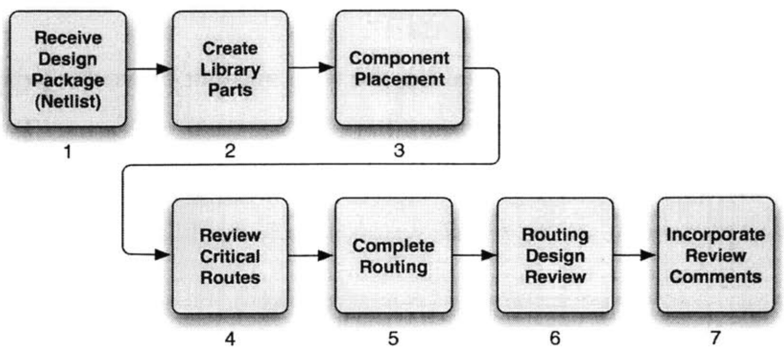


Figure 2–2: Diagram showing the steps involved in the circuit board design process as documented by the circuit board design group. It is interesting to note that the steps are represented as occurring sequentially, yet the data collected by the group shows that iteration between steps is the norm.

Records of these interactions were available for a 25 ECA designs. These 25 designs represent approximately two years of engineering activity. The ECAs included in this data set represent designs from both localized and distributed teams.

These exchanges of information are not formal enough to warrant a design review with detailed minutes and lists of attendees. They are significant enough to prompt the design group to make a note of their occurrence. These records of information exchange document the rate of collaborative engineering. The records do not provide any insight into the health of the project or its adherence to schedule. Case study #1 is used to study the pace of engineering iteration by observing the interactions between engineering departments. The detailed progress reports provide a record of engineering "in the small." By studying this data it is possible to make inferences about the impact of distributed development on particular interactions and portions of the design.

Gathering Information for Case Study #1

For each the 25 ECA designs the number of component placements as well as the date each component placement was completed was recorded. The placement of components can affect estimates of electrical performance, structural and thermal margin, and reliability. Consequently, numerous engineers are involved in the review of the component placement. The number of placement iterations is often a function of the complexity of a design. Simple designs may have only one or two placement revisions. Complicated designs may have as many as 15. Because the absolute number of placement iterations is as much a function of board complexity as anything else, it cannot in itself be used to assess the impact of the geographic distribution of design engineers. The rate of iterations is calculated and used for comparison to account for the differences in complexity.

The rate of design iteration is calculated by taking the total number of working days required to finalize a placement and dividing by the total number of placement iterations. This metric provides some measure of the pace of development. Numerous engineering departments are involved in developing a finalized placement so this metric seems a likely candidate to reveal changes in the pace of design.

The Nature of Distributed Engineering in Case Study #1

The ECAs which make up this data set represent a range of geographic distribution. Of the population of designs that was studied, six were developed for one project which shall be referred to as Project A. Of the six ECAs developed for Project A, three were developed by localized teams and three were developed by distributed teams. The localized teams were all working in the same building or campus of buildings. The engineers on localized teams were within walking distance of each other and could meet face to face whenever the need arose.

The distribution of engineering development on Project A occurred in the following manner. At the beginning of the program, the detailed electrical engineering was assigned to an offsite group of engineers approximately 30 miles from the primary engineering facility. Furthermore, the circuit board layout and routing was assigned to a group approximately 50 miles from the primary engineering facility. All other design tasks for these ECAs were undertaken by engineers in the primary facility.

As the project progressed and the detailed electrical design was nominally finished, the engineers who had been doing the work moved on to other projects. Responsibility for the electrical engineering was given to engineers in the primary facility. A similar pattern was followed for the circuit board layout. Once the design was nominally finished, responsibility for board layout was transferred to engineers at the primary facility. In both cases, even though the design was nominally complete, substantial work was done by the engineers in the primary facility on tasks that were originally assigned to engineers in remote facilities.

There was one further example of distributed engineering in the data collected for case study #2. The electrical engineering for one ECA was performed by an engineer who tele-commuted from Florida. For this design the mechanical and board design functions were located in the primary engineering facility.

The remaining designs in this population were designed by teams that were either localized or separated by distances which were small enough to allow face to face communication. Engineers that worked in the same group and met regularly through group meetings were considered to be essentially co-located even though they may not necessarily work in the same building.

Description of Case Study #2

Case study #2 comes from the schedule records for the development of the six Project A ECAs. All were developed during the same time period. These ECAs are a subset of the designs in case study #1. As detailed above, three Project A ECAs involved the geographic distribution of the mechanical, electrical, and board design functions. Each engineering group was separated by 30 to 50 miles.

Data was available for this project because the management team regularly updated a number of reports which tracked the progress of the program at regular intervals. Saved copies of these reports were available for study. Paramount among these reports was an Integrated Master Schedule (IMS). The IMS contains dependency, overall duration, and progress information for all of the tasks which are planned for the development of the project. The IMS also provides a baseline estimate of the time required to complete each task which was established at the beginning of the program. By studying changes made to the IMS it is possible to observe how the estimate of the time required to complete any particular task or group of tasks evolved over time.

The data in the IMS is useful for studying engineering "in the large." It records what happened but contains no record as to why. The tasks listed in the IMS may correspond to more than one individual engineering task. Consequently, each IMS task may involve engineers from a variety of disciplines as well as locations. It is known which ECAs were developed by distributed teams, so the design durations for the ECAs can be used to study the impact of distribution on the pace of ECA design. However, case study #2 provides no insight into the dynamics of project execution which may have impacted the pace of project development.

Gathering Information for Case Study #2

The IMS contained metatasks which correspond to the engineering design of the six ECAs. Each engineering design metatask was composed of several subtasks. Each subtask in the IMS equates to more than one of the subtasks which make up the DSM. In other words, the IMS is less granular than the DSM

The Project A management team saved a version of the IMS approximately weekly for the duration of the project. For each saved version of the schedule the following data were recorded: the duration of each ECA engineering design task, the percent complete of each ECA design metatasks, the duration of each circuit board design task, and the percentage complete of each circuit board design. This data was tabulated and analyzed for trends. The complete data table is presented in Appendix C.

CHAPTER 3: DEVELOPING THE DSM

The design process for the development of ECAs was studied and documented in a task-based DSM in the manner described in previous publications on this topic. [5, 27-29] The purpose of developing this DSM was to gain insight into the patterns of communication involved in the development of ECAs. The task-based DSM makes a record of the information connections between all of the tasks required for the development of an ECA. The following sections will document the process by which the DSM was constructed. The terms *stage of design* and *design task* will used throughout. In this context, the stage of design refers to the macro level design stages found in the process documentation. These are the waterfall design stages. Smaller design tasks comprise the stages of design. These tasks are the atomic level of process documentation in this thesis.

DATA SOURCES USED FOR DEVELOPING THE DSM

Process Documentation

The information foundation for the development of the DSM was process documentation. The aerospace company studied for this thesis has a large library of documents which describe the steps involved in designing ECAs. All available ECA process documentation was collected and analyzed. The process documentation as written divides the overall process description along two dimensions. The documentation is divided by process stage and by engineering discipline.

The first of these divisions represents the understanding of the development process by the authors of the documentation. The reigning paradigm of engineering development in use at the aerospace company is a waterfall development model. This is evident in the linked-list nature of meta-development tasks – or stages of development. They are shown in Figure 3–1. The second division, occurs because process documentation tends to be written by engineers for engineers. It is not written for insights into the overall process of engineering development. Consequently, each design stage tends to have a separate process document for each engineering discipline – in this case mechanical engineering, electrical engineering, circuit board design, and systems engineering each have their own process documents.

In order to create a DSM representing the complete process these two divisions had to be undone. The re-combination of the process documentation to create a single DSM is described later in this chapter.

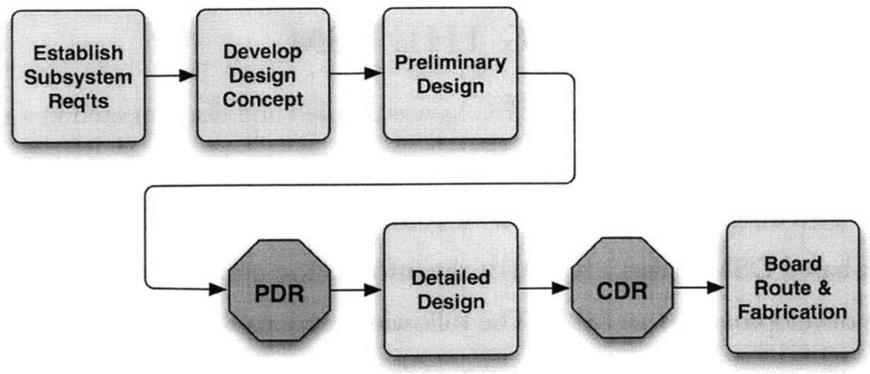


Figure 3–1: Diagram of the stages of design for the development of ECAs adapted from process documentation. There are five design stages and two formal design reviews in the process.

Semi-Structured Interviews

The process documentation describes the espoused process used to develop an ECA. However, the process-in-use by engineers may differ considerably from the espoused process. In order to gain an understanding of the process-in-use, engineers were engaged in semi-structured interviews. These interviews focused on discussions of the espoused process, covering both its shortcomings and the ways in which it is adequate. Interviews were conducted with engineers from multiple disciplines to get varied perspectives. Specifically interviews were conducted with representative from the following departments: electrical engineering, mechanical engineering, mechanical analysis, circuit board design, and software engineering.

The interviews resulted in minor changes to the list of tasks and the order in which they occur. The tasks listed in the process documentation encompass the vast majority of the design tasks the engineers perform. Furthermore, the order of the tasks was not much changed through the process of interviewing the engineers. The tasks in the espoused process are largely the same tasks identified by engineers when describing what they actually do to complete a project. The aspect of the DSM which was most changed by conversations with engineers was the information connections between tasks.

CONSTRUCTING THE DSM

Developing Mini-DSMs

As mentioned above, the process documentation makes two simplifying assumptions which had to be undone before the DSM could be constructed. The first assumption made by the authors of the process documentation was that the processes for each engineering discipline could be written separately. For example, the process documentation for electrical engineers is a separate document from the process for mechanical engineers. To capture the interactions between the multiple engineering disciplines involved with the design it was necessary to combine multiple engineering domain specific process documents into single, multi-domain process.

This was accomplished by extracting the tasks from the process documentation and concatenating them into a single list of tasks for each stage of design. In most instances the task descriptions were identical for each engineering discipline. Where unique tasks existed they were added to the task list for that stage of design. An example of a combined task list is shown with it's accompanying mini-DSM in Figure 3–2. This early stage mini-DSM shows only the information connections explicitly mentioned in the process documentation.

		1	2	3	4	5	6	7	8	9	10	11	12	13
Review Design Guidelines	1	•												
Develop Detailed Block Diagram	2	X	•	X	X	X								
Develop Prelimary HW ICD	3	X	X	•	X									
Develop Prelimary SW ICD	4	X	X	X	•									
Refine Test Strategy	5	X	X	X	X	•								
Review CAD Guidelines	6	X	X				•							
Develop Preliminary Parts List	7		X	X	X	X	X	•						
Develop Library Parts (Footprints)	8							X	•					
Complete Preliminary Schematic	9		X	X	X	X	X		X	•	X			
Complete Preliminary Component Layout	10			X		X	X		X	X	•			
Complete Preliminary Themal/Power Est.	11		X				X					•		
Submit Preliminary CAD BOM to ISE	12						X		X	X			•	
Update Req'ts Matrix/Risk & Cost Assessment	13											X	X	•

Figure 3–2: Mini-DSM representing the preliminary design stage of the process flow. Boxes which are marked with an X indicate information flow required between tasks. This DSM documents information connections which are explicitly mentioned in the ECA development documentation.

The mini-DSM shown above represents only the espoused information connections between design tasks. In order to properly represent the design process it was necessary to capture the actual information dependencies for ECA design. This was accomplished by showing the mini-DSMs to design engineers so that the tasks could be reviewed and information connections could be added or removed as necessary. Each engineer was shown all seven of the mini-DSMs and asked for comments.

When discussing information dependencies with engineers it was necessary to establish likelihood threshold to aid in deciding which dependencies rose to the level of being documented and which dependencies were too unlikely to be important. The guideline given to engineers was to only indicate an information dependency in the DSM if there was a substantial probability that one portion of a design would impact another. A one-in-four likelihood was used as a rough guideline. It was necessary to establish this threshold because at some level nearly every aspect of a design impacts every other. If even the most unlikely of interactions was documented on the DSM, the matrix would be completely populated rather than sparsely populated. A fully populated DSM provides no insight in to the design process. The likelihood threshold acts as a filter to help maintain a high signal to noise ratio in the DSM.

