INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

**Managerial Coordination between Hardware and
Software Development during Complex Electronic
System Design**

by

Christine Koch, B. B. A.

*A thesis submitted to the Faculty of Graduate Studies and
Research in partial fulfillment of the requirements for the
degree of*

Masters of Management Studies

School of Business

Carleton University

Ottawa, Ontario

May, 1997

Copyright © 1997, Christine Koch

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-22135-0

The undersigned recommend to the Faculty of Graduate Studies and Research acceptance
of the thesis

MANAGERIAL COORDINATION BETWEEN HARDWARE AND SOFTWARE
DEVELOPMENT DURING COMPLEX ELECTRONIC SYSTEM DESIGN

submitted by Christine Koch in partial fulfillment of the requirements for the degree of
Master of Management Studies.

A handwritten signature in cursive script, reading "J. Callahan", written over a horizontal line.

Thesis Supervisor

A handwritten signature in cursive script, reading "Vinod Kumar", written over a horizontal line.

Director, School of Business

School of Business
Ottawa, Ontario

May 7, 1997

Abstract

Companies that design systems which include both hardware and software components are finding that parallel design techniques are being increasingly employed as software becomes a larger component of complex electronic systems.

This study investigated hardware-software coordination, from a managerial versus technical perspective, by developing a questionnaire which was administered to 18 projects from companies in complex electronic system design industries.

A model testing coordination mechanisms and their effects on project success was developed. The results showed that two of the hypothesized organizational-level coordination mechanisms, number of levels to common report and physical distance between hardware and software design teams, were found to be correlated with project success. Two hypothesized variables which were thought to help coordinate interdependent tasks, developers that had mutual hardware and software design experience and shared design tools, were negatively correlated with project success.

Table of Contents

List of Tables.....	3
Acknowledgments.....	5
1. Thesis Overview.....	6
2. The Design of Complex Electronic Systems	9
2.1 Literature on the Design of Complex Electronic Systems	9
2.2 Interviews with Industry Experts.....	10
Data Collection Process:	10
Industry experts' comments:.....	11
Coordination Problems:	13
2.3 Case Study.....	15
The Company:	15
The Design Project:.....	16
Important Events:	16
The Data Collection Process:.....	19
Organization of Project:	19
Hardware-Software Development Coordination:.....	21
Coordination Mechanisms Used in the Case Study:.....	23
3. Theory Development	25
3.1 Coordination in Product Development Projects	25
Differentiation:.....	25
Interdependence:.....	26
Cross-functional Coordination:.....	28
3.2 A Model of Coordination of Hardware and Software in Complex Electronic System Design	31
Independent Variables:	32
Dependent Variables:.....	37
Project Cost.....	37
Product Design Cycle Goals:	37
Product Functionality Goals:.....	38
4. The Questionnaire Survey	39
4.1 The Questionnaire	39
Measures:.....	40
4.2 The Survey Sample.....	42
Sample Selection Process:.....	42
Participating Companies:.....	44
Survey Response:	45
4.3 Survey Data Set Description	45
Descriptive Data for Projects.....	45
Coding the Data Set:	46
Exploratory Data Analysis:.....	49
Summary of Exploratory Data Analysis:	52

5. Survey Data Analysis	55
5.1 Similarities and Differences Among Survey Participants	55
Part One: Comparing Means	55
Part Two: Correlations.....	59
5.2 Individual Respondent Data Set Correlations.....	61
5.3 Creating the Project Data Set	69
Exploratory Data Analysis: Project Data Set of 18 cases	71
5.4 Testing the Model.....	77
Hypotheses Test Results:.....	86
6. Discussion and Conclusions.....	91
Discussion of Findings.....	91
Recommendations for Managers	93
Recommendations for Future Research	94
References	97
Appendix A: One Page Summary of Research and Survey sent to Participating Organizations.....	100
Appendix B: Measures for Project Data Set Variables	102
Appendix C: Project Manager, Hardware and Software Developer Questionnaires.....	103

List of Tables

Table 1: Detailed Timeline of Events and Milestones in Case.....	18
Table 2: Descriptive Information on Companies	44
Table 3: Descriptive Data for Projects.....	46
Table 4: Variable Label Prefixes	47
Table 5: Respondent Demographic Variables.....	47
Table 6: Organization Demographic Variables	48
Table 7: Project Demographic Variables.....	48
Table 8: Independent Variables from Questionnaire	48
Table 9: Dependent Variables from Questionnaire.....	48
Table 10: Frequencies for Independent Variables.....	50
Table 11: Frequencies for Dependent Variables	50
Table 12: Histograms on Independent Variables	51
Table 13: Histograms for Dependent Variables	51
Table 14: Histograms for Organization Demographic Variables.....	52
Table 15: Histograms for Project Demographic Variables.....	52
Table 16: Comparison of Means: Respondent Type and Success Variables.....	56
Table 17: Comparison of Means	58
Table 18: Correlations between independent and dependent variables, by respondent type.....	59
Table 19: Correlations between Independent Variables Dependent Variable Cost Goal Success.....	62
Table 20: Correlations between Independent Variables and Dependent Variable Design Cycle Goal Success	63
Table 21: Correlations between Independent Variables and Dependent Variable Functionality Goal Success	64
Table 22: Correlations between Independent Variables and Project Management Support from the Developers' Perspective	65

Table 23: Correlations between Independent Variables and Project Manager Support from the Managers' Perspective.....	66
Table 24: Correlations between Communication Variables and other Independent Variables.....	67
Table 25: Correlations between Independent Variables and Dependent Variable Enjoyment of Work.....	68
Table 26: Organizational Demographic Variables for Project Data Set	69
Table 27: Project Demographic Variables for Project Data Set	69
Table 28: Independent Variables for Project Data Set	70
Table 29: Dependent Variables for Project Data Set	71
Table 30: Summary Statistics for Dependent Variables from a Project Managers' Perspective Project Data Set.....	72
Table 31: Summary Statistics for Dependent Variables from Hardware and Software Developers' Perspective.....	72
Table 32: Summary Statistics for Independent Variables – Project Data Set	73
Table 33: Summary Statistics for Other Demographic Variables – Project Data Set.....	74
Table 34: Independent Variables examined with Histograms – Project Data Set	74
Table 35: Dependent Variables examined with Histograms – Project Data Set.....	75
Table 36: Independent Variables used to test the Theoretical Model.....	78
Table 37: Comparison of Means for Successful and Unsuccessful Projects.....	79
Table 38: Correlations – Independent Variables and Combined Success Measures	80
Table 39: Correlations – Independent Variables with Independent Success Measures.....	81
Table 40: Spearman Correlations – individual task difficulty measures	82
Table 41: Aggregate Task Difficulty Measures with Project Success	83
Table 42: Spearman Correlations with Model Variables and Moderating Variable Developers' Perception of Task Difficulty.....	84
Table 43: Spearman Correlations with Model Variables and Moderating Variable Project Managers' Perception of Task Difficulty.....	85

Acknowledgments

Special thanks are due to my thesis supervisor. Professor John Callahan guided me throughout my thesis research, gave me the opportunity to present a paper at the Fifth International Conference on Management of Technology in Miami, Florida, and provided me with generous funding.

Special thanks are also due to my family, first and foremost to my husband John who supported me in the ups and downs and who proofread my work on many occasions. I also want to thank my parents and brother for their support.

Lastly I would like to thank my committee members Tony Bailetti and Shirley Mills for their cogent input.

1. Thesis Overview

The complex electronic systems design market is changing rapidly in two directions. In the telecommunication, aviation, and other complex electronic systems design industries, competition is intensifying, while at the same time the complexity of these systems is increasing. The complexity of these systems is driven by the integration of hardware and software components at the product design stage. The coordination of the hardware and software components of a complex electronic system is difficult for two main reasons: both the design processes for hardware and software components and the educational and work experience of hardware and software developers are different.

This study investigated the differences and similarities between hardware and software developers and the processes they use during product design. The study's sample included 18 projects from 15 different companies. Three to four respondents from each project participated for a total of 55 respondents. Participating companies were chosen on the basis of convenience.

Reasons for hardware and software coordination problems during product development were identified. This led to an investigation of coordination mechanisms between hardware and software developers used during product development. These coordination mechanisms were identified and used to build an explanatory model in order to understand their effect on project success. This model was then tested by means of a quantitative survey study.

In order to better understand the design of a complex electronic system it was necessary to research this environment in two stages: interviews with industry experts and then a case study. The case study allowed for a more in-depth understanding of how hardware and software developers work with each other and what coordination mechanisms are used.

It was concluded that differences between hardware and software developers can make it difficult for project managers to coordinate the two design functions during product design. In order to identify effective coordination mechanisms, and thereby help product development managers make better decisions during the product development process, it was necessary to:

- understand the differences and similarities between hardware and software developers,
- identify the origins of hardware/software coordination problems,
- identify mechanisms used to coordinate hardware and software developers and
- investigate the effectiveness of these coordination mechanisms.

Literature on coordination in product development explains the importance of differentiation, interdependence and cross-functional coordination. From this literature review coordination mechanisms were identified. A research model and hypotheses were developed and tested in a questionnaire survey.

The quantitative survey results indicated that coordination mechanisms at the organizational level are correlated with coordination and reduction of negative outcomes of differentiation between hardware and software developers. Conversely, some initiatives to manage interdependence did not correlate with

project success. One explanation for the results of organizational-level initiatives may be their ability to decrease the negative outcomes of design team differentiation by decreasing conflict, barriers and misunderstandings. With regards to managing interdependence, using developers with both hardware and software design experience does not positively correlate with project success. It may better suit managers to facilitate orientation towards specific expertise, rather than mutual experience.

2. The Design of Complex Electronic Systems

2.1 Literature on the Design of Complex Electronic Systems

Coordination during complex electronic system design is an increasingly important issue for high-tech industry managers. Literature on complex system design describes how design tasks are becoming extremely complex. While the systems being designed are increasing in complexity, developers are also having to adjust to new design tools at the same time.

Kalavade and Lee (1993) state that "the development of large electronic systems is becoming too complex to be designed by the traditional method of partitioning the hardware and software design processes where there is a lack of a unified representation, simulation and synthesis framework." Tough design cycle, product performance, and quality goals for such systems gives designers more reason to use systematic design methods. Co-design, for example, is a design strategy in which "hardware and software designs proceed in parallel, with feedback and interaction between the two" (Kalavade and Lee, 1993). Within this type of environment, special design tools, for example simulators, emulators, prototypes, etc., are needed in order to unify software and hardware representation, synthesis and simulation (Kalavade and Lee, 1993).

At the same time system design project goals become increasingly complex as hardware and software design processes are changing with regards to new technologies. New versions of design tools, for example computer-aided design (CAD) or electronic design automation (EDA), are constantly being introduced to

which developers need to adapt. During the design process, hardware developers use CAD or EDA tools which allow them to capture, implement and validate the hardware design. This process of entry, implementation and validation is iterative, allowing hardware developers to obtain optimization of hardware components much more quickly than software developers (Fawcett, 1993). Software developers use a variety of computer-aided software engineering (CASE) tools which enable the software developer to plan, specify, develop and evolve their programs.

Because hardware developers often finish design cycles before software developers, managing design cycle schedules between the two functions is difficult, as is managing design teams which work with different design tools and languages. These differences are especially evident during project meetings where each development team needs to understand design issues from each other's perspective.

2.2 Interviews with Industry Experts

Interviews examined differentiation and coordination problems between hardware and software development. Interview respondents were recruited by posting requests on the Usenet newsgroup network and on the IEEE electronic newsletter. Project managers, hardware developers and software developers were interviewed.

Data Collection Process:

Requests were posted to two newsgroups: comp.software-eng and comp.electronics, as well as the IEEE electronic newsletter for the Ottawa-Carleton

region. The requests were addressed to hardware and/or software developers and asked them to respond to the following questions:

- What are the main coordination problems between hardware and software developers during product development?
- What type of specifications and/or tools do hardware and software developers use to help coordinate the two components during product development?

Eight respondents replied by email with comments, providing useful information. These led to more contacts from high-tech organizations in the Ottawa-Carleton region. These were subsequently contacted and interviewed. The information gathered confirmed that coordination between hardware and software developers can be problematic.

Industry experts' comments:

One of the roots of the differences between hardware and software developers' understanding of each other's components are their differing design processes. Hardware and software components' functional optimizations are different. The sense of economy between hardware and software development processes is also completely different; hardware is designed for manufacturability while software is designed for the end user. Clive Smith (Sept. 12, 1994), a project management consultant, explained: "hardware is a material economy, based on cost of goods. Software can be large and cheap at the same time. The economy here is more of a service economy, where maintenance of code is key, therefore elegance and organization is important."

Hardware and software developers' schedules are not synchronized. Hardware takes less time to update than software. After several discussions with hardware and software developers, it was noted that hardware engineers benefit from a greater availability of standard components, and thus they spend less time inventing new parts for a system. Software developers spend much more time inventing new parts (programming) and have little access to already established libraries.

Both the product hardware and software designs can constrain each other. As one hardware/software developer stated, "if the hardware can only resolve +/- 10mV then the software decisions are constrained by that resolution."

One firmware developer described a multi-site situation where it was necessary to communicate weekly by videoconferencing. He explained that people worked well together when personal relationships had been established, even in a situation where one developer was situated in another city and video conferencing was required for communication. When a relationship had not been established and two developers needed to communicate by video conferencing, the two developers often did not work well together -- video conferencing did not create the personal relationships required for team work. He further explained that no amount of technology could substitute for workers who are cooperative and willing to work well together. Many disagreements and arguments grew out of this multi-site situation, requiring open-minded workers to solve.

Coordination Problems:

The informal interviews and discussions with hardware and software developers confirmed that there are coordination problems between the two functions which stem from differences between the two groups. Reasons for coordination problems between hardware and software developers were identified:

- Insufficient communication (both formal and informal) between the hardware and software developers during the design process.
- An insufficient number of activities prompted by management which enhance communication and coordination between the hardware and software teams.
- Lack of hardware and software developers' understanding of each others' environment.

Smith (Sept. 12, 1994) explained how this lack of understanding is "based on things such as education, paradigm, ways of solving problems and approaches to optimization. This last point is key -- the software engineer thinks that optimization means making the hardware friendly to the software environment. The hardware engineer thinks that optimization is when the hardware is cheap and efficient cost-wise and production-wise, and feels that the software engineers should simply write more complex code to handle the simpler hardware. The two (hardware and software development teams) simply cannot understand why their own demands of the other are unreasonable."

Another electronic systems developer described the situation as "sometimes the lack of understanding borders on contempt for the other. Hardware folks are often more software-savvy than software folks are hardware-savvy. Although the hardware developers can program, they do not grasp the need for discipline in the

software development process that they would insist on for hardware development. Too often one hears 'software is easy to change, so I should not have to argue, even very late in the project cycle'."

- No common feeling of unity between the hardware and software developers.

One software/hardware developer explained, "All too often the hardware and software are done in different sub-organizations in different locations. This makes it really difficult to maintain a common focus. Again, hardware often times gets 'done' earlier in the cycle than software and later when assistance is needed, the hardware engineer has to become reacquainted with the project and why it is important."

Smith's comments on this coordination problem were that "this is because during the early stages, each [the hardware development team and the software development team] resents the other side's demands. Right from the start, each is making the other work harder -- more complex hardware, or more complex software. The result is a feeling that the other side is narrow-minded, or lazy, or both. Then during testing, the question always is 'is this a hardware problem or a software problem?' Often it is both and neither."

Another electronic systems developer commented that "there is often a lack of 'systems view' which tends to isolate the two domains from each other. It is often beneficial to carry out systems design as far as possible before requirements are allocated to either hardware or software."

- Poor management regarding the diversion of resources and responses to slipping time schedules.

Regarding this coordination problem, Smith commented, "The basic rule is 'Software is always late'. This is no less true than Newton's laws of motion. Nevertheless, this issue again relates to the question of optimizing the global goals of the project. When software demands a hardware change, whose fault is it when the boards are late? And then because the boards are late, whose fault is it that testing was late, and so on . . . an 'us and them' mentality. The key is to engender a sense of global understanding of the project. Management must teach the global picture, especially the reason for doing the project."

2.3 Case Study

The information gathered from the literature review and developer interviews illustrated that the environment in which hardware and software developers work is unique and can be problematic during the integration stage. It is the nature of the differentiation and interdependence between hardware and software developers which directed the author to more closely examine various coordination mechanisms which may facilitate project success. As a case study, a specific examination of hardware-software coordination was conducted in a telecommunications systems supplier company. Coordination mechanisms used in this case study were identified and their perceived effectiveness was examined.

The Company:

The organization under study is in the telecommunications industry and is one of the industry's leading telecommunications switching product developers. This telecom organization comprised several software groups (for each of their products)

and one hardware group. The hardware group, organized as a separate business unit within the organization, is always contracted by software groups for new projects. The hardware group was described as 'its own company within the organization' by the case study participants.

The Design Project:

Product management's goal was to add a new circuit pack to an existing PBX platform that would be flexible enough so that future options could be added in software. Product management wanted future adaptation to the circuit pack to be inexpensive and 'stress free' for the customer and field support. With this objective in mind the first task for the project team was to decide whether or not this concept could be implemented using software only. If not, product management had to decide what would be needed from hardware. In the end it was decided that new hardware would be needed. An official software and hardware team was established.

The main reason a new hardware platform was needed was the complexity of future options this circuit pack was planned to run – an important feature for the customer. The new hardware design consisted of many more parts than the previous hardware platform. This made the design and layout process difficult. This new circuit pack was designed to keep current customers happy but also add innovative features that would attract new customers.

Important Events:

The first stage, research, was performed by a product management and a software team, and ran from April to November of 1992. Its objective was to

determine whether or not new hardware needed to be developed (discussed above).

The second stage -- detailed analysis and definition -- began in December, 1992, when the hardware team of two developers was added. The main stages of this project's life cycle were:

- research stage -- define requirements
- analysis and definition -- define hardware and software product requirements in more detail-- what the product should do
- design -- how the product should work
- detailed design -- software codes and hardware lays out blocks of logic
- integration -- test and evaluate -- ensure software and hardware do what they are supposed to do

Table 1: Detailed Timeline of Events and Milestones in Case

Date	Project Milestone
April '92	The beginning of the new circuit pack research phase – learning and determining product requirement.
Sept. '92	FAD (functional Analysis and Definition) – product requirements completed.
Jan. '93	First hw/sw/product mgt. group meeting about different options to use regarding the card – hardware begins DOS and software begins FAD.
Feb '93	Software FADs approved.
May '93	Hardware starts design.
June '93	Hardware has rough specifications.
July '93	CAD schematic captured – transfers to layout tool. Software begins firmware design.
July '93	Hardware group holds meeting – project at risk because of tools.
July '93 - Oct. '93	Hardware layout cycle.
Aug. '93	Software design complete.
Oct. 10, '93	Hardware layout complete and PCBs ordered.
Oct. '93	Software runs simulations – functionality testing complete.
Nov. 26, '93	Hardware verification complete.
Dec. '93	Integration and verify software and hardware.
Aug. - Sept. '94	System build and test.
Nov. '94	Project completed.

The Data Collection Process:

Data sources included: personal interviews and documentation, including minutes of meetings, letters, electronic messages, and design specifications. (Yin, 1989). Structured interviews were conducted with a software project manager, a project coordinator, a hardware developer and a software developer. These interviews lasted up to two hours each. Key questions included:

- To what extent have companies undertaken the following initiatives to improve the way that hardware and software teams work together?
- How often do hardware and software teams meet during project development?
- On average, to what degree are hardware developers able to see issues and problems from software developers' perspectives, and vice versa?
- How much support does senior management provide in order to improve the way that hardware and software teams work together?

Organization of Project:*Individuals:*

The software team's tasks revolved around both application and system level software. The software team had first to decide where the software was to be located. Second, they had to identify the interfaces. Finally, they had to program the algorithms and structure the data.

In designing the new hardware platform, the hardware team had an older design to work from. The hardware developer had to work with new parts but followed a familiar process to put the parts together. However, the new design consisted of many more parts than before. This made the design and layout process difficult. In the end the hardware design consisted of 20 blocks of logic and no

ASICs (application-specific integrated circuits) in the final design, which was a printed circuit board.

A project coordinator was responsible for keeping the project on schedule and within budget. This included ensuring the hardware was delivered to the software team on time.

Teams and Management:

Differences between the hardware and software design teams included: the way each team was organized, the size of teams, the developers' educational backgrounds and work experience, and their overall attitudes towards the project.

The hardware and software teams worked under different management groups. Each team had to report to different groups of supervisors as if they worked for two different 'organizations' at the same site.

The size of each team also differed; the hardware team was considerably smaller than the software team. It was more common for the hardware team to seek out other resources -- support people -- when in a difficult situation because the 'hardware organization' did not have as many developers to allocate to the project. This lack of resources was especially difficult for the hardware team because of the tools with which they were working -- they were new to the developers and created many problems.

The hardware and software developers interviewed had different educational backgrounds and work experiences. The hardware developer was educated in engineering and applied physics and studied some computer science. He also had

some experience in software design so he was able to understand the software developers' issues and concerns and felt comfortable communicating with them. The software developer had an educational background in computer science and no education in physics or engineering. This made it difficult for him to completely understand hardware issues. The software developer explained that he was a bit uncomfortable in meetings and when communicating with the hardware developers, as he did not fully understand the hardware development jargon used.

Attitudes towards the project also differed between the hardware and software developers. This was related to the types of problems each group faced. The hardware team worked with tools that created many unnecessary problems and frustration - something the software developers did not have to face. The software team's biggest worry through to the design stage was the question of when they would receive the hardware.

Hardware-Software Development Coordination:

An early managerial challenge was the lack of a detailed design document. The hardware developer did not have the time to write a formal and detailed design document thus making it more difficult for the software development team to fully understand their requirements and constraints.

The hardware and software developers were located within a five-minute walk of each other. Neither developer felt this distance was too great.

At the beginning of development, in order to coordinate the software team, the software manager held weekly group meetings to discuss everyone's status.

Respondents felt that these group meetings became less and less relevant as time went on and as more informal communication channels were employed.

The project coordinator used two methods of coordinating hardware and software developers: calling group meetings to ensure communication, and providing a shared view of how each team was progressing at different stages of the development cycle. When it was not possible to have group meetings due to work schedules, she would create a review document that provided a shared view of the project. This document was in the form of a spreadsheet which outlined where the hardware and software teams were in their design cycles, and what their milestone goals were. This information was enough to indicate what they needed and/or expected from each other in the near future.

Over the life of the project, the reasons why developers used project communication mechanisms changed. Initially, they were used to provide team members with basic information regarding project goals, then to provide project status reports, and finally to identify and solve project problems. The group meetings were effective when the software team did not have the hardware document to refer to, and also near the end of the project – when more problems surfaced.

In summary, there were differences between both design teams and the design processes they employed. Within the design teams hardware and software developers differentiated in terms of their educational and work experiences. In

terms of the design processes, hardware and software developers performed different tasks, and used different design languages and tools.

Coordination Mechanisms Used in the Case Study:

Senior Management Control:

Throughout the development cycle hardware and software developers were given substantial amounts of autonomy which allowed them to work and solve problems among themselves. They were not required to verify decisions with upper-level management unless unbudgeted costs were an issue.