The mini-DSM representing the espoused process and the process in use for the preliminary stage of design is shown in Figure 3–3. Using the procedures described above mini-DSMs were constructed for all seven stages of the development process. All of the mini-DSMs are presented in Appendix A:. It is interesting to note that the engineers were aware of many undocumented information dependencies in the process of designing ECAs.

		1	2	3	4	5	6	7	8	9	10	11	12	13
Review Design Guidelines	1	•												
Develop Detailed Block Diagram	2	X	•	X	X	X								
Develop Prelimary HW ICD	3	X	X	•	X									
Develop Prelimary SW ICD	4	X	X	X	•									
Refine Test Strategy	5	X	X	X	X	•								
Review CAD Guidelines	6	X	X				•							
Develop Preliminary Parts List	7		X	X	X	X	X	•						
Develop Library Parts (Footprints)	8							X	•					
Complete Preliminary Schematic	9		X	X	X	X	X		X	•	X			
Complete Preliminary Component Layout	10			X		X	X		X	X	•			
Complete Preliminary Themal/Power Est.	11		X				X					•		
Submit Preliminary CAD BOM to ISE	12						X		X	X			•	
Update Req'ts Matrix/Risk & Cost Assessment	13											X	X	•

(a)

		1	2	3	4	5	6	7	8	9	10	11	12	13
Review Design Guidelines	1	•												X
Develop Detailed Block Diagram	2	X	•	X	X	X					X	X		X
Develop Prelimary HW ICD	3	X	X	•	X						X	X		X
Develop Prelimary SW ICD	4	X	X	X	•						X			X
Refine Test Strategy	5	X	X	X	X	•					X			X
Review CAD Guidelines	6	X	X				•					X		
Develop Preliminary Parts List	7		X	X	X	X	X	•						
Develop Library Parts (Footprints)	8							X	•					
Complete Preliminary Schematic	9		X	X	X	X	X		X	•	X	X		
Complete Preliminary Component Layout	10			X		X	X		X	X	•	X		
Complete Preliminary Themal/Power Est.	11		X				X					•		
Submit Preliminary CAD BOM to ISE	12						X		X	X			•	
Update Req'ts Matrix/Risk & Cost Assessment	13											X	X	•

(b)

Figure 3–3: A comparison between the DSM representing the espoused process for the preliminary design stage of development (a) and the actual process for the same stage as revealed through interviews with engineers (b).

Concatenating the Mini-DSMs

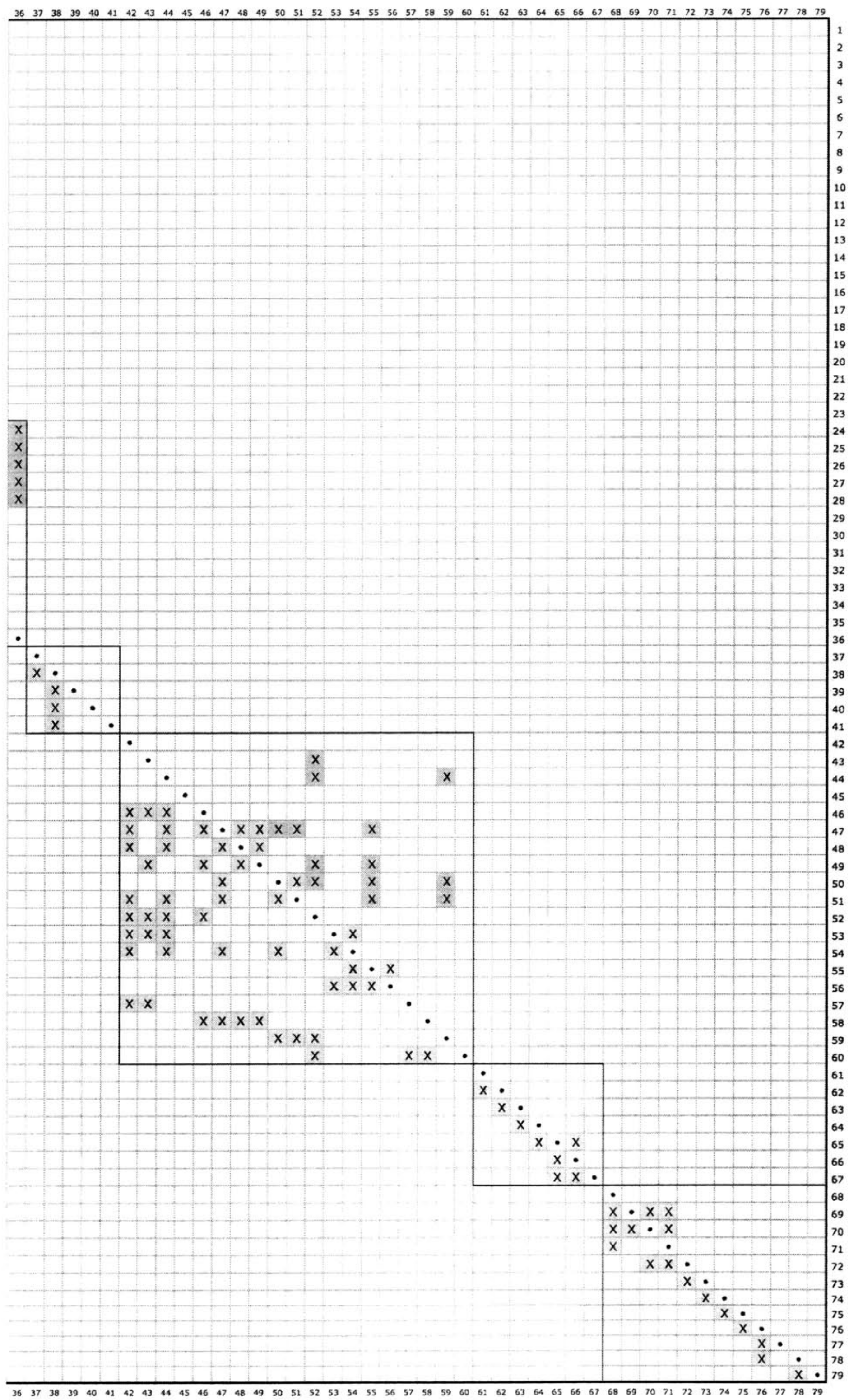
The second assumption in the process documentation which needed to be undone was that each stage can be documented separately from the others. This assumption is the result of an adherence to a waterfall development paradigm. Each stage is documented as a process unto itself which is impacted only by the design stages which have come before. Under this paradigm information is passed forward to upcoming design tasks but information is never passed backward. This manner of documenting the development process is useful because it simplifies the prescription for how to perform the engineering tasks. However, this manner of describing the tasks masks the information dependencies present in the context of the end-to-end design process.

This assumption was undone in two steps. The first step involved concatenating the mini-DSMs into a single large DSM. This is a straightforward process and the result is shown in Figure 3–4. This large DSM lists the tasks for the entire ECA development process. It remains divided into highly iterative design stages because information dependencies between stages have not yet been added.

To fill in the remaining information dependencies this end-to-end DSM was taken back to engineers for a second round of discussion. Viewed in the context of the larger engineering process, the engineers once again made changes to the list of tasks. Some tasks were eliminated. Others tasks were combined. More importantly, the engineers now made comments about information dependencies between the stages of design. The majority of these dependencies involved the feed-forward of information from one stage to the next. These dependencies are represented by marks below the diagonal of the DSM and outside of the highly iterative design stage regions. A second sort of dependency was also added. These dependencies are potential rework loops represented by marks above the diagonal in the DSM. The result of the second round of interviews with engineers is shown in Figure 3–5.

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
Establish Subsystem Req'ts	1 Authorize to Proceed	•																																		
	2 Review Contract Specification	X	•																																	
	3 Review System Architecture		X	•																																
	4 Develop Preliminary Req'ts Compliance Matrix		X	X	•																															
	5 Establish Primary Derived Requirements		X	X		•																														
	6 Establish Sub-Element CAIV Targets		X	X			•																													
	7 Identify Driving Req'ts for Further Trade-off		X	X	X			•																												
	8 Establish Supportability Req'ts		X	X					•																											
	9 Complete Requirements Analysis Checklist				X	X	X	X	X	•																										
	10 Create Design Development File									X	•																									
Develop Design Concept	11 Address Technology Issues											•	X				X		X	X																
	12 Address Producibility Issues											X	•				X			X																
	13 Develop Mechanical Approach													•					X		X															
	14 Define Design Approach													X	•				X	X																
	15 Indentify Firmware Requirements															•			X																	
	16 Define ICD Approach																•																			
	17 Define Interfaces																	•																		
	18 Review Prelim. Concept Block Diagram															X	X	X	•																	
	19 Generate Final Derived Oper. Req'ts																			•																
	20 Review Key Characteristics																				•															
	21 Update Req'ts Matrix/Risk Assesment																					•														
	22 Develop Factory Test Approach																			X	X	X	•													
	23 Develop BIT Philosophy												X		X				X				X	•												
Preliminary Design	24 Review Design Guidelines																								•											
	25 Develop Detailed Block Diagram																								X	•	X	X	X					X	X	
	26 Develop Prelimary HW ICD																								X	X	•	X						X	X	
	27 Develop Prelimary SW ICD																								X	X	X	•						X		
	28 Refine Test Strategy																								X	X	X	X	•					X		
	29 Review CAD Guidelines																								X	X				•					X	
	30 Develop Preliminary Parts List																									X	X	X	X	X	•					
	31 Develop Library Parts (Footprints)																															•				
	32 Complete Preliminary Schematic																									X	X	X	X	X		X	•	X	X	
	33 Complete Preliminary Component Layout																										X		X	X		X	X	•	X	
	34 Complete Preliminary Themal/Power Est.																									X				X					•	
	35 Submit Preliminary CAD BOM to ISE																													X		X	X			•
	36 Update Req'ts Matrix/Risk Assesment/Cost																																		X	X
PDR	37 Electrical Peer Design Review																																			
	38 Update Design Documents																																			
	39 Hold PDR (Design)																																			
	40 Hold PDR (Producability)																																			
	41 Hold PDR (Testability)																																			
Detailed Design	42 Complete Schematic Capture																																			
	43 Develop Critical Timing Diagrams																																			
	44 Complete Parts List																																			
	45 Submit NPRs as Necessary																																			
	46 Complete Net List																																			
	47 Complete Layout																																			
	48 Producibility Review																																			
	49 Complete Routing Rules																																			
	50 Complete Mechanical Structure Design																																			
	51 Complete Power Calc/Thermal Analysis																																			
	52 Complete Electrical Analysis/Simulation																																			
	53 Complete Software ICD																																			
	54 Complete Hardware ICD																																			
	55 Complete Testability Review																																			
	56 Create Test Spec																																			
	57 Complete Theory of Operation																																			
	58 Develop Design Rules Package																																			
	59 Complete Reliability Prediction																																			
	60 Update Requirements Compliance Matrix																																			
CDR	61 Electrical Peer Design Review																																			
	62 Update Design Documents																																			
	63 Hold Critical Design Review																																			
	64 Action Item Review & Response																																			
	65 Update Design Documentation																																			
	66 Finalize Parts List																																			
	67 Release Documentation to Vault																																			
Board Route & Fab	68 Release Design Routing Package																																			
	69 Routing Set-up (Drafting)																																			
	70 Review Final Component Placement																																			
	71 Review Critical Routes																																			
	72 Complete Routing																																			
	73 Final Routing Review																																			
	74 Incorporate Routing Review Action Items																																			
	75 Routing Documentation Review																																			
	76 Release Data Package to Vault																																			
	77 Release Test Spec																																			
	78 PWB Fabrication																																			
	79 Assemble First Piece																																			
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35

Figure 3–4: (This page and facing) This DSM, which shows the concatenation of the mini-DSMs representing each design phase, reflects the waterfall development paradigm used to create the ECA development documentation.