Ability to Adjust to Each Other's Needs:

The hardware and software developers interviewed had different educational and work experience backgrounds. However, the hardware developer did have an understanding of software design and was therefore able to understand the issues that were important to the software developers.

Location:

The hardware and software teams were located in the same building, just a five minute walk from each other, and were able to coordinate their informal and formal meetings very easily. The hardware developer commented that "if they [software] were on the other side of town, that would have been a problem."

Success Measure for Project:

During the examined project, the project coordinator called many group meetings to facilitate communication between the two teams and to discuss project status. The hardware and software developers also had many informal meetings throughout the entire development process.

Shared Documentation and Design Tools:

The lack of the hardware design document seemed to make the design tasks for the software developer more difficult. The software developer was not able to refer to the hardware design document when he had questions because this document was being written by the hardware developer as the software developer needed it.

3. Theory Development

3.1 Coordination in Product Development Projects

Differentiation:

Complex electronic systems require that hardware and a software components work well together. The design and development of each type of component requires two types of developers with different educational backgrounds and expertise. The differences between these two types of developers are natural and necessary for the success of the product in the market place. McCann and Galbraith (1981) express such differences a “natural outcome of the division of labour.”

One way of defining differentiation is by the cognitive and emotional orientation among managers (and developers) in different functional departments. Lawrence and Lorsch (1967) posit that functional departments can differ along four separate dimensions: orientation toward particular goals, time orientation, interpersonal orientation and formality of structure of the department. McCann and Galbraith (1981) imply that differentiation between departments can lead to an increase in conflict which, in turn, can lead to negative outcomes and perceptual differences.

The personal skills and traits of hardware and software developers are also quite different. "Dalton (1959) and Thompson (1960) found that personal dissimilarities, such as background, values, education, age, and social patterns lowered the probability of interpersonal rapport between unit representatives, and in

turn decreased the amount of collaboration between their respective units" (Walton and Dutton, 1969: 77).

These two ways of thinking can be viewed as two different thought worlds (Dougherty, 1992; 180). A thought world is a community of persons engaged in a certain domain of activity who have a shared understanding about that activity in terms of two aspects: 1) what they know of the activity and 2) how they know it. Members of the same thought world all know about the typical procedures, constraints, goals, tools, language, etc. of their activities. This practical knowledge is developed by common educational and work backgrounds. Because of specialization, a certain thought world is likely to best understand certain issues, but also to ignore information that may be equally essential to the total task, such as information relevant to the thought worlds of other project team members.

Interdependence:

During complex electronic system design organizations often form project teams consisting of several functions, including software and hardware teams. Management processes are needed in order to address the negative effects of differentiation between different types of teams. Within these project teams the issue of managing interdependence is addressed.

"Interdependence is reciprocal in a situation where the outputs of each become inputs for the others" (Thompson, 1967: 55-56). Thompson also states that greater coordination effort is required when interdependence between two units is high. Reciprocal interdependence, when coordinated by mutual adjustment, is most

demanding of communication and decision effort (Thompson, 1967: 55-64). Reciprocal interdependence exists between the hardware and software design processes during development of complex electronic system products. Hardware and software development teams both produce parts of an overall system design where the outputs of both the hardware and software teams become the inputs of the other. The process of integrating each other's outputs is often referred to as the integration stage (of hardware and software components) of product development.

McCann and Galbraith (1981; 63-66) state that "interdependence implies that there is something of interest between parts. Two departments depend on each other for resources, work or information . . . interdependence, in part, is a function of the goals and performance measures of the departments at question . . . actions taken in one department affect the goal accomplishment of the other department." Walton and Dutton (1969; 73) have a similar definition where "task dependence is the extent to which two units depend upon each other for assistance, information, compliance, or other coordinate acts in the performance of their respective tasks." For example, it is important that the hardware developers inform the software developers on the amount of memory provided in the hardware component. The software developers will be constrained by the amount of memory in terms of the lines of code they program for the product's features. Software developers may feel that implementing part of the product's functionality into the hardware component would be more efficient and therefore need to inform the hardware developers of their reasoning. If important information is not communicated or if design tasks are

not coordinated in order to assure that issues regarding the system's design specifications are clear to both hardware and software development teams, conflict and problems will arise in the design process. Walton and Dutton (1969, 75) state that "conflict results when each of the interdependent departments has responsibility for only one side of the dilemma embedded in the organizational tasks." Information regarding the outcome of the hardware component is of crucial importance to the software team and vice versa. This high degree of interdependence is one reason why the hardware and software development process is difficult to manage. In this study, interdependence will be defined as the degree to which two functions depend on each other to complete their tasks (Thompson, 1967).

Cross-functional Coordination:

As Cooper and Kleinschmidt (1987) argue, cross-functional coordination is an important factor in the overall success of new product development. This is especially true during complex electronic system design, because the high levels of both differentiation and interdependence between the hardware and software development teams involved.

There is a considerable literature regarding specific means for encouraging cross-functional coordination during new product design. These usually involve integrating the work of different functional teams around individual projects. McCann and Galbraith and Thompson both discuss methods by which cross-functional coordination can be realized in these situations. By reviewing their work

and that of other authors, a list of coordination mechanisms appropriate to the hardware and software development environment was built. Because of its focus on the coordination of different functional teams during manufacturing design, Rusinko's 1992 results were found to be most relevant for this study, and were therefore applied to the complex electronic systems design process examined here.

The following six coordination mechanisms were identified for further analysis in a quantitative study:

1. Number of Levels to Common Report

Rusinko (1992; 36-37) summarizes the findings of other researchers on the importance of the number of levels to common report to function coordination: "Researchers maintain that the smaller the number of levels or the smaller the hierarchy between project level personnel and the first manager with authority over both functions, the greater the probability of project success . . . [these authors] recommend resolving conflict at the lowest competent level."

2. Rotation

Walton and Dutton (1969; 77) maintain that a method of reducing differentiation is to increase common experience, which can reduce communication barriers and provide common referents. McCann and Galbraith (1969; 76) explain how some organizations rotate their members to create informal networks that promote decentralized decision making and integrate values and actions.

3. Co-location

McCann and Galbraith (1969; 71-72) explain the importance of physically clustering departments. "This is a coordination strategy used to manage high-levels

of interdependence where the reduction of spatial barriers between highly interdependent people can reduce response times . . . by substituting more personal, flexible forms of communication and can help prevent conflict by encouraging friendships."

4. Group-based Evaluations

When a group (or project team consisting of hardware and software developers) is evaluated as a whole and incentives and bonuses are based upon group performances, and especially inter-group performances, cooperative behaviors are exhibited (McCann and Galbraith, 1981; 75). Walton and Dutton (1969; 75) state that "the more the evaluations and rewards of higher management emphasize the separate performance of each department rather than their combined performance, the more conflict."

5. Extent of Shared Design Documents

Rusinko (1992, 50) describes how shared design documents is a coordination mechanism that can ensure common understanding of project goals.

6. Extent of Shared Design Tools

The value of shared design tools, especially for simulation and testing, was found to be a useful method of ensuring team coordination from discussions with hardware and software developers and the case study.

3.2 A Model of Coordination of Hardware and Software in Complex Electronic System Design

The information obtained from the Request for Comments from Usenet groups and interviews with industry experts, the case study, coordination literature, and Rusinko's study helped to construct the following research model.

The research model consisted of seven independent variables and one set of dependent variables (cost, design cycle, and product functionality goal success).

Figure 1: Research Model Diagram

The unit of examination used to test this model was the completed project. However, in order to collect appropriate data, the aggregated results of responses from individual members of project teams were used.

Independent Variables:

The independent variables identified in the diagram above are applicable to the hardware-software environment in the development of complex electronics systems. These coordination mechanisms, described in section 3.1, address conflicts that can arise due to miscomprehension of other developers' environment, tasks, and educational background. Based on the literature review and an in-depth understanding of the hardware and software environment during product development the following hypotheses were proposed:

Hypothesis 1: The number of levels to common report (between hardware and software developers) has a negative effect on project success.

The number of levels to common report within a system development group will vary depending on the size and complexity of the project. The diagrams below are three examples of what different system development groups may look like.

Figure 2: Project Structure -- Sample A

In diagram A there is one level between hardware and software developers.

Figure 3: Project Structure -- Sample B

In diagram B there are two levels between hardware and software developers.

Figure 4: Project Structure -- Sample C

Diagram C is an example of a system development group where there is more than one hardware and software team working together on a project. Because the

model was tested by comparing projects, not individuals, in cases such as these the number of management levels within such a group were counted as the maximum number of levels to common report between hardware and software developers.

Hypothesis 2: The percentage of developers within a system development group that have both software and hardware design experience is positively related to project success.

Understanding the development process, problem issues and tradeoffs from both a hardware and software perspective is easier when developers have both hardware and software design experience.

Hypothesis 3: Physical distance is negatively related to project success.

Physical distance was measured by the distance between the hardware and software development teams during product development.

Hardware and software developers often work on projects where the hardware and software components are developed by different suborganizations in different locations, sometimes on two continents, which can make it difficult to maintain a common focus during the project and obtain an understanding of the other's working environment.

Hypothesis 4: Group-based evaluations are positively related to project success.

Group-based evaluation was defined as the degree to which the project manager evaluates the hardware and software developers as one combined team versus evaluating the teams separately or evaluating individuals.

Hypothesis 5: The shared use of hardware and software design documents is positively related to project success.

Hypothesis 6: The shared use of hardware and software design tools is positively related to project success.

Hardware and software design documents and design tools are commonly, if not always, used during the development of complex electronic systems. They aid hardware and software developers to coordinate and adjust their components. Hardware and software design documents and tools can be thought of as alternative means of coordinating the interdependent tasks of hardware and software development.

Common design documents ensure that both hardware and software developers are working effectively to the same ends. Design simulation tools allow software developers to design and test software against the proposed hardware. If software developers find problems regarding the proposed hardware configuration it can be communicated to the hardware developers before a complete prototype is built, thus saving time and money. These tools also allow the hardware developers to design hardware components and test them for accuracy before building prototypes.

Hypothesis 7: All things being equal, it is expected that task difficulty moderates project success.

It is important to take into account the difficulty of a project when considering the realization of project goals. The design tasks in some projects may be more difficult to manage than design tasks in other projects. For example, projects may attempt to cut cost or design cycle or improve product functionality by certain percentages. The degree (level of difficulty) to which the project intends to cut costs, reduce design cycle and/or improve product functionality, can have an effect on the developers' tasks and hence respondents' assessments of project success. Therefore, the difficulty of the tasks hardware and software developers face in order to fulfill cost, design cycle and product functionality goals is of interest.

For example, to reach product size and cost targets, the hardware components of a system may increase the number of application-specific integrated circuits (ASICs) in its layout. ASIC design is a much more complex design task than the simpler process of laying out printed circuit boards. The nature of the function being designed as well as customer requirements may also increase the level of difficulty of a project. The technology being used and the market the product is intended for also have an effect on task difficulty. In some situations a project may use new technology (i.e. design methods or tools) or design a product for a new or constantly changing market. Some projects use current or existing technology during new product development.

Dependent Variables:

The dependent variables were project cost, design cycle and product functionality.

Project Cost

Early defect detention in the design cycle can help reduce a project's cost. Project costs mostly come from design hours which include programming, extensive hours of testing and re-work due to bugs and defects found in the system as well as the development of prototypes -- one of the most expensive activities that can take place during the development process.

Better hardware-software coordination can lead to early detection of defects in a system. This can prevent larger problems from occurring later on in the design cycle, for example, at integration time, thus reducing the hours needed to complete the project. Design errors regarding the functionality of the system not detected early on will cause a delay because the system will not implement the correct function. This reduction can be in the form of fewer design hours needed to complete the project and/or fewer prototypes needed in order to test for accuracy.