36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79

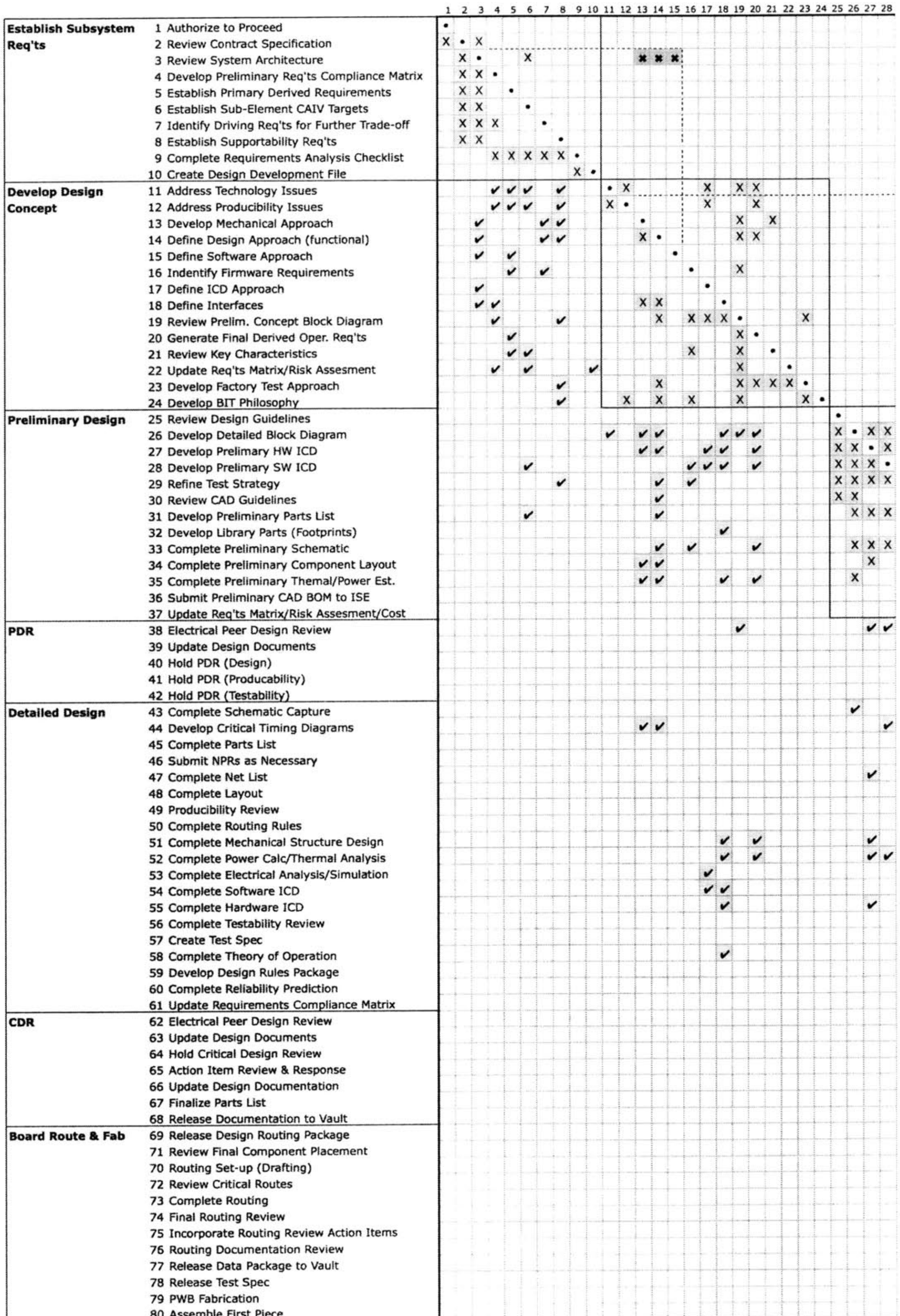
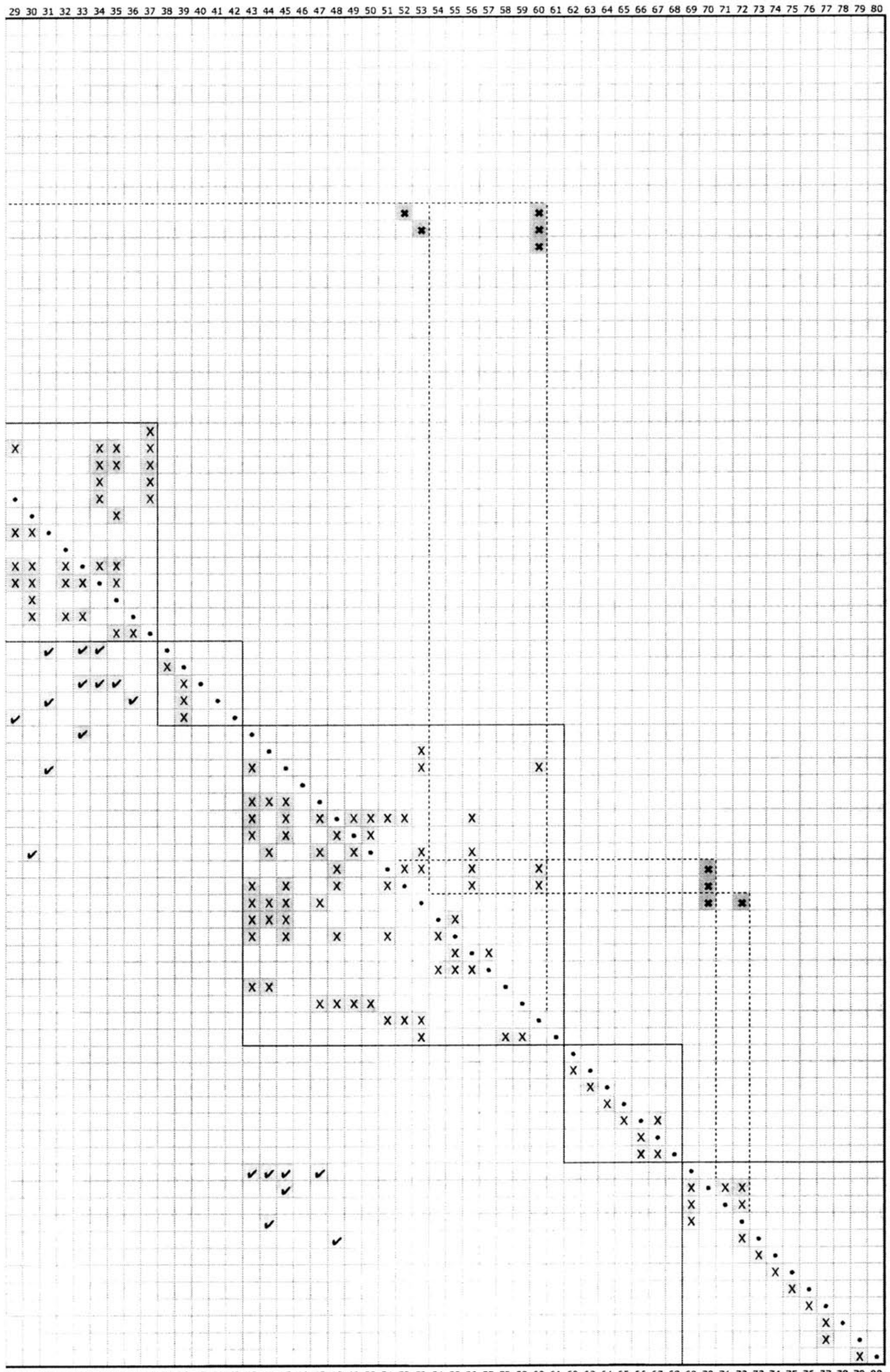


Figure 3–5: (This page and facing) The completed DSM incorporating the interview feedback of engineers which documents the ECA design process. Rework loops and iterative development phases are clearly visible.



THE FINAL DSM

Engineering process documentation was combined with two rounds of semi-structured interviews with engineers to arrive at a DSM which represents the ECA development process-in-use. Engineers were asked to comment on the DSM at two different levels. First they were asked to inspect each design stage separately. For a given stage of the engineering process (i.e. preliminary design, detailed design) the engineers reviewed the tasks required and the information dependencies inside of design stages. Any tasks that were missing were added. Any sequencing corrections were made. In some cases tasks were deleted. Then engineers were asked about the information flows during the design process. Additions were made to the information dependencies derived from the process documentation. This was the first level of discussion. The second level of discussion involved looking outside of the process stages for information dependencies. At this level engineers identified rework loops which could result in a portion of the design having to be re-engineered. Also identified were feed-forward information dependencies which result from information carried forward from one design stage to the next.

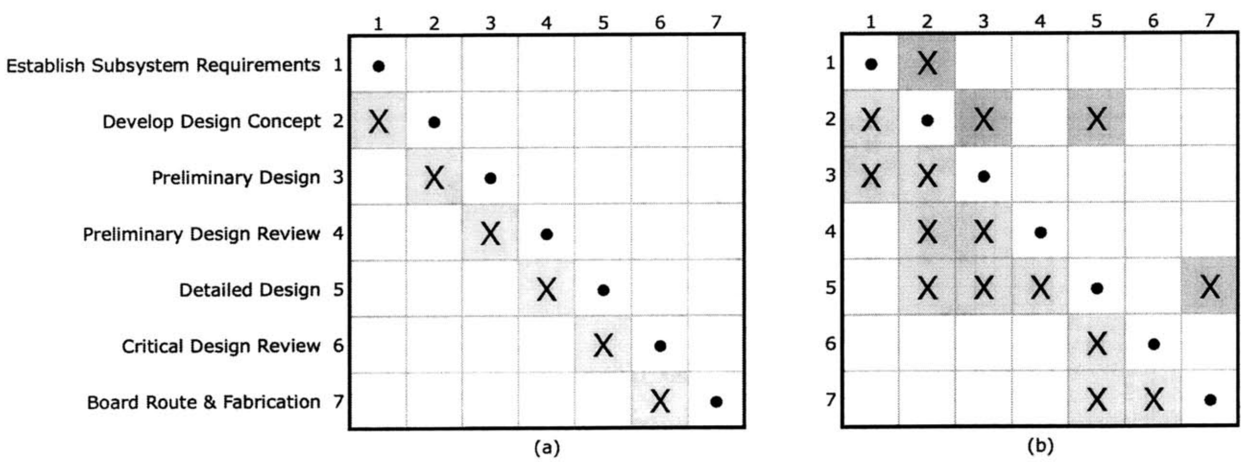


Figure 3–6: Meta-DSM showing the information dependencies between design stages. The figure on the left (a) represents the espoused process. The figure on the right (b) represents the process in use.

The resulting DSM shows that the ECA design process is highly integrated. Three features are evident. First, there are stages of highly iterative engineering development along the diagonal the DSM. This is partially the result of the manner in which the DSM was constructed. However, it also represents the reality of the design process as regular design reviews are typically mandated by the customer. This review structure segments the design process. Second, there is a large amount of information fed forward to future design stages. This is seen in the below the diagonal marks in the

DSM. This feature of the waterfall development process is assumed but not mentioned explicitly in the process documentation. It doesn't fundamentally alter consecutive nature of the design process. Unless an analysis such as is done in this thesis is anticipated, it is not necessary to document these information exchanges. The third noteworthy feature is the appearance of above the diagonal marks in the final DSM.

All of this is more clearly seen in the meta-DSM shown in Figure 3–6. The meta-DSM shows the design process at the level of information dependencies between design stages. The tasks which comprise each design stage are rolled up so each stage is a single line. The espoused process shows a sequential waterfall development process. This process represents each stage of design as a project unto itself and implies that any given stage can be completed in isolation provided that the preceding stages have finished. The meta-DSM representing the process-in-use shows a different picture. It shows that the stages of the design are highly connected and that it is impossible to complete one stage in isolation from the others. The implication of this result is that if one stage of the design, such as circuit board engineering (board route and fabrication), is developed remotely substantial information will have to communicated to the engineers working on the other stages of the design.

CHAPTER 4: ANALYSIS, RESULTS & DISCUSSION

CASE STUDY #1

Analysis & Results from Case Study #1

Case study #1 involves records from the development of 25 circuit boards. The data is presented in tabular form in Appendix C. The records used to generate the data span the past two years. Included in the data are the circuit boards developed for Project A which represent a majority of the distributed engineering examples in this data set. The Project A designs are studied further in case study #2.

The data collected for this case study come from records of information exchange between engineering departments. Each documented exchange between departments was noted and the date of the exchange recorded. This data was collected for all of the tasks in the circuit board design process. The task focused upon in this case study is the component placement, or layout, step in circuit board design. This task was chosen because it is interdependent with other engineering tasks as can be seen in the final DSM (Figure 3–5). Component placement is step number 71. This interdependence means that placements should be impacted by restricted communications bandwidth between engineering teams.

The same analysis documented here could be performed by replacing layouts with any of the steps which make up circuit board design. Placement revisions was chosen because this was the only step which had complete data for a large enough number of designs. Netlist revisions were also tabulated and are presented in the data in the appendix. Netlist data was not available for enough circuit boards to be used as part of this analysis.

Before comparisons can be made, the inherent complexity of each design must be taken into account. In this data set, the inherent complexity is reflected in the number of placement iterations required to complete a design. This number ranges between one and fifteen. A metric which is less dependent on design complexity was developed by calculating the rate at which placement updates occur. The rate of development was calculated for each circuit board by dividing the total number of design days required to finalize the placement by the number of iterations. The resulting measure is in units of days per revision. Figure 4–1 shows the development trajectory for a typical circuit board in this case study. In this example the average rate of design iteration is 62 working days divided by 13 design iterations – 4.8 days per revision.