Product Design Cycle Goals:

Increased coordination efforts between hardware and software development teams during product development can lead to fewer mismatches between the hardware and software components. This in turn will reduce rework -- the number of design cycles components need to go through -- and will allow developers to finish the project on time.

Product Functionality Goals:

The use of coordination mechanisms during product development can lead to a better match between hardware and software components at the integration stage. This will assure the developers that the functionality conforms to the design specifications. Before product design begins both hardware and software developers can often make a legitimate case for having almost any function in hardware or software. It is crucial that these issues are discussed and thought out between the hardware and software teams so that functionality is developed to conform to the design specifications. Product functionality refers to the how well the product conforms to the final set of design specifications.

A list of all the model variables can be found in Appendix B.

4. The Questionnaire Survey

4.1 The Questionnaire

Three types of questionnaires, the first designed for the project manager and the latter two designed for the hardware and software developers, were sent by fax or surface mail to each representative from participating organizations.

The three questionnaires were identical in structure -- all contained questions on the independent and dependent variables in a similar format. Some of the questions directed towards the project manager were not answerable by the hardware and software developers and therefore were either omitted from their questionnaire or asked in a different way. The first section of each questionnaire asked questions regarding general background information. The next four parts of the questionnaire requested information on organizational-level factors. The sixth section of the questionnaire asked questions on project specific factors. The seventh section of the questionnaire addressed the moderating variable - task difficulty.

The survey consisted of four types of questions: multiple response, seven-point Likert scales, binary response and short verbal answers.

Before the questionnaire was administered it was pre-tested by the case study participants: a project manager, a hardware developer and a software developer. Case study participants were asked to complete the questionnaires and point out any portion that was confusing. This was to make sure all the questions were understandable and meaningful for their purposes.

Measures:

General Background Information

In this section, background information was gathered on respondents, their organizations and their projects. Most of the questions in this section were multiple choice with one seven-point Likert question.

Measures of Independent Variables

Number of levels to common report was defined as the number of organizational levels between hardware and software project team members and their first common supervisor. This variable was tested using ratio and ordinal scale variables. This section of the survey:

- captured a picture of what the system development group looked like including the number of hardware and software development teams and the managers involved,
- measured the strength of the relationship this common manager had with the design process at the functional level, and
- measured the autonomy given to the hardware/software developers during the development process.

Developers were asked whether they had only hardware or software design experience, or both. This variable was tested using a ratio scale variable. In this section of the survey the following statements were used to measure this variable:

- how many developers from each project had hardware and software design experience, and
- how useful it was to have both hardware and software design experience during the projects specified for this survey

Physical distance was measured by the distance between the hardware and software development teams during product development. This variable was

measured using ratio, ordinal and nominal scale variables. In this section of the questionnaire the following was measured:

- the frequency of informal interaction -- the more the informal interaction, the closer the two teams were situated to each other, and
- the physical distance between the hardware and software teams -- the location of all hardware and software teams was of interest here.

Group-based evaluation was defined as the degree to which the project manager evaluated the hardware and software developers as one team combined versus evaluating the teams separately or evaluating individuals. This variable was tested using both nominal and ordinal scales. The perceived importance of these meetings during the project as well as the degree to which they promoted a common focus between the two development teams was measured here.

Design documents and design tools were measured by asking the respondents what type of hardware and/or software design documents and design tools were used during the design process. These variables were measured using ordinal and nominal variable scales. Some examples were: detailed specifications, full simulations, software emulators, hardware emulators, hardware prototypes and microprocessor boards. Only the hardware and software developers were asked to respond to this section of the questionnaire

Moderating Variable: Task difficulty

Task difficulty was measured by asking respondents how easily they felt it was to meet the three project goals of cost, design cycle and product functionality. The scales ranged from 1 = "strongly disagree" to 7 = "strongly agree". A matrix was used to assess the level of task difficulty in terms of the technology being used as

well as the market the product was directed towards. A project using new technology for a 'first of a kind' product was thought to be more difficult than a project using existing technology for an existing market.

Measure of Dependent Variables

Cost, design cycle, and product functionality goal success was measured by asking respondents to answer on Likert-scale questions ranging from "strongly disagree" to "strongly agree". The questions used were:

- "In the end, the cost goals were easily met."
- "In the end, all of the design cycle schedules were easily met."
- "In the end, the project's functionality goals were easily met."

In addition to these three statement questions, five additional questions were asked regarding the amount of functionality from hardware implemented into software and vice versa, as well as how much functionality had to be removed from or added to the product.

A copy of the three surveys in their entirety can be found in Appendix C.

4.2 The Survey Sample

Sample Selection Process:

Participants were contacted prior to the administration of the survey and had the choice of responding by mail, electronic mail or facsimile.

The sample frame was designed to include: projects from organizations who design (and produce) complex electronics systems in the computer, telecommunications and aviation equipment industries.

A convenience sampling frame was created. The list of projects came from contacts made during interviews with industry experts, which led to contacts within telecommunication, computer and aviation companies. Because of the limited size of this sampling frame, the entire frame was surveyed.

A letter which stated the benefits to the organization of participating in the survey as well as the purpose of the study was sent to a representative at each of the organizations from the sampling frame. The representative was then asked:

- if his or her organization was willing to participate,
- to confirm that they had an appropriate project that had been finished within the last year and,
- that one hardware and software developer and at least one project manager who all worked on the specified project were willing to respond to the questionnaire.

All respondents were promised a copy of the results from the study as an incentive for their participation.

The initial company contact had control of who would participate from their organization. Each company contact was asked that only respondents who conformed to the following titles or roles within their organization were used: project managers, and hardware or software developers. Attempts were made to obtain a project manager and at least one hardware and one software developer as questionnaire respondents for each project.

The project must have been completed within a year prior to participation so that respondents still had a 'fresh' memory of the events which took place during the

design process. Requiring that both hardware and software development teams had worked simultaneously allowed the variables of interest to be tested.

Participating Companies:

Projects gathered were from companies who designed and developed complex electronic systems; most were in the aviation, telecommunication or computer industries.

Table 2: Descriptive Information on Companies

Company	Location	Size (employees at site)	Industry/Products
Fisher & Paykel	Auckland, New Zealand	2000	Firmware
Hewlett Packard	Vancouver, BC	150	Computer
Boeing	Seattle, WA	35 - 400	Aviation
Mitel	Kanata, ON	1500	Telecommunication systems
MDA	Vancouver, BC	700	Systems Engineering (digital electronic systems)
Lumonics	Kanata, ON	150	Laser Systems
Argus Control Systems	White Rock, BC	9	Agricultural Process Control and data acquisition Systems
Glenayre	Vancouver, BC	650	Paging and Mobile data equipment
Litton Guidance and Control Systems	Woodland Hills, CA	1500	Global Positioning Systems
Northern Airborne	Kelowna, BC	60	Aviation
Spar Aerospace	Kanata, ON	320	Aviation
Racal Canada	Vancouver, BC	40	High Frequency Radio
Dynapro	New Westminster, BC	350	Industrial Interfaces
Sierra Wireless	Richmond, BC	38	Wireless Telecommunications
Allied Signal	Redmond, WA	700	Aviation

The above table illustrates the variety of companies who participated in the survey. Most companies were from British Columbia, a few from Ontario and

Washington State and one from New Zealand. The sizes of each company site ranged from 9 to 2000 employees.

Survey Response:

In total 31 sets of questionnaires were distributed. A total of 18 projects from 15 companies was gathered. This made for a total of 55 individual cases and a response rate of 58%. All of the received questionnaires were from companies headquartered in Canada and the United States except one which was based in New Zealand.

4.3 Survey Data Set Description

Descriptive Data for Projects

The data set included responses from project managers, hardware and software developers from 18 projects. Descriptive data for each project can be seen in the table below:

Table 3: Descriptive Data for Projects

No	No. of project manager respondents	No. of HW Developer Respondents	No. of SW Developer Respondents	Total Number of Respondents	Cost (\$million)	Length in months	# of people in project
1	1	1	1	3	2.5	30	15
2	1	1	1	3	1.5	18.67	6
3	1	1	1	3	100	72	100
4	1	1	1	3	Unknown	25.33	25
5	1	1	1	3	Unknown	30	15
6	1	1	1	3	1.8	12	12
7	1	1		2	2.5	19	20
8	1			1	1.0	240	30
9	1	1	1	3	1.5	26	10
10	1	1	1	3	.023	13	3
11	1	2	1	4	0.75	22.50	25
12	1	1	1	3	1	14.50	23
13	1	2	1	4	0.3	30	12
14	1	1	1	3	02.5	18	30
15	1	1	1	3	.35	24	8
16	1	1	2	4	1.5	14.75	60
17	1	2	1	4	2.5	16	14
18	1	1	1	3	1.2	32	20

Table 3 illustrates that most companies were able to supply three participants per project. In one case there was only one respondent, a project manager, and in some cases companies were able to supply two hardware or software developers.

Coding the Data Set:

Once the data was received it was coded into the SPSS statistical software package. There were three main sections to the questionnaire – a demographics section, a section on the independent variables and a section on the dependent variables.

Most of the questions in the demographic section were short verbal answer or multiple choice questions. In this section the verbal responses were coded as they were written by the participant.

In the following two sections – sections on the independent and dependent variables – most of the questions were seven-point Likert scales. These questions were coded in SPSS from one to seven. In some cases there were short answer type questions and these were coded in a similar way as those in the demographic section. In one question, the participant was asked to draw a diagram of his or her system development structure.

Variable Label Prefixes:

Variables were labeled by respondent type. In some cases all three respondent types answered the same question and in other cases only the project manager or developers responded to questions. The table below describes how the survey variables were coded:

Table 4: Variable Label Prefixes

Questions asked of:	Variable Label Prefix:
All three respondents	q
Project Managers only	pq
Hardware and Software developers only	hs
Hardware developers only	hw
Software developers only	sw

Variables Used in Data Analysis:

The following variables were used in subsequent data analysis.

Table 5: Respondent Demographic Variables

Variables:	Variable Label:
job title	q1
level of education	q2
age	q3

Table 6: Organization Demographic Variables

Variables:	Variable Labels:
people in organization	pq1
industry	q4

Table 7: Project Demographic Variables

Variables:	Variable Labels:
length of project in months	q5
people involved on project	pq2
target market	q6
initial project goals	q7
cost of project	pq3

Table 8: Independent Variables from Questionnaire

Independent Variables:	Questionnaire Variable:
number of levels to common report	q8levels
hardware and software design experience	pq7
physical distance	hs8
group-based evaluations	pq17
design documents and tools	hs17, hs20b

Table 9: Dependent Variables from Questionnaire

Dependent Variables:	Variable Labels:
cost goal	q14
design cycle goal	q16
functionality goal	q18

Exploratory Data Analysis:

Prior to analysis and model testing the data set was examined for accuracy of data entry, missing values and outliers. This was accomplished by analyzing SPSS frequencies, histograms and scatter plots.

Data screening prior to data analysis was important for two reasons. First, it was necessary to look for unusual respondents and or responses in the data set. It was important to have an understanding of these outliers prior to data analysis so that, after running various statistical tests, these cases could be examined for their effects on the tests. Second, since correlations tests were used in the data analysis section, it was important to examine the normality of each variable. Since the variables were not normal Spearman's correlations tests were used.

Outliers and distribution normality was assessed by graphical examination of the data. The survey data set was too small to accurately run such normality tests as the Komolgorov-Smirnov test. Therefore, data screening was accomplished by:

1. Frequencies:

Table 10 shows frequencies for the six independent variables (number of levels to common report (q8levels), hardware/software design experience (pq7), distance between hardware and software teams (hs8), method of evaluating developers (pq17), design document accessibility (hs17) and design tool accessibility (hs20b)).