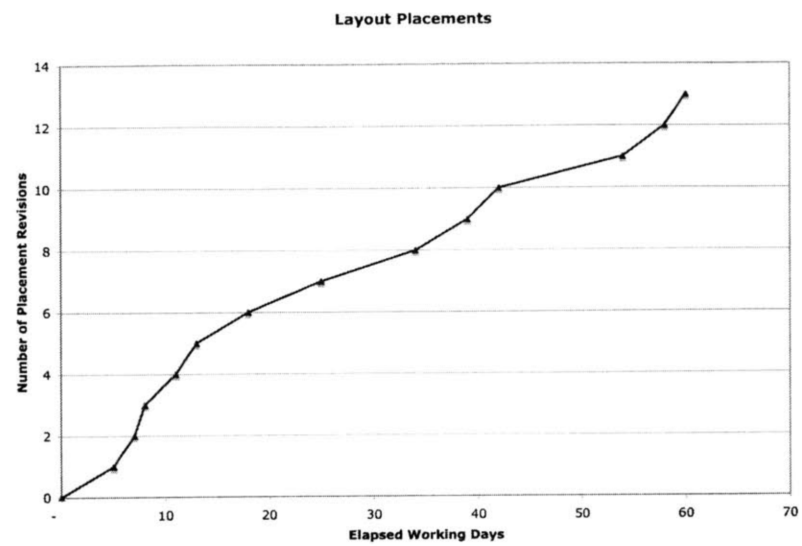


Figure 4–1: Graph showing the development of component placement iterations plotted against elapsed working days. The average number of days per revision was found by dividing the total number of working days by the number of revisions.

The distance separating the team members was then calculated based on knowledge of the work location of each of the engineering disciplines. Three engineering departments are primarily involved in the layout portion of design – circuit board design, mechanical engineering, and electrical engineering. The distance between these three pairs (as described in Figure 4–2) was recorded for each design. The total team distance was calculated by taking the sum of the three distances between the pairs.

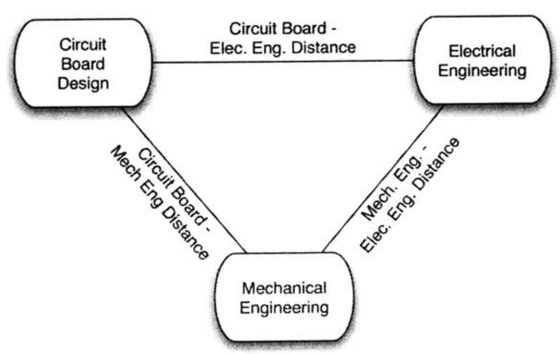


Figure 4–2: Diagram showing the three distances recorded for each circuit board to capture the level of team distribution in data set #1.

By taking the average rate of component placement for each circuit board as described above and plotting these against the total team distance we are able to produce Figure 4–3. This graph shows that these data exhibit no significant dependence on the total team separation distance. The variance present in data far outweighs any trend which might be observed.

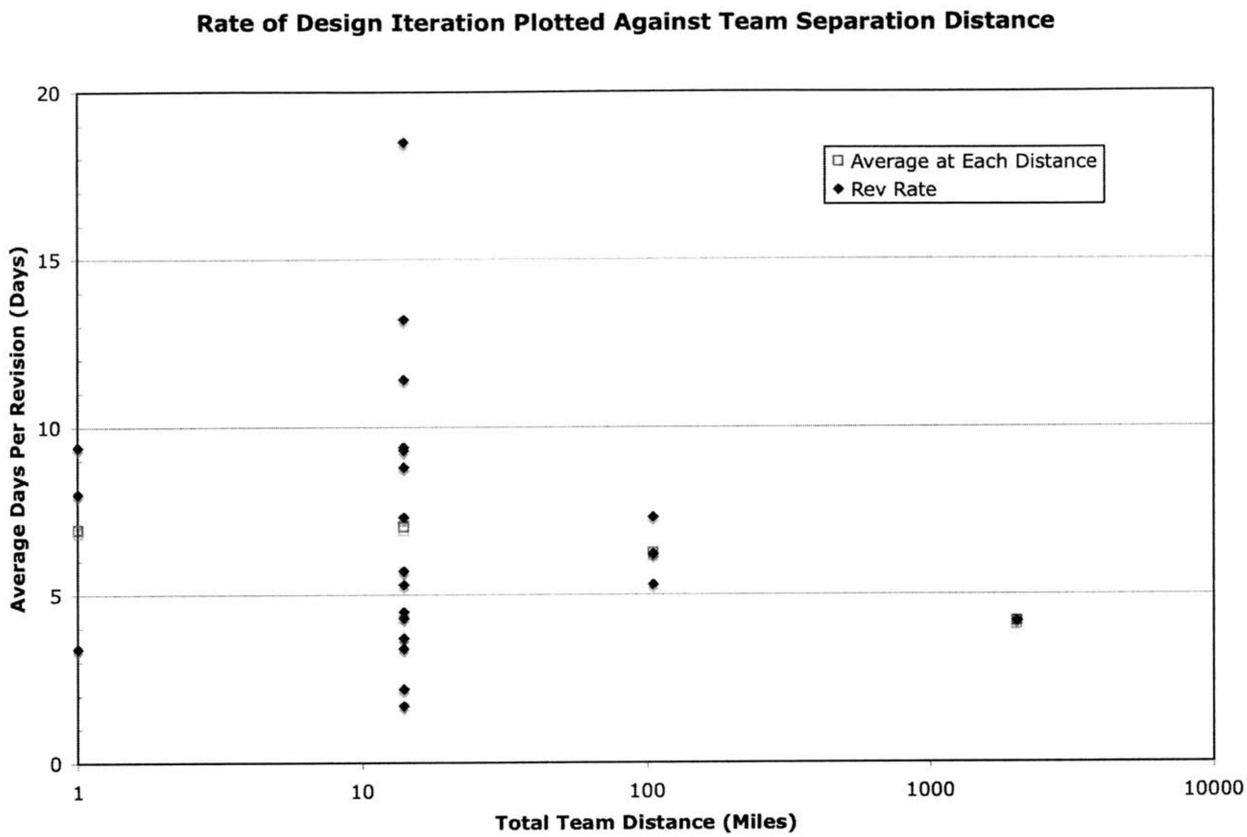


Figure 4–3: Plot of the average rate of component placement iteration as a function of the total distance separating the engineers. There is no statistically significant trend in this data.

Discussion of Case Study #1 Results

The data presented above is highly variant and does not expose any dependence of the rate of design iteration on the distance separating engineers. To the extent that the rate of placement iteration represents the pace of engineering, a decision maker inspecting the graph generated from this data would reasonably conclude that the geographic distribution of engineers does not impact the pace of engineering development.

It was expected there would be a step function in the data. The step up in development times was expected to occur where the distance impeded the regular face to face contact of engineers. It was expected that designs developed by localized teams would proceed at a quicker pace than designs developed by distributed teams.

There are many reasons why the data from case study #1 may not reflect the expected trend. Many day to day occurrences affect the pace of layout updates that are not associated with the separation of engineers. Designers are often working on more than one project simultaneously. Shifting job priorities can result in long delays between iterations. Even something as trivial as the illness of an engineer can impact the data.

Anecdotal reports of engineers suggest that geographic dispersion interferes with their design practices. Not being able to converse face to face is noisome and engineers seek to avoid it when possible. Because of these facts, it is tempting to say that the data does not adequately capture the phenomenon being studied. However, the process of designing circuit boards is highly digitized. Designers work with layout tools and CAD programs which readily create images of design which can be emailed to peers. The lead engineer in the circuit board design group, who has over 25 years of experience, felt that "distance won't make any difference." His collaborators were not as convinced. But the possibility remains that there is no trend to be found.

CASE STUDY #2

Analysis & Results From Case Study #2

Appendix D contains the data collected from the Integrated Master Schedule for the ECAs studied in case study #2. Project A actually involved seven ECAs. One ECA was withheld from these datasets because email correspondence showed it was not on the critical path for the project. As a result the slack in the schedule was included in the reported time to complete this ECA. Therefore, the amount of development time recorded in the IMS for this ECA is significantly larger than what was actually required. Because of this idiosyncrasy ECA #7 is not included in the results for case study #2.

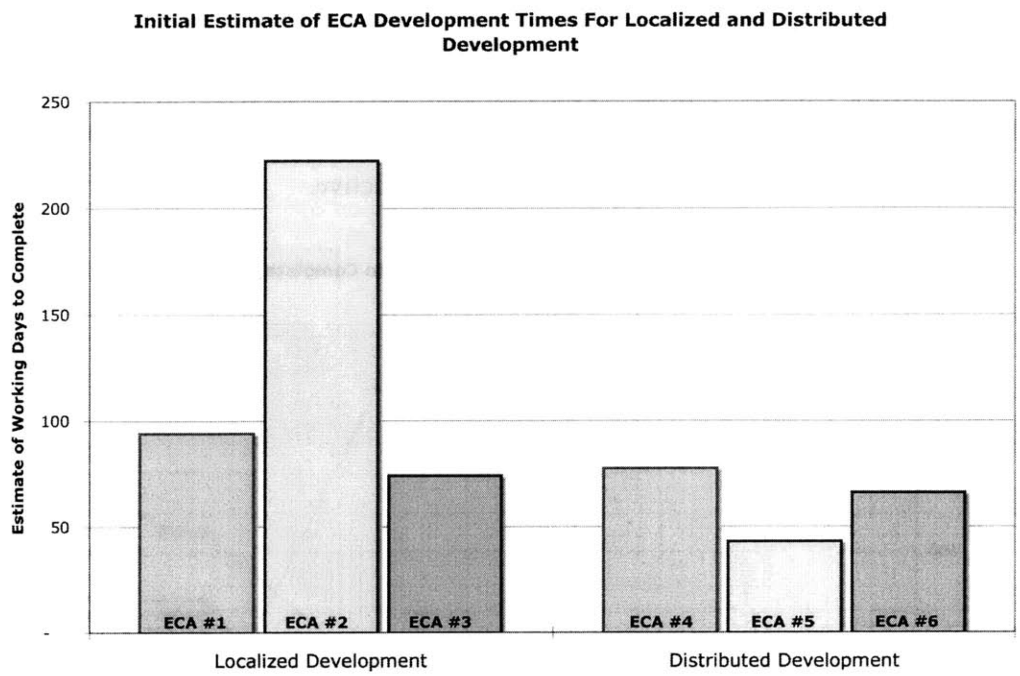


Figure 4–4: Graph depicting the range of initial estimates of working days required to complete the Project A ECAs.

Before the data from data set #2 can be used to make comparisons, the data must be normalized to account for the varying complexities of the projects. Every ECA which was studied has a different inherent complexity. This inherent complexity is related to a number of factors which are intrinsic to the designs: the level of sophistication of the electrical functionality, the types of components used, and the physical environment which the hardware must endure, *et cetera.* A consequence of these differences in inherent complexity is that the times required to complete the engineering design vary from ECA to ECA. This variability in design time is understood by the project planners and is reflected in the estimates of engineering design time at the outset of the project as shown in Figure 4–4. The data shown in this figure do not represent the time it actually took to design the ECAs, they are the initial estimates of time required to complete the designs made by engineers. This initial estimate of design time is used as a proxy for design complexity.

Because designs vary in complexity it is not meaningful to simply compare the number of working days required to complete a design. Before making any comparisons, the design times are normalized by their inherent complexity by dividing the actual number of working days required to complete an ECA by the early estimate of the required number of days. The initial estimates are recorded in the IMS as the task duration when the schedule was initially developed. Inspecting the data from the IMS reveals that the first estimates are quickly revised as the preliminary design development commences. These trends are depicted in Figure 4–5. After being revised at the beginning of the preliminary design stage, the estimates of time to complete stabilize and are not revised until the detailed design is underway. During detailed design, revisions to design time tend to be made to accommodate schedule slip and are reactive rather than predictive.

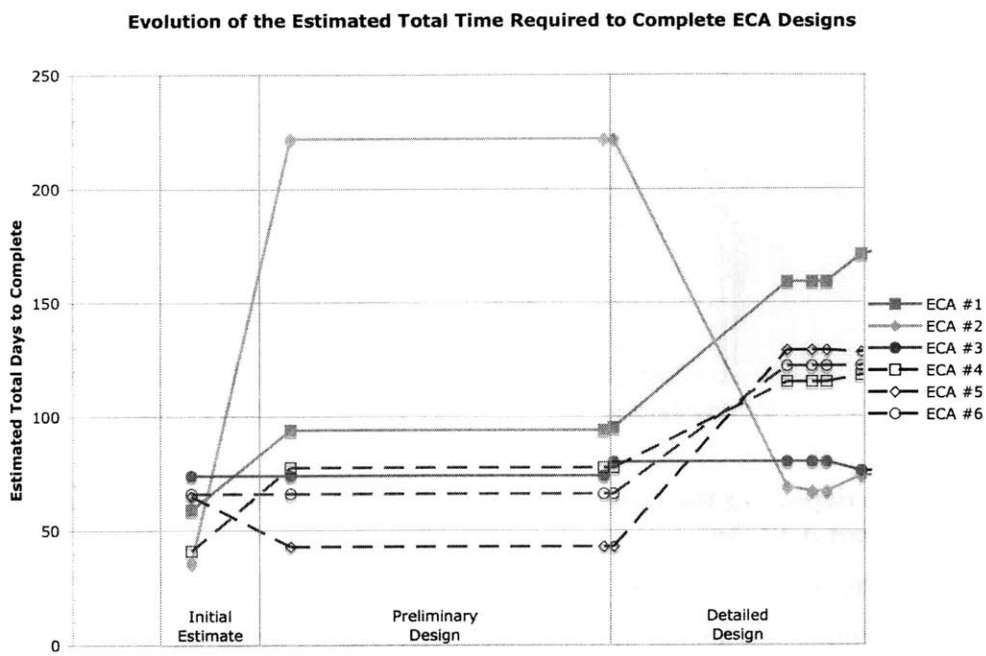


Figure 4–5: Plot showing the evolution of the estimated time to complete each design as function of design maturity. Note that during preliminary design estimates are revised based on insights gained about the design. During detailed design all estimates in this data set increase because the projects have fallen behind schedule. Dashed lines represent distributed development teams.

As a result of these observations, the revised estimates made during preliminary design represent the most informed estimate of the complexity of the design task. Consequently, the value used to normalize the time-to-complete data is the stable estimate developed during the preliminary design stage.

When the development times for each design are plotted as a fraction of the initial estimate of time-to-complete we see that there is a remarkable difference between ECAs developed by localized teams and those designs developed by geographically distributed teams. This data is plotted in Figure 4–6. The data show that distributed teams overshoot their budgeted development times by nearly twice as much as co-located teams. The two populations – localized and distributed teams – are statistically separate. The result of a t-test on the two populations is 0.03.

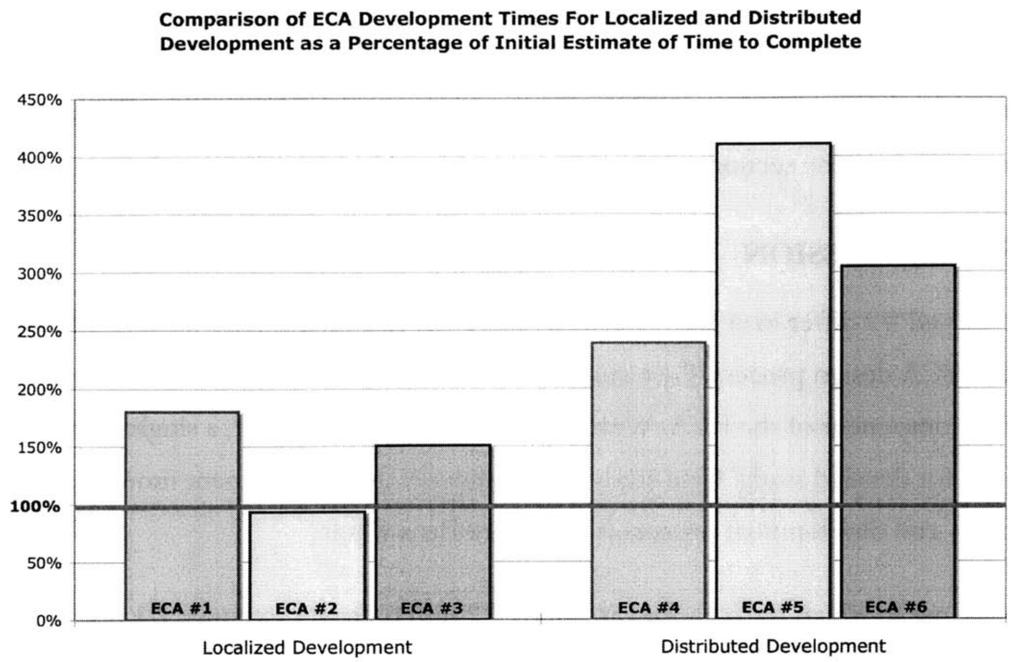


Figure 4–6: Graph depicting the ECA development times for localized and geographically distributed teams. Results are presented as a fraction of the initial estimate of the time required to complete the design.

The IMS data identifies only high level trends. There is little more that can be extracted from the data. There is no indication of the mechanism by which engineering design is slowed. Yet the trend in the data indicates that the geographic distribution of engineers does slow the pace of design.

Discussion of Case Study #2 Results

The results of case study #2 show a distinct difference the in pace of development for localized teams versus distributed teams. This result agrees with the anecdotal observations of the engineers involved with Project A. Participants recounted having to compose lengthy emails describing design changes rather than marking up a drawing and carrying it to a colleague's desk for discussion. The emails led to frequent misinterpretations and additional rounds of communication were required where a quick face-to-face conversation would have sufficed. The geographic distances involved in the distribution of engineers for this project were large enough to prohibit regular face to face meetings. The time required to drive between the primary facility and the circuit board designers was two hours each way. In fact, no face to face meetings occurred except for a single design review of the electrical design.

Further, engineers recollect and project records confirm that when the portions of the design which were assigned to remote team members were nominally completed, substantial work remained. It is possible that there was more undiscovered rework in the sections of the design completed by distributed teams than in the sections of the design completed locally.

CROSS CASE DISCUSSION

Case studies #1 and #2 differ in at least two ways. First, the two case studies involve different portions of the ECA design process. Case study #1 looks at data for the development of just the circuit board, a component of the ECA. Within circuit board development, a single step in the design stage is selected for detailed study. Case study #2 documents the development times for the entire ECA; the end-to-end development process is considered as a whole.

The second manner in which the data for case study #2 differs from case study #1 is in detail of the information available. The IMS data studied in case study #2 recorded only changes to the schedule. No explanatory information was available. The data for case study #1 is much more detailed. It provides records of the dynamics of engineering development but does not keep track of higher level program concerns such as adherence to schedule.

While the two data sets are not alike in scope or scale, it is important to note that the case study #2 designs are a subset of the case study #1 designs. For this reason it is particularly interesting that the trend observed in case study #2 is not exposed in case study #1. There are at least two possible explanations for this result. The first is that the trend observed in case study #2 is not real. It is possible, given the small size of the data set, that the results were coincidental and not a result of geographic distribution of engineers. However, the anecdotal reports of engineers who worked on this program support the hypothesis that distribution slows the pace of development.

A second possible explanation for the results is that the data used in case study #1 does not capture the impact of the geographic distribution. This could happen in a number of ways. The natural scatter in the data may be so large that it overwhelms any trend that might be present. Or, perhaps other factors beyond geographic distribution impact the data more than geographic distribution. Perhaps engineers work over-time or work more quickly when deadlines loom. Perhaps the data reflects a "get the job done" attitude which compels engineers to do whatever necessary to meet a schedule. Furthermore, as described by Steward in one of the original works on DSMs:

> "Engineers working on variables within the same block, and particularly within smaller blocks, should work in close proximity, by physically and organizationally, to facilitate communication. It is often possible, by close communication, to work out a consistent set of variables within a small block without a formal documented design iteration process." [24]

It is possible that the data collected on design revisions does not capture informal iterations that occur when engineers are working in close proximity. Only the documented iterations are captured. When engineers are in close proximity perhaps iterations happen more quickly than can be captured by the records process used by the circuit board design group. The actual pace of revision may be faster than indicated by the recorded data.

The discrepancy between the results of the case studies indicates that the disadvantages of distributed engineering development can be masked in the data that might be used to evaluate whether or not to distribute engineers. This result suggests that the level of aggregation of tasks used when assessing the pace of engineering development can impact the ability to elucidate trends.

CHAPTER 5: CONCLUSIONS & FUTURE WORK

SUMMARY OF FINDINGS

Distributed Development Projects Overrun Schedules by Twice as Much as Localized Projects

The hypothesis tested here is whether or not the unintended consequences of the decision to distribute engineering development can slow the pace of engineering development to the point that the advantages are overcome by the additional engineering burden. The results of case study #2 support this hypothesis. The designs which were distributed overran their scheduled development budget by more than twice as much as the localized designs. Presumably, if the project planners were aware of the impacts of distributed engineering they would have done a better job of budgeting time to complete these designs. The schedule overrun indicates not only that the distribution of design engineers slows the pace of engineering development but also that the consequences of distributing engineering development are not well understood by project planners.

The most tangible result of this work is shown in Figure 4–6. The results of case study #2 show a significant difference in project execution time between the ECAs developed by localized teams and the ECAs developed by distributed teams. The distributed teams overran their schedules by more than twice as much as the localized teams.

While there are many reasons for the distribution of engineering development, the consequences of distributing engineers must be understood and weighed against the benefits. The design schedule overruns documented for the distributed development of ECAs correspond directly with cost overruns. If the consequences of geographic distribution were better understood we would expect project planners to account for the reduced pace of engineering development in their estimates of the time required to complete a project.

In addition to the reduction in the pace of development, engineers working on distributed teams are often frustrated by communication limitations. Further, cultural boundaries and lack of personal relationships with colleagues can result in personality clashes which may be mitigated by co-locating engineers.

The Hardware Development Process Does Not Follow a Waterfall Pattern

In contrast to the waterfall model espoused by the process documentation, the engineering of projects studied consists of iteration among interdependent tasks. The omission of cyclical development cycles may partly result from the inability to represent these relationships in typical scheduling tools. Conventional project management tools are not capable of representing the cyclic dependencies present in the engineering development of complex, integral products. These cyclic dependencies are readily depicted in a DSM. Consequently, the DSM is an excellent tool for understanding the dynamics which could result from changes to the design, the design process, or the environment where the process is executed.

The drawback of the DSM is that it is of little utility for planning and scheduling a project. Because the DSM accommodates the representation of cyclic project dependencies, it does not help in creating a deterministic schedule Through the use of statistical simulation the DSM can be used to generate an expected schedule, but the DSM is not useful for generating an exact schedule.

The inability to generate an exact schedule does not reveal a weakness in the DSM tool. Rather it points out that the deterministic schedules generated by work breakdown structures and Gantt charts are incomplete representations of the work that may need to be done. By simplifying the project dynamics so that there is no possibility of rework and iteration, the traditional tools generate a simplified picture of the dynamics of project execution. This simplification is reasonable and useful at a certain level of abstraction. When the scheduled tasks are general enough, any required iteration happens within a task and appears only as additional time required to execute the task. Documenting the engineering tasks at this level allows deterministic scheduling and places all uncertainty in the duration of the design tasks rather than in the sequence by which the tasks will be executed. Moving the uncertainty to reside only within the duration of design tasks enables the use of management reserve and tools like critical path management.

When the task list in the schedule becomes too granular, for sufficiently integral systems, iteration between tasks becomes inevitable. The sequence in which tasks will be executed is no longer deterministic. Instead, changes made in one stage of the design may impact choices made earlier causing iteration or rework. This uncertainty may be what makes DSMs less common in the schedule driven world of corporate engineering.

However, because the schedules mask the detailed dynamics of the engineering process they are poor tools for making decisions about the manner in which engineering tasks will be performed. Particularly for this thesis, by masking the information exchange between tasks the use of simple Gantt chart schedules may facilitate making uninformed decisions about how best to geographically divide work. The final meta-DSM shown in Figure 3–6 illustrates the stark difference between the waterfall development process espoused by the engineering process documentation and the interdependent, iterative process in use by the engineers.

The Level of Aggregation Used to Study the Data Impacts the Identification of Trends

A comparison of the results of the two case studies suggests that the level of task aggregation used to investigate the impact of distributed engineering affects the elucidation of trends. Case study #1 found no evidence of the impact of distributed engineering. However, when the level of task aggregation was increased, a subset of the designs from case study #1 showed a strong dependence on the distribution of engineers.

This finding is significant because the data from case study #1 may be used to assess the health of project. This is the data available to and used by engineers and front-line managers to monitor designs. However, at this fine level of detail the impacts of distributed engineering are not present or are masked by the variance in the data. When the level of task aggregation is increased, the impact of distributed engineering becomes apparent.