Table 10: Frequencies for Independent Variables

Frequencies	Number of Levels to Common Report (q8levels)	Hardware and Software Design Experience (pq7)	Distance between Hardware and Software (hs8)	Method of Evaluating Developers (pq17)	Extent of Shared Design Documents (hs17)	Extent of Shared Design Tools (hs20b)
Valid Cases	55	18	35	17	36	36
Mean	1.6	2.8	1.4	1.7	4.0	1.0
Median	1.0	3.0	1.0	1.0	1.6	1.2
Standard Deviation	.8	2.4	1.1	1.1	1.6	1.2
Min/Max	1 - 4	0 - 9	1 - 5	1 - 4	1 - 7	0 - 4

Different variables had different numbers of valid cases because not all types of respondents (project managers, hardware and software developers) were asked the same questions. See table 4.

Table 11 shows frequencies for the three dependent variables (ability to meet cost goals (q14), ability to meet design cycle goals (q16), and ability to meet functionality goals (q18).)

Table 11: Frequencies for Dependent Variables

Frequencies	Cost Goal Success (q14)	Design Cycle Success (q16)	Functionality Goal Success (q18)
Valid Cases	54	55	55
Mean	3.5	2.8	4.4
Median	3	2	5
Standard Deviation	1.6	1.5	1.7
Min/Max	1 - 6	1 - 6	1 - 7

Table 11 shows that respondents tended to find that it was easier to meet functionality goals than cost or design cycle goals.

2. Histograms with Bell Curves:

The two sets of variables in tables 12 and 13 were examined by producing histograms with normal curves. It was important to see how the variables followed the normal curve. Outliers, variables which were far away from the majority, were identified and analyzed further with scatter plots.

Table 12: Histograms on Independent Variables

Independent Variables:	Questionnaire Variable:	Histogram Results:
number of levels to common report	q8levels	somewhat normally distributed
hardware and software design experience	pq7	not normally distributed
physical distance	hs8	not normally distributed
group-based evaluations	pq17	not normally distributed
design documents	hs17	not normally distributed
design tools	hs20b	somewhat normally distributed

Table 13: Histograms for Dependent Variables

Dependent Variables:	Variable Labels:	Histogram Results:
cost goal	q14	not normally distributed
design cycle goal	q16	somewhat normally distributed
functionality goal	q18	not normally distributed

Due to the non-normality of distribution in the dependent variables, Spearman correlations tests rather than Pearson correlation tests were run in the data analysis section.

The demographic variables listed in Table 14 were examined by producing histograms with normal curves. Again, it was important to examine how these variables followed the normal curve. Outliers were identified and analyzed further by running scatter plots.

Table 14: Histograms for Organization Demographic Variables

Variables:	Variable Labels:	Histogram Results
people in organization	pq1	normally distributed
industry	q4	somewhat normally distributed – positive kurtosis

Table 15: Histograms for Project Demographic Variables

Variables:	Variable Labels:	Histogram Results
length of project in months	q5	somewhat normally distributed – positive kurtosis
people involved on project	pq2	bimodal

Summary of Exploratory Data Analysis:

After scanning the data set by analyzing SPSS frequencies, histograms and scatter plots, the following variable distributions contained cases which were identified as possible outliers. For each variable outlier cases were noted.

1. Length of project

This variable was examined more closely because of the range in responses. The shortest project took 8 months to complete and the longest project took 240 months, making the average 28 months.

2. Number of managers

For similar reasons as the 'length of project' variable, 'number of managers' per project was examined more closely. This variable ranged from one manager per project to nine managers per project where the average was three. After examining a histogram of this variable it was clear that the 'number of managers' was skewed

towards fewer numbers. When 'number of managers' was examined in a scatter plot with 'length of project' it was concluded that a project which had nine managers was an outlier.

3. Total number of teams

The 'total number of teams' variable had a very similar histogram output – heavily weighted towards fewer number of teams. However, there was one outlier – a response of 19 teams where the average was six.

4. Number of people in the organization

It became clear that the participants (project managers) from the 18 projects had different perceptions of the question regarding 'number people in the organization'. When analyzing responses it became clear that different respondents answered this question differently (e.g. one respondent said that the Boeing organization employs only 400 people).

5. Number of people involved in the project

Most of the projects had 30 or fewer people involved. Two projects had more team members – one involved 60 people and one involved 100.

After analyzing a scatter plot with 'number of people involved on the project' with 'length of project' the projects with 60 and 100 people involved were identified as outliers.

6. Cost of project

Four of the 18 projects responded to this question by stating that the customer funded the project or part of the project and that the total cost was not known. This meant that for 22% of the cases the total cost per project was unknown.

7. Developers with both hardware and software experience and hardware developers with both experience

Question pq7 asked project managers for the total number of developers with both hardware and software design experience, while pq8 asked for the number of hardware developers with both hardware and software design experience. Logically the value of pq7 should be greater than or equal to pq8 for each case. For one case, however, this was not true as is illustrated in Figure 1.

Figure 5: pq7 versus pq8 Scatter Plot

The above exploratory data analysis shows that there were two extreme outliers: one project with a duration of 20 years and a second with a cost of \$100M. These two cases were flagged for possible exclusion at various steps at the model test stage.

5. Survey Data Analysis

5.1 Similarities and Differences Among Survey Participants

Inter-group comparisons among survey respondents were made by comparing group means and correlation tests.

Part One: Comparing Means

Variables labeled Type 1, 2 and 3 were created in order to compare the three respondent types:

Type 1 = project manager
Type 2 = hardware developers
Type 3 = software developer

Group means between respondent types and hypothesized variables were compared using the Kruskal-Wallis One-Way Analysis of Variance with significance at the .10 level. The Kruskal-Wallis test was used instead of the F-Test because: 1) the data was not normally distributed and 2) the data set did not consist of interval data.

In Table 16 the project manager, hardware and software developers' responses were analyzed by comparing means with the three success measures (q14 = ability to meet cost goals, q16 = ability to meet design cycle goals and q18 = ability to meet functionality goals):

Table 16: Comparison of Means: Respondent Type and Success Variables

Ranks			
	TYPE	N	Mean Rank
Q14	1.00	18	22.72
	2.00	20	33.72
	3.00	16	25.09
	Total	54	
Q16	1.00	18	25.81
	2.00	20	27.17
	3.00	17	31.29
	Total	55	
Q18	1.00	18	26.14
	2.00	20	30.17
	3.00	17	27.41
	Total	55	

Test Statistics ^{a,b}			
	Q14	Q16	Q18
Chi-Square	5.417	1.177	.662
df	2	2	2
Asymp. Sig.	.067	.555	.718

^a. Kruskal Wallis Test
^b. Grouping Variable: TYPE

The K-W test found a significant difference between the three groups and their perceived levels of project cost goal success. Hardware developers were more likely to think it was successful than either project managers or software developers.

Table 17 below compares means with the project manager, hardware and software developers with other independent variables:

- q8levels = number of levels to common report
- hs8 = distance between hardware and software teams
- hs17 = number of design documents accessible to both hardware and software developers,
- hs20b number of design tools accessible to both hardware and software developers,
- hw3 and sw3 -- developer who has both hardware and software design experience,
- q7 = ambiguity of initial goals,

- q9 = frequency of group meetings,
- q26 = (enjoyment of work) variables.

Table 17: Comparison of Means

Respondent Type and Independent Variables

Ranks			
	TYPE	N	Mean Rank
Q8LEVELS	1.00	18	28.56
	2.00	20	27.77
	3.00	17	27.68
	Total	55	
HW3	2.00	20	10.50
	Total	20 ^a	
SW3	3.00	17	9.00
	Total	17 ^a	
HS8	2.00	18	16.89
	3.00	17	19.18
	Total	35	
Q7	1.00	18	27.78
	2.00	20	27.63
	3.00	17	28.68
	Total	55	
Q9	1.00	18	23.75
	2.00	20	28.83
	3.00	17	31.53
	Total	55	
HS17	2.00	19	19.03
	3.00	17	17.91
	Total	36	
HS20B	2.00	20	17.38
	3.00	16	19.91
	Total	36	

a. There is only one non-empty group.
Kruskal-Wallis Test cannot be performed.

Test Statistics ^{a,b}

	Q8LEVELS	HS8	Q7	Q9	HS17	HS20B
Chi-Square	.041	1.430	.047	2.308	.105	.599
df	2	1	2	2	1	1
Asymp. Sig.	.980	.232	.977	.315	.746	.439

a. Kruskal-Wallis Test

b. Grouping Variable: TYPE

There was no significant differences between the three groups of respondents and their perceived measures of the independent variables.

Part Two: Correlations

Correlations between variables were run in order to see which variables were correlated with project success among the three different respondent types.

Spearman's tests was used because of the non-normality of the data.

Table 18: Correlations between independent and dependent variables, by respondent type

Correlations						
TYPE				Q14	Q16	Q18
1.00	Spearman's rho	Correlation Coefficient	Q8LEVELS	-.530	-.177	-.174
		Sig. (2-tailed)	Q8LEVELS	.024	.483	.489
		N	Q8LEVELS	18	18	18
2.00	Spearman's rho	Correlation Coefficient	Q8LEVELS	-.246	.045	.244
		Sig. (2-tailed)	Q8LEVELS	.295	.850	.301
		N	Q8LEVELS	20	20	20
3.00	Spearman's rho	Correlation Coefficient	Q8LEVELS	-.082	-.156	-.270
		Sig. (2-tailed)	Q8LEVELS	.763	.550	.294
		N	Q8LEVELS	16	17	17

Correlations						
TYPE				Q14	Q16	Q18
1.00	Spearman's rho	Correlation Coefficient	PQ7	-.321	-.045	-.351
		Sig. (2-tailed)	PQ7	.195	.859	.153
		N	PQ7	18	18	18
2.00	Spearman's rho	Correlation Coefficient	PQ7	.	.	.
		Sig. (2-tailed)	PQ7	.	.	.
		N	PQ7	0	0	0
3.00	Spearman's rho	Correlation Coefficient	PQ7	.	.	.
		Sig. (2-tailed)	PQ7	.	.	.
		N	PQ7	0	0	0

Correlations

TYPE			Q14	Q16	Q18
1.00	Spearman's rho	Correlation Coefficient	HS8	.	.
		Sig. (2-tailed)	HS8	.	.
		N	HS8	0	0
2.00	Spearman's rho	Correlation Coefficient	HS8	-.415	-.121
		Sig. (2-tailed)	HS8	.087	.631
		N	HS8	18	18
3.00	Spearman's rho	Correlation Coefficient	HS8	-.426	-.214
		Sig. (2-tailed)	HS8	.100	.410
		N	HS8	16	17

Correlations

TYPE			Q14	Q16	Q18
1.00	Spearman's rho	Correlation Coefficient	PQ17	-.250	-.049
		Sig. (2-tailed)	PQ17	.333	.853
		N	PQ17	17	17
2.00	Spearman's rho	Correlation Coefficient	PQ17	.	.
		Sig. (2-tailed)	PQ17	.	.
		N	PQ17	0	0
3.00	Spearman's rho	Correlation Coefficient	PQ17	.	.
		Sig. (2-tailed)	PQ17	.	.
		N	PQ17	0	0

Correlations

TYPE			Q14	Q16	Q18
1.00	Spearman's rho	Correlation Coefficient	HS17	.	.
		Sig. (2-tailed)	HS17	.	.
		N	HS17	0	0
2.00	Spearman's rho	Correlation Coefficient	HS17	.317	.417
		Sig. (2-tailed)	HS17	.186	.076
		N	HS17	19	19
3.00	Spearman's rho	Correlation Coefficient	HS17	.104	-.223
		Sig. (2-tailed)	HS17	.700	.390
		N	HS17	16	17

Correlations

TYPE			Q14	Q16	Q18
1.00	Spearman's rho	Correlation Coefficient	HS20B	.	.
		Sig. (2-tailed)	HS20B	.	.
		N	HS20B	0	0
2.00	Spearman's rho	Correlation Coefficient	HS20B	.327	-.294
		Sig. (2-tailed)	HS20B	.160	.208
		N	HS20B	20	20
3.00	Spearman's rho	Correlation Coefficient	HS20B	-.225	-.412
		Sig. (2-tailed)	HS20B	.420	.113
		N	HS20B	15	16

The above correlations tests illustrate that, for project managers, 'number of levels to common report' is significantly negatively correlated with 'ability to meet cost goals'. The tests also illustrate that, from both the hardware and software developers' perspectives, the distance between hardware and software teams is negatively correlated to 'ability to meet cost goals'. Finally, for hardware developers, the extent of shared documentation is positively correlated with 'ability to meet design cycle schedules'. All of these findings are congruent with the study's initial coordination mechanism hypotheses.