That the impact of distributed engineering is only apparent at more granular level of aggregation data is counterintuitive because the dynamics of information exchange between engineers are hidden at this level of aggregation. In case study #2 the interactions between engineers typically occur inside of a single entry in the IMS. The level of granularity is too large to expose the day to day interactions between distributed groups. The low-level, detailed data from case study #1 exposes the communications dynamics between engineers but does not expose the differences between localized and distributed development. It is as if small deviations from the schedule which are not noticed at the finest scale are integrated up into a larger impact at the project level. In order to apprehend the trend in the data it is necessary to choose the proper level of task aggregation for study.

Threats to the Validity of this Study

Subjective Measures of Complexity

In order to make comparisons between ECA design projects it was necessary to normalize out the differences in design complexity between projects. The estimate of time to required to design the ECA made be engineers at the beginning of the project was used as a proxy for the complexity of the design. This estimate was made based on the inputs of the various engineering disciplines involved in executing the design. It was informed by the discoveries made by the engineers during the preliminary design phase.

However, if the estimated time to complete the ECAs is not a good proxy for the complexity of the designs, then the results of case study #2 could be called into question. Even if the estimate to complete does not correlate to complexity, the results of case study #2 still reflect the amount of schedule overrun exhibited by the ECAs. The schedule overrun for distributed teams was substantially larger than the overrun for localized teams.

Applicability to Modular Product Architectures

The ECAs studied have an integral product architecture which begat an integral development process. In order to modularize the design of ECAs it would be necessary to specify interfaces between systems. By definition these interfaces could not be re-established for every ECA. To be useful they would need to be pre-defined and general. Defining the interfaces would allow engineering groups to work in isolation and be assured their designs could be integrated into the product. The trade-off made in this scenario is that the defined interfaces will likely require some compromise.

A thought experiment helps illustrate this point. It is possible to imagine establishing a thermal design for an ECA heatsink so that it can accommodate a specified heat flux over its surface area. This pre-determined heat flux would constitute a standard interface for board design. As long as electrical engineers selected components which did not violate this predetermined heat flux requirement, the electrical design and the thermal analysis could be decoupled. There are at least two compromises made by this scenario. First, electrical engineers will be limited in the components they can select. All components must have a dissipated heat flux lower than the established limit. This limit would exist even though it may be possible to exceed the heat flux limit for small areas or locations close to the edge of the board without redesigning the heatsink. However, it will not be possible to include higher heat flux components without performing a thermal analysis. Requiring a thermal analysis re-couples these tasks and reinstates an integral process.

The second compromise resulting from the defined interface in this thought experiment occurs when the heat fluxes for the selected components are substantially lower than the established limit. In this case the thermal management system is over designed. It would be possible to perform a thermal analysis and determine the optimal amount of thermal management hardware. Again, this analysis re-couples these two stages of design. Leaving the stages decoupled means being satisfied with a suboptimal system which may be larger or heavier than necessary.

Products with a modular architecture would not be as prone to communication problems between engineering teams. Modular architectures with specified interfaces would facilitate the distribution of engineering.

Applicability to Other Design Domains

The systems studied in this thesis come from a single company. They were all designed for similar environments and using the same development process. To make broad generalizations about the impact of distributed engineering of the pace of engineering development it would be necessary to study engineering systems in other environments, markets, and applications.

FUTURE WORK

Future work could address the threats to the validity of this study described above. This study was not able to compare ECAs which all perform the same function with varying amounts of geographic distribution of engineers. By studying a population of ECAs which all perform the same function the complexity of the designs would be approximately equal. This would hold constant one more variable affecting the pace of design. By collecting data for even more designs it may be possible to get a population of ECAs which are all of the same complexity and study them for indications of the impact of distributed engineering.

Additional future work could include expanding the scope of study to other industries and engineering disciplines. The standard practices of the aerospace industry include extensive review and documentation steps which reinforce a waterfall development process. Integral product architectures exist in many industries which do not adhere to a waterfall development model and it would be interesting to see if the dynamics observed in the aerospace industry are present elsewhere.

APPENDIX A: PRIMER ON READING DSMS

Traditional engineering process visualization tools fail to capture the complexity of the engineering design process. Large engineering organizations typically keep records of project performance. This is particularly true in the aerospace industry where government contracts often mandate the use of standard project tracking tools. The standard suite of tools includes Gantt Charts (which are often constructed using Microsoft Project), Pert Charts and critical path/critical chain management, and documentation of standardized process steps required to complete the task at hand.

The traditional tools used to document projects and track their execution can typically represent two types of relationships between tasks: independent and dependent. However, there are actually three different ways that any two design tasks can be related. [12] They are shown in Figure A–1. Traditional project management tools adequately capture the first two. DSMs are capable of representing all three types of task relationships.

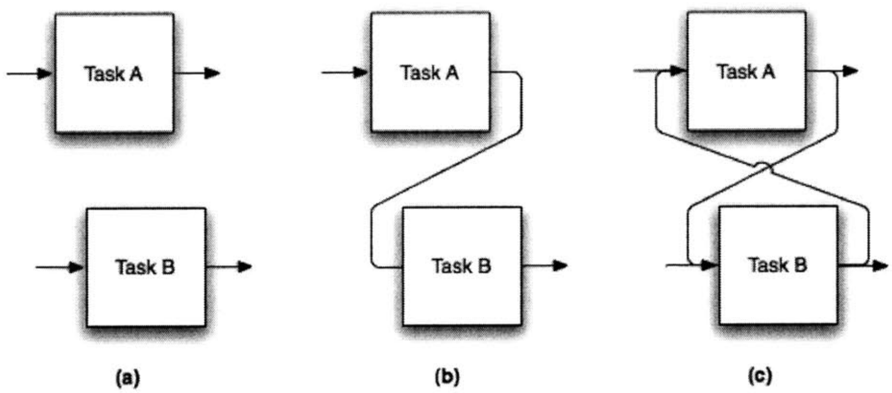


Figure A–1: Illustration of the three types of relationships which can exist between two design tasks: (a) independent, (b) dependent, and (c) interdependent.

DSMs in the context of this thesis are a compact representation of the information flows required within an organization to develop a product. For those readers that are not familiar with DSMs a brief review of how to read them is presented below. Readers who are familiar with DSMs could skip this section without impacting their understanding of this thesis.

It is informative to compare the familiar representation of related tasks in a Gantt chart with the same tasks represented in a DSM. In this manner it is possible to gain some intuition about the topography of a DSM. Figure A–2 shows a very simple schedule with four linked tasks. Accompanying it is the DSM which represents the same tasks.

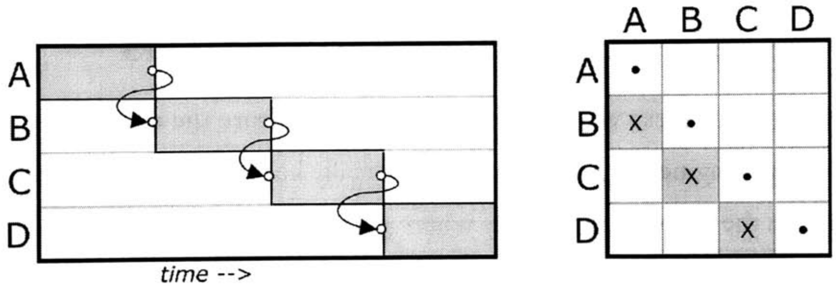


Figure A–2: A sequence of four consecutive tasks is represented in a Gantt Chart on the left and in a DSM on the right.

The DSM is read in the following manner:

- An X in a row means the task which is represented by that row *requires* information from the tasks which have X's in their columns.
- An X in a column means that the task which is represented by that column *provides* information to the tasks which have X's in their rows.

To exemplify this, consider task C. In row C there is an X in column B. This indicates that task C requires information from task B. Further, in column C, there is an X in row D. This indicates that task C provides information to task D. Notice that the X's in the boxes correspond to the *arrows* which connect tasks in the Gantt chart.

Of course it is common for tasks to require input from more than one predecessor task. An example of this is shown below in both schedule and DSM format in Figure A-3.

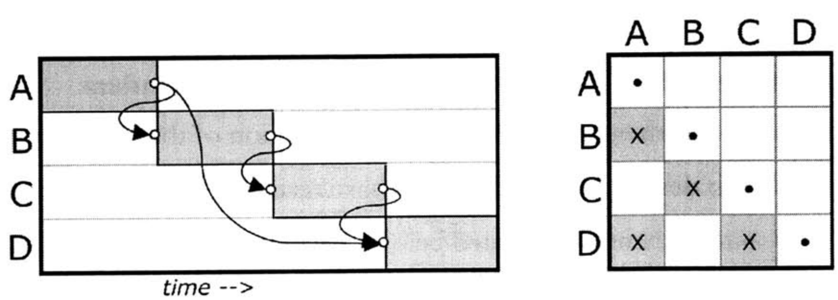


Figure A–3: Four consecutive tasks are represented in a Gantt Chart on the left and in a DSM on the right. In this figure task D requires inputs from both tasks A and C.

Here task D requires information from both tasks A and C to be completed – thus the two X's in row D. If tasks are sequenced as above in such a way that information is passed forward from one task to another and there is never any need to backtrack or iterate, the DSM will always be lower triangular. There will never be any X's above the diagonal. These types of task dependencies are represented equally well on Gantt charts as on DSMs.

It is in the upper triangle above the diagonal where DSMs are capable of representing dependencies not allowed in conventional project management tools. Iteration is necessary and often beneficial to engineering development. However, it is not possible to adequately represent on a schedule. Below is a DSM which shows two interrelated tasks. It is shown in Figure A-4 next to its corresponding schedule (which violates the rules for dependencies in schedules).

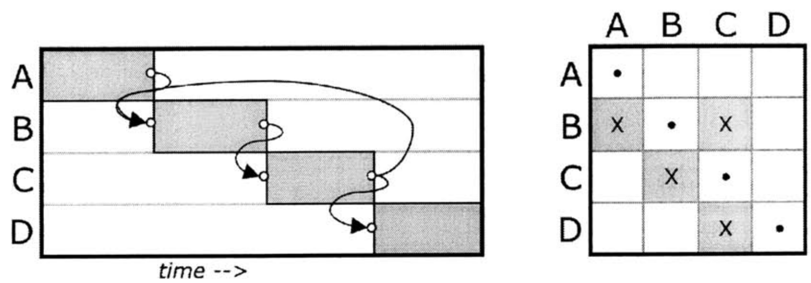


Figure A–4: Interdependent tasks are shown as they would be represented in a Gantt Chart (left) and a DSM (right). Note that the cyclic dependency invalidates any deterministic completion date.

Task B requires information from tasks A *and* C – thus the two X's in row B. But task C comes after B. And task C requires information from task B. In practice, it will be necessary to iterate between tasks B and C to complete the project. The interdependence is shown in the DSM by the X which is above the diagonal. Again, notice that the X's in the DSM represent the arrows in the Gantt chart. X's which are above the diagonal correspond to arrows which point backward in the Gantt chart. For this reason, upper triangular entries in the DSM are of particular interest.

Note that it not possible to say how long it will take to complete the schedule shown in Figure A–4. The time required will depend on how many times the loop between tasks B and C is followed.

When engineering processes are documented it is common to find interrelated tasks. However, it is desirable to keep upper diagonal entries as close to the diagonal as possible. The further above the diagonal an X appears, the greater the amount of rework potentially required.

APPENDIX B: MINI-DSMS FOR EACH PHASE OF DESIGN

Establish Subsystem Requirements:

		1	2	3	4	5	6	7	8	9	10
Authorize to Proceed	1	•									
Review Contract Specification	2	X	•								
Review System Architecture	3		X	•							
Develop Preliminary Requirements Compliance Matrix	4		X	X	•						
Establish Primary Derived Requirements	5		X	X		•					
Establish Sub-Element CAIV Targets	6		X	X			•				
Identify Driving Req'ts for Further Trade-off	7		X	X	X			•			
Establish Supportability Req'ts	8		X	X					•		
Complete Requirements Analysis Checklist	9				X	X	X	X	X	•	
Create Design Development File	10									X	•

Develop Design Concept

		1	2	3	4	5	6	7	8	9	10	11	12	13
Develop Mechanical Approach	1	•												
Define Design Approach	2	X	•											
Indentify Firmware Requirements	3			•										
Define Interfaces	4			X	•									
Review Prelim. Concept Block Diagram	5					•								
Generate Final Derived Oper. Req'ts	6						•							
Review Key Characteristics	7							•						
Update Req'ts Matrix/Risk Assesment	8					X	X	X	•					
Develop Factory Test Approach	9									•				
Develop BIT Philosophy	10										•			
Develop Design Verification Approach	11											•		
Preliminary Test Strategy	12									X	X	X	•	
Concept Review	13		X		X				X				X	•

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	•	X				X		X	X				
2	X	•				X			X				
3			•					X		X			
4			X	•				X	X				
5					•			X					
6						•							
7							•						
8					X	X	X	•					
9									•				
10										•			
11											•		
12									X	X	X	•	
13		X		X				X				X	•

Preliminary Design

		1	2	3	4	5	6	7	8	9	10	11	12	13
Review Design Guidelines	1	•												
Develop Detailed Block Diagram	2	X	•	X	X	X								
Develop Prelimary HW ICD	3	X	X	•	X									
Develop Prelimary SW ICD	4	X	X	X	•									
Refine Test Strategy	5	X	X	X	X	•								
Review CAD Guidelines	6	X	X				•							
Develop Preliminary Parts List Using SPL and NPR	7		X	X	X	X	X	•						
Develop Library Parts (Footprints)	8								•					
Complete Preliminary Schematic	9		X	X	X	X	X		X	•	X			
Complete Preliminary Component Layout	10			X		X	X		X	X	•			
Complete Preliminary Themal/Power Est.	11		X				X					•		
Submit Preliminary CAD BOM to ISE	12						X		X	X			•	
Update Req'ts Matrix/Risk Assesment/Cost	13											X	X	•

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	•												X
2	X	•	X	X	X					X	X		X
3	X	X	•	X						X	X		X
4	X	X	X	•						X			X
5	X	X	X	X	•					X			X
6	X	X				•					X		
7		X	X	X	X	X	•						
8								•					
9		X	X	X	X	X		X	•	X	X		
10			X		X	X		X	X	•	X		
11		X				X					•		
12						X		X	X			•	
13											X	X	•

Preliminary Design Review

Electrical Peer Design Review	1	•				
Update Design Documents	2	X	•			
Hold Preliminary Design Review (Design)	3		X	•		
Hold Preliminary Design Review (Producability)	4		X		•	
Hold Preliminary Design Review (Testability)	5		X			•

Detailed Design

Complete Schematic Capture	1	•																		
Develop Critical Timing Diagrams	2		•																	
Complete Parts List	3			•																
Submit NPRs as Necessary	4				•															
Complete Net List	5	X	X	X		•														
Complete Layout	6	X		X		X	•	X	X											
Producibility Review	7	X		X			X	•	X											
Complete Routing Rules	8		X			X		X	•											
Complete Mechanical Structure Design	9						X			•	X									
Complete Power Calc/Thermal Analysis	10	X		X			X			X	•									
Complete Electrical Analysis/Simulation	11	X	X	X		X						•								
Complete Software ICD	12	X	X	X									•	X						
Complete Hardware ICD	13	X		X			X			X			X	•						
Complete Testability Review	14													X	•	X				
Create Test Spec	15												X	X	X	•				
Complete Theory of Operation	16	X	X														•			
Develop Design Rules Package	17					X	X	X	X									•		
Complete Reliability Prediction	18									X	X	X							•	
Update Requirements Compliance Matrix	19											X					X	X		•

1	•																		
2		•									X								
3			•								X							X	
4				•															
5	X	X	X		•														
6	X		X		X	•	X	X	X	X				X					
7	X		X			X	•	X											
8		X			X		X	•			X			X					
9						X			•	X	X			X				X	
10	X		X			X			X	•				X				X	
11	X	X	X		X						•								
12	X	X	X									•	X						
13	X		X			X			X			X	•						
14													X	•	X				
15												X	X	X	•				
16	X	X														•			
17					X	X	X	X									•		
18									X	X	X							•	
19											X					X	X		•

Critical Design Review

Electrical Peer Design Review	1	•						
Update Design Documents	2	X	•					
Hold Critical Design Review	3		X	•				
Action Item Review & Response	4			X	•			
Update Design Documentation	5				X	•	X	
Finalize Parts List	6					X	•	
Release Documentation to Vault	7					X	X	•

Board Route & Fabrication

	#												
Release Design Routing Package	1	•											
Routing Set-up (Drafting)	2	X	•	X	X								
Review Final Component Placement	3	X	X	•	X								
Review Critical Routes	4	X			•								
Complete Routing	5			X	X	•							
Final Routing Review	6					X	•						
Incorporate Routing Review Action Items	7						X	•					
Routing Documentation Review	8							X	•				
Release Data Package to Vault	9								X	•			
Release Test Spec	10									X	•		
PWB Fabrication	11									X		•	
Assemble First Piece	12											X	•

APPENDIX C: DATA FROM CASE STUDY #1

Project A

Board #1 — Placements

Date	Days	#
1/23/06	-	1
1/25/06	2	2
1/26/06	3	3
2/3/06	11	4
2/10/06	18	5
DAYS/REV	3.6	

Board #2 — Placements

Date	Days	#
2/3/06	-	1
2/10/06	7	2
2/17/06	14	3
2/21/06	18	4
DAYS/REV	4.5	

Board #3 — Placements

Date	Days	#
11/10/05	-	1
11/15/05	5	2
11/17/05	7	3
11/18/05	8	4
11/21/05	11	5
11/23/05	13	6
11/28/05	18	7
12/5/05	25	8
12/14/05	34	9
12/19/05	39	10
12/22/05	42	11
1/13/06	54	12
1/17/06	58	13
1/19/06	60	14
DAYS/REV	4.3	

Board #4 — Placements

Date	Days	#
11/16/05	-	1
11/17/05	1	2
11/23/05	7	3
12/15/05	29	4
12/21/05	35	5
12/22/05	36	6
1/17/06	52	7
1/31/06	66	8
2/1/06	67	9
DAYS/REV	7.4	

Board #5 — Placements

Date	Days	#
11/22/05	-	1
11/29/05	7	2
1/26/06	55	3
2/1/06	61	4
2/7/06	67	5
DAYS/REV	13.4	

AVG DAYS/REV 6.6

Project B

Board #1 — Placements

Date	Days	#
9/12/05	-	1
9/21/05	9	2
9/23/05	11	3
9/26/05	14	4
10/4/05	22	5
10/5/05	23	6
DAYS/REV	3.8	

Board #2 — Placements

Date	Days	#
6/13/05		

Board #3 — Placements

Date	Days	#
7/13/05	-	1
7/18/05	5	2
7/20/05	7	3
7/21/05	8	4
7/25/05	12	5
DAYS/REV	2.4	

Board #4 — Placements

Date	Days	#
11/2/05	-	1
11/7/05	5	2
11/9/05	7	3
11/14/05	12	4
11/16/05	14	5
11/18/05	16	6
11/29/05	27	7
DAYS/REV	3.9	

Board #5 — Placements

Date	Days	#
7/15/05		

AVG DAYS/REV 3.4

Project C

Board #1

NetLists		
10/28/04	-	1
11/12/04	15	2
11/23/04	26	3
12/14/04	47	4
12/16/04	49	5
1/14/05	78	6
1/18/05	82	7
1/26/05	90	8
1/31/05	95	9
2/16/05	111	10
3/15/05	138	11
3/17/05	140	12
Placements		
7/1/04	-	1
11/17/04	139	2
12/16/04	168	3
NB: Backplane Design		
DAYS/REV	56.0	

Board #2

NetLists		
2/14/05	-	1
2/17/05	3	2
2/28/05	14	3
3/3/05	17	4
3/14/05	28	5
Placements		
2/25/05	0	1

Board #3

NetLists		
3/3/05	-	1
4/15/05	43	2
5/13/05	71	3
Placements		
4/15/05	0	1

Board #4

NetLists		
3/9/05	-	1
3/11/05	2	2
3/18/05	9	3
4/1/05	23	4
4/4/05	26	5
4/11/05	33	6
5/4/05	56	7
5/19/05	71	8
5/24/05	76	9
6/3/05	86	10
6/14/05	97	11
7/25/05	138	12
7/26/05	139	13
Placements		
4/1/05	-	1
4/12/05	11	2
6/9/05	69	3
6/15/05	75	4
DAYS/REV	18.8	

Board #5

NetLists		
3/21/05	-	1
3/25/05	4	2
3/28/05	7	3
3/29/05	8	4
4/13/05	23	5
4/17/05	27	6
4/25/05	35	7
4/28/05	38	8
5/2/05	42	9
5/6/05	46	10
5/10/05	50	11
5/18/05	58	12
5/23/05	63	13
5/27/05	67	14
5/27/05	67	15
6/3/05	74	16
7/1/05	102	17
Placements		
5/6/05	-	1
5/16/05	10	2
6/3/05	28	3
6/13/05	38	4
DAYS/REV	9.5	

Board #6

NetLists		
4/19/05	-	1
4/28/05	9	2
5/12/05	23	3
6/6/05	48	4
6/21/05	63	5
6/30/05	72	6
7/18/05	90	7
7/27/05	99	8
8/4/05	107	9
8/19/05	122	10
8/24/05	127	11
8/26/05	129	12
9/1/05	135	13
9/9/05	143	14
9/12/05	146	15
9/14/05	148	16
Placements		
5/26/05	-	1
6/2/05	7	2
6/9/05	14	3
6/15/05	20	4
6/16/05	21	5
6/22/05	27	6
7/11/05	46	7
7/12/05	47	8
7/13/05	48	9
7/14/05	49	10
7/15/05	50	11
DAYS/REV	4.5	

Board #7

NetLists		
6/29/05	-	1
7/7/05	8	2
7/7/05	8	3
7/14/05	15	4
7/22/05	23	5
7/25/05	26	6
7/26/05	27	7
8/1/05	33	8
8/4/05	36	9
8/5/05	37	10
8/15/05	47	11
Placements		
7/22/05	-	1
7/25/05	3	2
7/29/05	7	3
DAYS/REV	2.3	

AVG DAYS/REV 8.8

Project D

Board #1 — Placements

Date	Days	#
10/25/04	-	1
11/24/04	30	2
12/16/04	52	3
1/10/05	67	4
1/19/05	76	5
1/24/05	81	6
1/24/05	81	7
DAYS/REV	11.6	

Board #2 — Placements

Date	Days	#
12/2/04	-	1
12/10/04	8	2
12/20/04	18	3
DAYS/REV	6.0	

AVG DAYS/REV 8.8

Project E

Board #1 Placements			Board #2 Placements			Board #3 Placements			Board #4 Placements			Board #5 Placements		
9/23/04	-	1	9/28/04	-	1	9/24/04	-	1	9/28/04	-	1	9/20/04	-	1
9/24/04	1	2	10/4/04	6	2	9/28/04	4	2	10/13/04	15	2	10/13/04	23	2
9/30/04	7	3	10/14/04	16	3	9/30/04	6	3	10/26/04	28	3	10/29/04	39	3
10/22/04	29	4	10/22/04	24	4	10/22/04	28	4	10/28/04	30	4	11/3/04	44	4
10/25/04	32	5	10/25/04	27	5	10/26/04	32	5	11/3/04	36	5	11/5/04	46	5
			10/27/04	29	6				11/5/04	38	6	11/7/04	48	6
			11/5/04	38	7							11/11/04	52	7
DAYS/REV	6.4		DAYS/REV	5.4		DAYS/REV	6.4		DAYS/REV	6.3		DAYS/REV	7.4	

AVG DAYS/REV 6.4

Project F

Board #1 Placements			Board #2 Placements			Board #3 Placements			Board #4 Placements		
9/22/04	-	1	8/20/04	-	1	10/1/04	-	1	9/2/04	-	1
9/24/04	2	2	8/27/04	7	2	10/11/04	10	2	9/27/04	25	2
10/7/04	15	3	9/23/04	34	3	10/18/04	17	3	10/7/04	35	3
10/13/04	21	4	9/29/04	40	4	10/28/04	27	4	10/20/04	48	4
10/24/04	32	5	10/7/04	48	5	11/5/04	35	5	10/22/04	50	5
10/25/04	33	6				11/19/04	49	6	10/26/04	54	6
DAYS/REV	5.5		DAYS/REV	9.6		DAYS/REV	8.2		DAYS/REV	9.0	

AVG DAYS/REV 8.1

Project G

Board #1 Placements		
1/2/03	-	1
1/15/03	13	2
1/16/03	14	3
1/17/03	15	4
1/20/03	18	5
DAYS/REV	3.6	

AVG DAYS/REV N/A

Project H

Board #1 Placements		
8/20/04	-	1
8/27/04	7	2
9/23/04	34	3
9/29/04	40	4
10/7/04	48	5
DAYS/REV	9.6	