5.2 Individual Respondent Data Set Correlations

During the course of data analysis a number of interesting correlations were found that touch on the theoretical model. While these findings do not imply causality they are nonetheless of interest. In all of the following correlations where q5 (length of project) was used, one response of 240 months was removed.

Cost Goals

The following variables had significant correlations with 'ability to meet cost goals' (q14):

Table 19: Correlations between Independent Variables Dependent Variable Cost Goal Success

Independent Variables	Dependent Variable Cost Goal Success (q14)
q5 (length of project)	-.228 (.100)
q8a (number of managers)	-.300 (.033)
q10 (frequency hardware and software developers agree on goals)	.251 (.073)
q8e (total number of teams)	-.351 (.011)
hs8 (distance between hardware and software teams)	-.448 (.008)

From Table 19 we see that project size is correlated with meeting project cost goals. There were significant negative correlations between length, number of managers and teams, all of which have to do with project size. These correlations imply that when a project was lengthy or involved several managers or teams, the less likely it was to meet project cost goals. The table also shows that the greater the distance between hardware and software teams, the less likely it was to meet project cost goals. As well, when hardware and software developers were able to agree on goals more often, the more likely it was to meet project cost goals.

Design Cycle Goals

The following variables had significant correlations with 'ability to meet design cycle goals':

Table 20: Correlations between Independent Variables and Dependent Variable Design Cycle Goal Success

Independent Variable	Dependent Variable Design Cycle Goal Success (q16)
q8b (number of hardware teams)	-.309 (.029)
q14 (ability to meet cost goals)	.342 (.011)
hs15 (hardware/software developers see issues from software/hardware perspective)	.448 (.006)

Table 20 again shows the influence of project size on project success. There was also a significant correlation, as one would expect, between 'ability to meet cost goals' and 'ability to meet design cycle goals'. Table 21 also shows that design cycle schedules were more likely to be met when hardware and software developers were able to understand issues from the other's perspective.

Functionality Goals

The following variables had significant correlations with 'ability to meet product functionality goals':

Table 21: Correlations between Independent Variables and Dependent Variable Functionality Goal Success

Independent Variables	Dependent Variable Functionality Goal Success (q18)
q5 (length of project)	-.251 (.067)
q9 (frequency of group meetings)	.277 (.041)
q20 (functionality from hardware moved to sw)	-.398 (.003)
q24 (functionality removed from product)	-.214 (.117)
q25 (functionality added to product)	-.310 (.023)
q26 (enjoyment of work)	.246 (.072)
pq14 (frequency providing support)	.575 (.013)
pq15 (hardware see issues from software perspective)	.454 (.058)
pq16 (software sees issues from hardware perspective)	.585 (.001)
hs15 (hardware/software developers see issues from software/hardware developers perspective)	-.360 (.031)
hs8 (distance between hardware and software teams)	-.261 (.130)

Here it is apparent that when functionality had to be added, removed or moved from hardware to software or vice versa, project functionality goals were less likely to be met. The table also shows that the longer the project and the greater the distance between hardware and software teams, the less likely the project met functionality goals. However, it seems that frequency of group meetings does not facilitate 'ability to meet functionality goals', since this variable was measured using an inverted scale.

Management Support

There were two separate questions which addressed management support during product design. When examining at the question asked of hardware and software developers, "how often was the project manager willing to provide support", three variables were significantly correlated with it:

Table 22: Correlations between Independent Variables and Project Management Support from the Developers' Perspective

Independent Variables	Dependent Variable Project Manager Support (hs9)
q5 (length of project)	-.243 (.153)
q10 (frequency hardware and software developers agree on goals)	.436 (.009)
hs15 (hardware/software developers see issues from software/hardware developers perspective)	.363 (.032)

The following conclusions were identified:

- the longer the project, the less likely it was that management provided support.

Management support is associated with a common understanding between hardware and software teams:

- management support was correlated with developer agreement on project goals
- management support was associated with mutual understanding by developers of project team goals.

Table 23: Correlations between Independent Variables and Project Manager Support from the Managers' Perspective

Independent Variables	Dependent Variable Project Manager Support (pq14)
q18 (ability to meet functionality goals)	.575 (.013)
q26 (enjoyment of work)	.527 (.025)
pq16 (software developer sees issues from hardware developer perspective)	.553 (.017)

The variable pq14, which asked only project managers “how often were you willing to provide support (i.e. time, money)” was positively correlated with:

- the ability to meet functionality goals,
- the project manager’s enjoyment of his roll in the project and,
- the degree to which software developers saw issues from the hardware developer’s perspective.

Communication

The following table shows various 'communication' variables and their significant correlations with other independent variables:

Table 24: Correlations between Communication Variables and other Independent Variables

Communication Variables	Independent Variables	Significance
hs13 (teleconference)	q5 (length of project)	.343 (.068)
hs13 (teleconference)	hs8 (distance between hardware and software teams)	.366 (.060)
hs13 (teleconference)	q26 (enjoyment of work)	-.329 (.081)
hs13 (teleconference)	c1 (number of locations)	.516 (.004)
hs12 (phone)	hs8 (distance between hardware and software teams)	.419 (.024)
hs14 (face-to-face)	hs8 (distance between hardware and software teams)	-.322 (.059)
hs14 (face-to-face)	hs15 (hardware and software developers see issues from software and hardware developers perspective)	.545 (.001)

The above table illustrates that when hardware and software developers used teleconferencing during product design, projects were more likely to be longer, and have several distant locations. At the same time developers were less likely to enjoy their work when teleconferencing was used during product design.

When developers used face-to-face communication they were more likely to see the products' overall goals from each others' perspective.

Enjoyment of Work

The following variables had a significant correlation with 'enjoyment of work':

Table 25: Correlations between Independent Variables and Dependent Variable Enjoyment of Work

Independent Variables	Dependent Variable Enjoyment of Work (q26)
q18 (ability to meet functionality goals)	.245 (.072)
q20 (functionality from hardware moved to sw)	-.361 (.007)
pq14 (management support)	.527 (.025)
hs8 (distance between hardware and software teams)	-.403 (.016)
c1 (number of different locations)	-.255 (.060)
hs13 (communication – teleconference)	-.329 (.081)

When functionality goals were met, the more likely it was that developers enjoyed their work on the project. When product functionality did not meet its goals, for example when functionality needed to moved from hardware to software, it was less likely that developers enjoyed their work on a project.

Management support also seemed to be positively correlated with developers enjoying their work. Distance between hardware and software developers, number of locations of teams and number of tools new to developers were negatively correlated to developers enjoying their work.

5.3 Creating the Project Data Set

The main objective of this study was to test the effectiveness of certain coordination mechanisms against project success. In order to test this theoretical model it was necessary to create a project data set – a data set which contained 18 cases, one for each project. Again, there were three sections to the project data set: demographic, independent and dependent variables, which were first analyzed using exploratory data analysis.

The variables in these sections were coded by either taking the only the project manager's response or the average of the hardware and software developer, whichever was appropriate. The three sections included the following variables:

Background Data:

Table 26: Organizational Demographic Variables for Project Data Set

Variables	Variable Labels
people in organization	pq1
industry	q4 (project manager response)

Table 27: Project Demographic Variables for Project Data Set

Variables:	Variable Labels:
length of project in months	q5 (project manager response)
people involved on project	pq2
cost of project	pq3

Independent Variables:

Table 28: Independent Variables for Project Data Set

Independent Variables:	Questionnaire Variable:
number of levels to common report	q8levels (project manager response)
hardware and Software Design Experience	pq7 (project manager response)
physical distance	hs8 (average of hardware and software developer response)
group-based evaluations	pq17 (project manager response)
design documents and tools	hs17, hs20b (average of hardware and software developer response)

Dependent Variables:**Table 29: Dependent Variables for Project Data Set**

Dependent Variables:	Variable Labels:
cost goal	q14 (averaged across all respondents per project)
design cycle goal	q16 (averaged across all respondents per project)
functionality goal	q18 (averaged across all respondents per project)
SUCCPM	Type 1 (project manager) this variable is the average of: q14, q16, q18
SUCCHS	Type 2, Type 3 this variable is the average of: q14, q16, q18

Exploratory Data Analysis: Project Data Set of 18 cases

Again, exploring data variables prior to data analysis was important for two reasons. First, it was necessary to look for unusual projects or responses in the data set so that, after running various statistical tests, these 'odd responses' could be examined for their effects. Second, since correlations tests were used in the data analysis section, it was important to examine the normality of each variable. The type of correlation tests used depends on the normality of the variables.

Graphical exploratory analysis was conducted. The project data set was too small to run such normality tests as the Komolgorov-Smirnov test for normality. Therefore, exploring the data was accomplished in the following four ways:

examining summary statistics, histograms, P-P and Q-Q normality charts and scatter plots.

Summary Statistics:

Tables 32 and 33 show summary statistics for the three dependent variables (ability to meet cost goals (q14), ability to meet design cycle goals (q16), ability to meet functionality goals (q18), and an averaged measure of success for project managers (the average of q14, q16 and q18 from the project managers' perspective called SUCCPM). For developers, the variable SUCCHS was computed as the average of their responses to q14, q16, and q18.

Table 30: Summary Statistics for Dependent Variables from a Project Managers' Perspective Project Data Set

Frequencies	Cost Goal q14	Design Cycle Goal q16	Functionality Goal q18	Project Manager's Perception of Overall Success SUCC-PM
Valid Cases	18	18	18	18
Mean	3.1	2.6	4.2	3.3
Median	2.5	2.0	4.5	3.3
Standard Deviation	1.6	1.5	1.8	1.3
Min/Max	1 - 6	1 - 6	1 - 7	1 - 5.7

Project managers felt that, on average, project functionality goals were more likely to have been achieved than project cost or design cycle goals.

Table 31: Summary Statistics for Dependent Variables from Hardware and Software Developers' Perspective

Frequencies	Cost Goal q14	Design Cycle Goal q16	Functionality Goal q18	Developer's Perception of Overall Success SUCCPM
Valid Cases	17	17	17	17
Mean	3.8	3.1	4.6	3.8
Median	3.7	2.7	5.0	4.1
Standard Deviation	1.2	1.3	1.4	1.0
Min/Max	1.5 - 6	1.3 - 5	1.5 - 6.5	1.5 - 5.5

Hardware and software developers followed a similar pattern with respect to project goals. They were more likely to perceive that project functionality goals had been met.

The following table show summary statistics for the six independent variables (number of levels to common report (q8levels), hardware/software design experience (pq7), distance between hardware and software teams (hs8), method of evaluating developers (pq17), design document accessibility (hs17) and design tool accessibility (hs20b)).