AVG DAYS/REV N/A

	Placements									
	Project	**Board #**	**Board Type**	**# Revs**	**Rev Rate (Days/Rev)**	**Drafter**	**EE-Drafting Dist (Miles)**	**EE-ME Dist (Miles)**	**ME-Drafting Dist (Miles)**	**Total Distance**
1	A	1	A	5	3.6	A	7	7	0	14
2	A	2	B	4	4.5	D	7	7	0	14
3	A	3	C	14	4.3	E	1000	1000	7	2007
4	A	4	D	9	7.4	C	7	7	0	14
5	A	5	D	5	13.4	C	7	7	0	14
6	B	1	E	6	3.8	B	0	7	7	14
7	B	3	F	5	2.4	E	7	7	0	14
8	B	4	G	7	3.9	G	0	7	7	14
9	C	4	H	4	18.8	B	7	7	0	14
10	C	5	I	4	9.5	A	7	7	0	14
11	C	6	D	11	4.5	G	0	7	7	14
12	C	7	J	3	2.3	B	7	7	0	14
13	D	1	K	7	11.6	F	7	0	7	14
14	D	2	C	3	6.0	E	7	7	0	14
15	E	1	L	5	6.4	H	52.5	0	52.5	105
16	E	2	L	7	5.4	H	52.5	0	52.5	105
17	E	3	M	5	6.4	H	52.5	0	52.5	105
18	E	4	M	6	6.3	H	52.5	0	52.5	105
19	E	5	P	7	7.4	H	52.5	0	52.5	105
20	F	1	F	6	5.5	B	7	0	7	14
21	F	2	E	5	9.6	F	7	0	7	14
22	F	3	H	6	8.2	A	0	0	0	1
23	F	4	N	6	9.0	G	7	0	7	14
24	G	1	O	5	3.6	I	0	0	0	1
25	H	1	E	5	9.6	D	0	0	0	1

APPENDIX D: DATA FROM CASE STUDY #2

	Localized Team ECA #1				Localized Team ECA #2				Localized Team ECA #3			
	Design Days	Design %	PWB/Pkg Days	PWB/Pkg %	Design Days	Design %	PWB/Pkg Days	PWB/Pkg %	Design Days	Design %	PWB/Pkg Days	PWB/Pkg %
3/30/04	59	36%	15	0%	36	25%	10	0%				
4/19/04	94	23%	14	0%	222	52%	10	100%				
6/21/04	94	23%	14	0%	222	52%	10	100%				
6/23/04	95	24%	14	0%	222	51%	10	100%				
7/28/04	159	59%	30	0%	69	48%			74	43%		
8/2/04	159	59%	30	0%	67	56%			74	46%		
8/5/04	159	59%	30	0%	67	56%			74	46%		
8/12/04	171	54%	30	0%	74	57%			80	47%		
8/19/04	174	54%	30	0%	74	57%			80	47%		
8/27/04	189	51%	30	0%	78	64%			80	47%		
9/8/04	189	51%	30	0%	78	64%			80	47%		
9/20/04	161	89%	13	0%	83	94%			76	51%		
9/28/04	163	88%	13	0%	90	94%			76	86%		
10/5/04	158	94%	15	0%	91	95%			76	89%		
10/12/04	158	94%	1	0%	91	95%			76	89%		
10/19/04	179	82%	1	0%	119	57%			86	82%		
10/25/04	179	82%	1	0%	113	64%			99	71%		
11/2/04	179	88%	1	0%	113	64%			86	82%		
11/9/04	183	81%			124	85%			110	75%		
11/22/04	186	99%			139	87%			119	89%		
11/29/04	191	98%			139	85%			121	89%		
12/14/04	202	96%			151	88%			131	89%		
12/20/04	207	99%			151	93%			133	92%		
1/5/05	207	99%			151	93%			133	92%		
1/10/05	207	99%			158	92%			133	92%		
1/14/05	221	99%			158	95%			142	97%		
1/20/05	221	99%			161	94%			148	96%		
1/27/05	221	99%			162	98%			148	96%		
2/3/05	237	99%			174	98%			158	98%		
2/10/05	238	99%			175	99%			153	98%		
2/17/05	238	99%			175	99%			163	98%		
2/25/05					185	99%						

	Distributed Team ECA #4				Distributed Team ECA #5			
	Design Days	Design %	PWB/Pkg Days	PWB/Pkg %	Design Days	Design %	PWB/Pkg Days	PWB/Pkg %
3/30/04	41	26%	10	0%	65	1%	20	0%
4/19/04	77.5	78%	10	0%	43	65%	10	0%
6/21/04	77.5	78%	10	0%	43	65%	10	0%
6/23/04	77.5	78%	10	0%	43	65%	10	0%
7/28/04	115	41%	20	0%	129	38%	30	0%
8/2/04	115	41%	20	0%	129	38%	30	0%
8/5/04	115	41%	20	0%	129	38%	30	0%
8/12/04	118	47%	20	0%	128	47%	30	0%
8/19/04	118	47%	20	0%	136	47%	30	0%
8/27/04	127	47%	20	0%	143	44%	30	0%
9/8/04	127	47%	20	0%	143	44%	30	0%
9/20/04	125	81%	15	0%	134	74%	15	0%
9/28/04	125	82%	15	5%	135	74%	15	5%
10/5/04	125	82%	15	10%	135	74%	15	5%
10/12/04	130	80%	20	10%	145	73%	25	25%
10/19/04	139	75%	21	50%	164	51%	36	25%
10/25/04	139	75%	21	50%	164	51%	36	25%
11/2/04	139	75%	21	50%	164	62%	36	25%
11/9/04	146.5	90%	21	50%	160	70%	38	66%
11/22/04	145	98%			163	89%		
11/29/04	147	98%			162	89%		
12/14/04	160	98%			169	97%		
12/20/04	162	98%			175	97%		
1/5/05	162	98%			179	97%		
1/10/05	162	98%			189	97%		
1/14/05	153	100%			189	97%		
1/20/05					189	97%		
1/27/05					191	99%		
2/3/05					199	99%		
2/10/05								
2/17/05								
2/25/05								

	Distributed Team ECA #6				Localized Team ECA #7*			
	Design Days	Design %	PWB/Pkg Days	PWB/Pkg %	Design Days	Design %	PWB/Pkg Days	PWB/Pkg %
3/30/04	66	1%	15	0%	40	13%	10	0%
4/19/04	66	1%	15	0%	51	83%	10	100%
6/21/04	66	1%	15	0%	51	83%	10	100%
6/23/04	66	1%	15	0%	51	83%	10	100%
7/28/04	122	44%	29	0%	67	68%		
8/2/04	122	44%	29	0%	67	68%		
8/5/04	122	44%	29	0%	67	69%		
8/12/04	122	51%	29	0%	67	77%		
8/19/04	130	51%	29	0%	76	79%		
8/27/04	137	48%	29	0%	80	67%		
9/8/04	137	48%	29	0%	80	67%		
9/20/04	131	78%	15	0%	92	92%		
9/28/04	131	78%	15	0%	99	92%		
10/5/04	131	78%	15	0%	99	92%		
10/12/04	141	80%	17	25%	99	92%		
10/19/04	148	63%	18	25%	128	58%		
10/25/04	148	63%	18	25%	128	58%		
11/2/04	148	70%	18	25%	128	71%		
11/9/04	167	71%	37	80%	130	88%		
11/22/04	164	89%			145	95%		
11/29/04	164	88%			145	95%		
12/14/04	170	95%			149	96%		
12/20/04	175	97%						
1/5/05	177	97%						
1/10/05	188	96%						
1/14/05	184	98%						
1/20/05	190	98%						
1/27/05	190	98%						
2/3/05	200	98%						
2/10/05	204	98%						
2/17/05	204	98%						
2/25/05								

	ECA #1	ECA #2	ECA #3	ECA #4	ECA #5	ECA #6	ECA #7*
Actual PWB Days		20		55	55	41	20
Working Days		14.3		39.29	55	41	14.3
ECA Days	237	303	170	259	303	296	259
ECA Working Days	169	206	111	185	206	201	185

Note: Completion of ECA #7 extended due to schedule slack. Not included in ECA data set.

REFERENCES

[1] T. J. Allen. *Managing the Flow of Technology: Technology Transfer and the Dissemination of Technological Information with the R and D Organization.* MIT Press, Cambridge, MA, 1984.

[2] M. Blume and A. W. Appel. Heirarchical modularity. *ACM Transactions on Programming Languages and Systems*, 21(4):813–847, July 1999.

[3] T. Bollinger. The interplay of art and science in software. *Computer*, pages 125–128, October 1997.

[4] T. Browning. Integrative mechanisms for multiteam integration: Findings from five case studies. *Systems Engineering*, 1(2):95–112, 1998.

[5] T. Browning. Applying the design structure matrix to system decomposition and integration problems: A review and new directions. *IEEE Transactions on Engineering Management*, 48(3):292–306, August 2001.

[6] T. Browning and S. D. Eppinger. Modeling impacts of process architecture on cost and schedule risk in product development. *IEEE Transactions on Engineering Management*, 49(4):428–442, 2002.

[7] H. W. Chesbrough and K. Kusunoki. The modularity trap: Innovation, technology phase shifts and the resulting limits of virtual organizations. In I. Nonaka and D. J. Teece, editors, *Managing Industrial Knowledge: Creation, Transfer and Utilization*, chapter 10, pages 202–230. Sage Publications, London, U.K., 2001.

[8] E. F. Crawley. Creating the CDIO syllabus, a universal template for engineering education. *Frontiers in Education*, 2:8–13, 2002.

[9] S. Denker, D. Steward, and T. Browning. Planning concurrency and managing iteration in projects. *Center for Quality Management Journal*, 8(2):55–61, Autumn 1999.

[10] I. Edward F. McDonough, K. B. Kahn, and A. Griffin. Managing communication in global product development teams. *IEEE Transactions of Engineering Management*, 46(4):375–386, November 1999.

[11] S. Eppinger, D. Whitney, R. P. Smith, and D. A. Gebala. A model-based method for organizing tasks in product development. *Research in Engineering Design*, 6:1–13, 1994.

[12] S. D. Eppinger. Innovation at the speed of information. *Harvard Business Review*, 79(1):149–158, January 2001.

[13] D. Ernst. Limits to modularity: Reflections on recent developments in chip design. *Industry and Innovation*, 12(3):303–335, September 2005.

[14] B. A. Farshchian. Integrating geographically distributed development teams through increased product awareness. *Information Systems*, 26:123–141, 2001.

[15] T. Friedman. *The World is Flat: A Brief History of the Twenty-First Century*. Farrar, Straus and Giroux, New York, NY, 2005.

[16] E. Fuchs and R. Kirchain. Changing paths: The impact of manufacturing off-shore on technology development incentives in the optoelectronics industry. In *Proceedings of 2005 Annual Meeting of the Academy of Management*, 2005.

[17] P. J. Hinds and D. E. Bailey. Out of sight, out of sync: Understanding conflict in distributed teams. *Organization Science*, 14(6):615–632, November-December 2003.

[18] N. Joglekar, A. Yassine, S. Eppinger, and D. Whitney. Performance of coupled product development activities with a deadline. *Management Science*, 47(12):1605–1620, December 2001.

[19] H. Kerzner. *Project Management: A Systems Approach to Planning, Scheduling, and Controlling.* Wiley, New York, NY, 2001.

[20] J. Kostner. *Virtual Leadership: Secrets from the Round Table for the Multi-Site Manager.* Warner Books, New York, NY, 1994.

[21] A. MacCormack, J. Rusnak, and C. Baldwin. Exploring the structure of complex software designs: An empirical study of open source and proprietary code. Forthcoming in Management Science, 2006.

[22] A. A. Oloufa, Y. A. Hosni, M. Fayez, and P. Axelsson. Using DSM for modeling information flow in construction design projects. *Civil Engineering and Environmental Systems*, 21(2):105–125, June 2004.

[23] M. E. Sosa, S. D. Eppinger, and C. M. Rowles. Identifying modular and integrative systems and their impact on design team interactions. *Transactions of the ASME*, 125:240–252, June 2003.

[24] D. V. Steward. The design structure system: A method for managing the design of complex systems. *IEEE Transactions on Engineering Management*, 28:71–14, 1981.

[25] K. T. Ulrich and S. D. Eppinger. *Product Design and Development.* McGraw Hill, New York, NY, 3rd edition, 2004.

[26] D. Whitney. Designing the design process. *Research in Engineering Design*, 2:3–13, 1990.

[27] A. Yassine, D. Falkenburg, and K. Chelst. Engineering design management: An information structure approach. *International Journal of Production Research*, 37(13):2957–2975, May 1999.

[28] A. Yassine, N. Joglekar, D. Braha, S. Eppinger, and D. Whitney. Information hiding in product development: The design churn effect. *Research in Engineering Design*, 14(3):145–161, November 2003.

[29] A. Yassine, D. Whitney, S. Daleiden, and J. Lavine. Connectivity maps: Modeling and analyzing relationships in product development processes. *Journal of Engineering Design*, 14(3):377–394, September 2003.