Table 32: Summary Statistics for Independent Variables -- Project Data Set

Frequencies	Number of Levels to Common Report q8levels	Number of Developers with both HW and SW Design Experience pq7	Distance between HW and SW Teams hs8	Group-based Evaluations pq17	Extent of Shared Design Documents hs17	Extent of Shared Design Tools hs20b
Valid Cases	18	18	17	18	17	17
Mean	1.7	2.8	1.6	1.7	4.0	1.4
Median	1.0	3.0	1.0	1.0	3.7	1.5
Standard Deviation	1.0	2.4	1.4	1.1	1.1	1.3
Min/Max	1 - 4	0 - 9	1 - 5	1 - 4	2.3 - 6	0 - 4

Table 35 shows summary statistics for other descriptive variables (length of project in months (q5), cost of project (pq3), and number of people involved per project (pq2)) which were believed to be important with regards to case and project characteristics.

Table 33: Summary Statistics for Other Demographic Variables -- Project Data Set

Frequencies	Length of Project in Months q5	Cost of Project pq3 (\$000,000)	Number of People involved on Project pq2
Valid Cases	17	16	18
Mean	36	7.558	23.7
Median	18	1.5	23
Standard Deviation	53	24.665	23
Min/Max	12 - 240	.023 - 100	3 - 100

Histograms with Normal Curves:

The variables listed below were examined by producing histograms with normal curves, as it was important to see how they followed the normal curve. Outliers, cases which were far away from the majority, were identified and analyzed further by producing scatter plots.

Table 34: Independent Variables examined with Histograms -- Project Data Set

Independent Variables:	
number of levels to common report	q8levels (project manager response)
hardware and software design experience	pq7 (project manager response)
physical distance	hs8 (average of hardware and software developer response)
group-based evaluations	pq17 (project manager response)
design documents and tools	hs17, hs20b (average of hardware and software developer response)

Table 35: Dependent Variables examined with Histograms -- Project Data Set

Dependent Variables:	
cost goal	q14 (averaged across all respondents per project)
design cycle goal	q16 (averaged across all respondents per project)
functionality goal	q18 (averaged across all respondents per project)
SUCCPM	Type 1 (project manager) this variable is the average of: q14, q16, q18
SUCCHS	Type 2, Type 3, q14, q16, q18 this variable is the average of: q14, q16, q18

After analyzing histograms of the above variables from the managers' perspective it was clear that none of them were very normally distributed. This meant that all correlations had to be run using the Spearman Correlation test.

Similarly, after analyzing histograms of the dependent variables from the developers' perspective it was clear that all of them except q14 were not normally distributed. This also meant that correlations with them had to use the Spearman Correlation test.

When analyzing a histogram of pq3 (overall project cost) it was clear that it was not normally distributed.

The following histogram shows pq3 when the respondent who stated project costs were \$100M is removed. In all of the tests in section 5.4 this case was censored.

Histograms on the other two demographic variables (q5 – length of project and pq2 – number of people involved on project) showed similar results – that the variables were not normally distributed. In the case of q5 there was an extreme

outlier with a response of 240 months. The following histogram shows q5 with this case included and is then followed by a histogram with this case removed. In the model test section this case was censored.

5.4 Testing the Model

This section extends the exploratory work performed in section 5.1 to the actual model test. In section 5.1, Spearman correlations were used to examine the

differences between hardware developers, software developers, and project managers. In this section, we test the model by examining differences between different projects. As discussed in section 5.2, some of the variables used in this analysis were derived by computing averages or other summary measures of all the respondents from each project, while in other cases the responses of the project manager were used to describe the project.

The following variables were identified from the questionnaire to represent the independent variables of the theoretical model:

Table 36: Independent Variables used to test the Theoretical Model

Model Variable	Independent Variable	Questionnaire Type
Number of Levels to Common Report	number of levels of management	1
Hardware and Software Design Experience	number of developers in project with both hardware and software design experience	1
Physical Distance	distance from hardware/software teams	2, 3
Group-Based Evaluations	methods of evaluation	1
Extent of Shared Design Documents	frequency of design documents shared by both hardware and software developers	2, 3
Extent of Shared Design Tools	number of design tools shared by both hardware and software developers	2, 3

Average scores of independent variables for “successful” and “unsuccessful” projects indicate which factors were associated with project success.

1. Comparison of Means

Mean scores on independent variables were examined between “successful” and “unsuccessful” projects. “Successful” projects were defined, as in section 5.1, as those with an average score of 4.0 (where it was the midpoint on a seven-point Likert scale) or higher on cost goal, design cycle goal, and product functionality goal success measures.

Table 37: Comparison of Means for Successful and Unsuccessful Projects

Success	Number of Levels to Common Report (q8levels)	Developers with both HW and SW Design Experience (pq7)	Distance between HW and SW Teams (hs8av)	Group-based Evaluations (pq17)	Extent of Shared Design Documents (hs17av)	Extent of Shared Design Tools (hs20bav)
Unsuccessful (.00)	1.92	3.38	1.54	1.46	4.00	1.73
Successful (1.00)	1.20	1.40	1.80	2.20	4.10	.500
total	1.72	2.83	1.62	1.67	4.04	1.26

As Table 37 illustrates, projects that were deemed to be “successful” had, on average, a lower number of management levels to common report, fewer developers with both hardware and software experience, a greater distance between teams, and fewer design tools accessible by both hardware and software developers.

To test these differences for statistical significance Kruskal-Wallis analysis of variance tests were run. Only the difference for q8levels was significant.

Table 38: Correlations -- Independent Variables and Combined Success Measures

Model Variable	SUCCHS	SUCCPM
q8levels	-.394 (.118)	-.330 (.181)
pq7	-.588* (.017)	-.104 (.690)
hs8av	-.271 (.293)	.167 (.523)
pq17	.052 (.842)	-.195 (.438)
hs17av	.160 (.539)	.222 (.392)
hs20bav	-.467* (.059)	-.301 (.240)

*significant at .10

2. Correlations

Table 38 shows Spearman correlations between the model variables and a combined success measure for hardware and software developers (SUCCHS) and project manager (SUCCPM):

Table 38 illustrates that there are two independent variables which correlate with success: pq7 (number of developers with both hardware and software design experience) and hs20bav (extent of shared design tools). Both were negatively correlated with success from hardware and software developers' perspectives. These strong negative correlations do not agree with the original hypotheses.

The table below shows correlations between the model variables and individual success measures for project managers and hardware and software developers:

Table 39: Correlations -- Independent Variables with Independent Success Measures

Model Variable	q14hs	q14pm	q16hs	q16pm	q18hs	q18pm
q8levels	-.392 (.119)	-.530* (.024)	-.315 (.218)	-.177 (.483)	-.127 (.627)	-.174 (.489)
pq7	-.683* (.004)	-.254 (.325)	-.261 (.330)	.100 (.702)	-.447* (.082)	-.250 (.332)
hs8av	-.499* (.042)	.110 (.647)	-.159 (.542)	.211 (.417)	-.274 (.286)	-.177 (.498)
pq17	.206 (.429)	-.300 (.226)	.168 (.519)	-.020 (.937)	-.089 (.733)	.007 (.976)
hs17av	.075 (.774)	.315 (.219)	.041 (.876)	.070 (.788)	.220 (.397)	.231 (.372)
hs20bav	.179 (.493)	-.143 (.583)	-.716* (.001)	-.528* (.029)	-.310 (.226)	-.189 (.468)

*significant at .10

Table 39 shows that pq7 (number of developers on a project with both hardware and software design experience) is negatively correlated with q14hs and q18hs (ability to meet project cost and functionality goals, respectively). This does not agree with the original hypothesis which stated that the number of developers with both hardware and software design experience will facilitate project success.

The above tests were run again after eliminating two outlier cases (i.e, one case where the project duration was 20 years and a second case where the project cost was \$100M) in order to see if there were any further significant correlations between the independent and dependent variables. Results were similar to those in Table 38 and Table 39.

Variables which did not show significant correlations with project success in Table 39 were analyzed further using scattergrams in order to check for non-linear relationships. There did not seem to be any non-linear relationships.

Task Difficulty

Task difficulty was first analyzed by running Spearman correlations with the individual independent task difficulty measures for both the managers and developers.

Table 40: Spearman Correlations – individual task difficulty measures

	q11hs	q11pm	q12hs	q12pm	q13hs	q13pm
q11hs		-.008 (.975)	.580 (.015)	-.101 (.699)	.421 (.093)	-.057 (.829)
q11pm	-.008 (.975)		-.131 (.616)	.403 (.097)	.118 (.650)	.670 (.002)
q12hs	.580 (.015)	-.131 (.616)		-.142 (.587)	.416 (.097)	-.215 (.408)
q12pm	-.101 (.699)	.403 (.097)	-.142 (.587)		-.158 (.546)	.433 (.072)
q13hs	.421 (.093)	.118 (.650)	.416 (.097)	-.158 (.546)		-.093 (.742)
q13pm	-.057 (.829)	.670 (.002)	-.215 (.408)	.433 (.072)	-.093 (.742)	

The Spearman correlations in Table 40 indicate that developers and managers view task difficulty differently. While task difficulty variables are correlated within the developer and project manager groups, this is not the case between them.

In order to make further analysis less cumbersome the above individual task difficulty measures were combined in order to make two measures of overall task difficulty – one measure for project managers and a second measure for developers. These variables were computed by summing each individual task difficulty measure for each project. The new variable calculated for task difficulty from the manager's perspectives was called taskdpm and the new variable calculated for task difficulty from the developers' perspectives was called taskdhs. These two new aggregated

task difficulty measures were tested using Spearman's correlations to see if they were correlated. Since they were not correlated (-.164 at a significance level of .529) these two variables could not be combined.

Next the two summary variables were tested for correlations against project success. Please see table below:

Table 41: Aggregate Task Difficulty Measures with Project Success

	succhs	succpm
taskdhs	.626* (.007)	-.037 (.888)
taskdpm	.140 (.591)	.392 (.108)

Table 41 illustrates that when developers view task difficulty as high, project success is more easily met. When project managers view task difficulty as high project success is less easily met. This is an interesting difference between developers and project managers and their view of task difficulty and ability to meet project goals.

Task difficulty was originally hypothesized to have a moderating effect on the relationship between the independent variables and project success. In order to test this Spearman correlations were run between the independent variables and project success variables while moderating for low and high levels of task difficulty from both the developers' and managers' perspectives.

Table 42: Spearman Correlations with Model Variables and Moderating Variable Developers' Perception of Task Difficulty

	developers' perceptions of low task difficulty		developers' perceptions of high task difficulty	
	succpm	succhs	succpm	succhs
q8levels	-.767* (.016)	-.663* (.052)	.259 (.536)	-.103 (.810)
pq7		-.713* (.047)		-.599 (.116)

*significant at .10

Table 42 illustrates that there is a moderating effect from the variable task difficulty from the developers' perceptions. When task difficulty is low there is a significant correlation between 'number of levels to common report', 'developers with both hardware and software design experience' and project success (succhs). As well, there is a significant correlation between 'number of levels to common report' and project success (succpm). When task difficulty is high from the developers' perspective there are no correlations between the variables mentioned above.

Table 43: Spearman Correlations with Model Variables and Moderating Variable Project Managers' Perception of Task Difficulty

	project managers' perceptions of low task difficulty	project managers' perceptions of high task difficulty
	succhs	succhs
q8levels	-.076 (.872)	-.635* (.049)
pq7	-.533 (.276)	-.674* (.033)

*significant at .10

Table 43 illustrates that there is a moderating effect from the variable task difficulty from the project managers' perception. When task difficulty is high there is a significant correlation between 'number of levels to common report', 'developers with both hardware and software design experience' and project success (succhs). When task difficulty is low from the project managers' perspective there are no correlations between the variables mentioned above.

Hypotheses Test Results:

Number of Levels to Common Report

The independent variable 'number of levels to common report' was hypothesized to have a negative relationship with project success. This variable was measured by asking participants to draw a diagram (question 11 in questionnaire) illustrating the number of hardware and software development teams and the levels of management involved on the project specified for the survey. Interestingly, in several cases, respondents from the same project gave different 'pictures' for this question. In order to accommodate these discrepancies it was decided to take the project manager's response only.

When comparing means for 'number of levels to common report', it was found that successful projects had fewer levels of common report (1.20) than unsuccessful projects (1.92). This finding supports the hypothesis that the more levels to common report the less likely it is to meet project success. However, this attribute was not significantly different at the $p < .1$ level between successful and unsuccessful projects.

When running correlations the independent variable 'number of levels to common report' was only significantly related to 'ability to meet cost goals' from the project manager's perspective. This variable was not significantly related to any other individual or combined success measure.

Hardware and Software Design Experience

This variable was hypothesized to be positively related to project success, and was measured by asking the project manager how many developers who

participated in the project specified for this survey had both hardware and software experience. Individual responses from hardware and software developers were not included in the overall project variable.

This independent variable was also strongly negatively correlated with SUCCHS (a combined success measure for hardware and software developers) and strongly negatively correlated with q14hs (individual measures from the developers' perspectives of project cost goal success). When controlling for project cost (pq3), 'hardware and software design experience' was still significantly negatively correlated to SUCCHS and q14hs and q18hs.

Physical Distance

This variable was hypothesized to be negatively related to project success and was measured by asking hardware and software developers how far apart they were from each other during the design process for the project specified for this survey. The developers were asked if they were either: in the same building, a short car ride, a long car ride, a short flight or a long flight – a five point ordinal scale. The average score per project was taken as the measure for 'physical distance'.

When comparing the means for 'physical distance' there was not a great difference between successful and unsuccessful projects. In fact, unsuccessful projects had a slightly shorter distance (1.54) than successful projects (1.80). This difference was not significant.

'Physical distance' was also negatively correlated to q14hs which means that the closer the teams were to each other the more likely cost goals were met from

developers' point of view. This variable was not significantly correlated with any other combined or individual success measure.

Group-Based Evaluations

The independent variable 'group-based evaluations' was hypothesized to be positively related to project success, and was measured by asking the project manager how he/she evaluated the developers using a four point nominal scale where developers were either evaluated individually, by hardware or software teams alone, or hardware and software teams combined. There was also a fourth option for a combination of the others options.

When comparing means for 'group-based evaluations' for successful projects, 60% made individual evaluations and 40% made a combination of individual and group evaluations. For unsuccessful projects, 69% used individual evaluation methods where 15% used team (hardware or software) evaluation methods and 15% made hardware-software teams combined evaluations. The Fisher's Exact test results for the above differences showed that these outcomes were not significant at the $p < .1$ level.

There were no significant correlations between 'group-based evaluations' and both combined and individual success measures.

Extent of Shared Design Documents

The independent variable 'extent of shared design documents' was hypothesized to be positively related to project success. This variable was measure by asking the hardware and software developers how often they used design documents shared by both hardware and software developers. This variable was

measure by using a seven-point Likert scale where the average score from the hardware and software developers per project was used.

When comparing the means of 'extent of shared design documents' there was very little difference between successful (4.10) and unsuccessful (4.00) projects. As well this variable had no significant correlations with any of the success measures.

Extent of Shared Design Tools

The independent variable 'extent of shared design tools' was hypothesized to be positively related to project success, and was measured by asking both hardware and software developers how many design tools they used were shared by both teams. Respondent answers were averaged over projects.

When comparing means, successful projects, on average, had fewer shared tools. Unsuccessful projects, however, had more shared design tools. This difference proved to be significant and therefore disagrees with the original hypothesis.

'Extent of shared design tools' was also significantly negatively correlated with both SUCCHS – combined success measures from hardware and software developers. As well, this independent variable was significantly negatively correlated with three individual success measures: q16hs and q16pm (ability to meet design cycle goals from the developers' and project managers' perspective).

Task Difficulty

Task difficulty was hypothesized to have a moderating effect on the relationship between the hypothesized coordination mechanisms and project success. It was evident that there were differences between how developers' and

project managers' perceive task difficulty. As well, there seemed to be some evidence of a moderating effect from task difficulty on the relationship between 'number of levels to common report' and 'hardware and software design experience' with project success.

6. Discussion and Conclusions

Discussion of Findings

When comparing both means of independent variables between successful and unsuccessful projects and correlations between independent variables and project success measures, we find that four independent variables have an effect on project success:

- number of levels to common report,
- hardware and software design experience,
- physical distance and
- extent of shared design tools.

Two of the independent variables, 'number of levels to common report' and 'physical distance' affected project success as predicted while the effects of 'hardware and software design experience' and 'extent of shared design tools' contradicted the hypotheses.

First, as predicted, 'number of levels to common report' had a negative effect on cost goal project success. It was originally hypothesized that this variable would be negatively related to the all measures of project success but this was not the case. A similar study on coordinating design and manufacturing found a similar result where number of levels to common report was also negatively related to project success (Rusinko, 1993).

Second, the significant and negative relationship between 'hardware and software design experience' and project success was in the opposite direction of

that predicted. It was expected that developers with an understanding and experience in both 'thought worlds' would provide a better understanding of the whole project, versus an 'us and them' mentality during the design stages. But this variable was significantly negatively related to both combined and individual success measures. Rusinko's study again found similar results. She measured the effect of frequency of rotation (between design and manufacturing) and project success where it was expected that workers who rotated and gained mutual experience would facilitate project success. The findings of this study on hardware-software coordination enforce Rusinko's findings on this independent variable.

Third, the significant and negative relationship between 'physical distance' and project success was as predicted. However, 'physical distance' was only negatively related to cost goal success. Co-location was also predicted to facilitate coordination and lead to a less ambiguous environment. This study would indicate that closer teams are easier to manage at least in terms of project goals such as cost.

Fourth, the significant and negative relationship between 'extent of shared design tools' and project success was in the opposite direction as predicted. This variable was significantly and negatively correlated to both combined measures of success. Shared design tools were thought to encourage the coordination of the interdependent tasks of hardware and software development.

From testing the model on sample it seems that organizational level initiatives (number of levels to common report and physical distance) facilitate coordination and reduce negative outcomes of differentiation between functions. Conversely,

initiatives to manage interdependence (having both hardware and software design experience and shared tools) seem to impede project success. One explanation of the results for organizational-level initiatives may be their ability to decrease the negative outcomes of functional differentiation by decreasing conflict, barriers and misunderstandings. In the sample projects, initiatives to manage interdependence through the use of shared hardware and software design experience and design tools seemed to impede project success. It may therefore better suit managers to facilitate orientation towards specific expertise, rather than mutual experience.

Task difficulty also had a moderating effect on the model. Developers and project managers view task difficulty differently and there was some evidence of a moderating effect on the relationships between 'number of levels to common report' and 'hardware and software design experience' with project success.

Recommendations for Managers

Understanding the differences between hardware and software developers and the problems these two groups have during product development is a significant issue for project managers during product development of complex electronic systems. This study focused on the effects organizational-level and technological (project level) coordination mechanisms had on project success.

It was proposed that differentiation between hardware and software teams can be managed by reducing barriers, misunderstandings and perceptual differences with regards to project goals. It was also proposed that task interdependence between hardware and software developers could be managed through actions taken at the

group level, for example by use of shared design documents and tools between hardware and software developers.

The results of this study show that actions taken at the organizational level to facilitate coordination and communication between hardware and software teams are important to project success. Managers of projects which are similar to those examined in this study may want to consider concentrating their energies and resources on decreasing conflict, misunderstandings and perceptual differences between hardware and software developers during product development. For example, assuring that problems can be resolved at the lowest level of authority over both hardware and software teams, if not at their levels.

Distance between teams is also crucial to project success in this sample. Managers of similar projects to those examined in this study may want to keep in mind that projects which have hardware and software teams that are closer to each other are more likely to meet project cost goals. In some cases it is not possible to co-locate and therefore managers should keep in mind the disadvantages of distance and facilitate coordination and communication wherever possible. This study showed there to be strong positive correlations between management support for team-level coordination or communication initiatives and project success.

Recommendations for Future Research

This study was limited to a small sample that was self-selected. This sample may have made possibly significant relationships between the hypothesized independent variables and project success more difficult to find. Future research on

this subject should attempt to obtain a larger and more random sample size. Because the sample comes from organizations in high-tech industries it may be better to draw the sample by using email surveys, allowing data collection to be completed more quickly, easily.

When testing the independent variables it would be better to have one questionnaire for all three respondents versus having different questionnaires for each respondent. This would allow the researcher to more easily obtain a higher number of cases per predictor variable. As well, it would assure that all three types of respondents answer questions in the same context.

It is also recommended that future research pursue the effect 'hardware-software design experience' has on project success especially since Rusinko's study found similarly unexpected results. This independent variable had a strong negative correlation with project success, opposite the direction predicted in this study's hypotheses. This outcome strongly disagrees with other research which suggests that personal dissimilarities, such as background, age, education, social patterns lowers the probability of interpersonal rapport between unit representatives, and in turn decreases the amount of collaboration between their respective units (Walton and Dutton, 1969: 77). It also strongly disagrees with a comment made by one engineering consultant:

"The best engineers are those who have experience in both worlds, because they understand optimality in both paradigms, and can make

educated trade-offs, instead of simply getting into biased matches with their counterpart, making it an ego-based argument. (Smith)”

It is also recommended that future research pursue the negative relationship between ‘extent of shared design tools’ and project success.

References

- Affi, A. A. and Virginia Clark (1990) *Computer-Aided Multivariate Analysis* (2nd ed.)
New York: Van Norstrand Reinhold Company
- Cooper, R.G., and E.J. Kleinschmidt, "New Products: What separates winners from losers?" *Journal of Product Innovation Management*, vol. 4, pp. 169-184, 1987.
- Dougherty, Deborah (1992). "Interpretive Barriers to Successful Product Innovation in Large Firms", *Organization Science*, 3(2), May, 179-202.
- Emory, W.C. and Cooper, D.R., (1991). *Business Research Methods*. (4th ed.) Boston, MA: IRWIN.
- Fawcett, Bradley K., (1994). "Tools to speed FPGA development", *IEEE Spectrum*, November, 88-94.
- Kalavade, Asawaree and Lee, Edward A.,(1993). "A Hardware-Software Codesign Methodology for DSP Applications", *IEEE Design and Test of Computers*, September, 16-17.
- Lawrence, P.R. and Lorsch, J.W. (1967). *Organization and Environment: Design and Methods*. Newbury Park, CA: Sage Publications.
- Mancini, Giovanni (1994). *H/W-S/W Co-Verification in ATM*. Bell Northern Research, Ottawa, Ontario.

McCann, J.E. and Galbraith, J.R. (1981). Interdepartmental Relations. In P.C. Nystrom and W.H. Starbuck (Eds.), *Handbook of Organization Design* (pp. 60-84). New York: Oxford University Press.

Rusinko, Cathy (1992). Mechanisms that facilitate design for manufacturability: an exploratory study. Ph.D. Thesis. Ann Arbor, MI: UMI.

Rusinko, Cathy, Susman, Gerald I. and James W. Dean (1993). "Factors that Facilitate Effective DFM: A test of a grounded-theory model." Paper in Progress. Academy of Management TIM/POS Symposium, 1993.

Siegel, S., and N.J. Castellan, Jr. (1988) *Nonparametric Statistics for the Behavior Sciences*, 2nd ed. Toronto: McGraw-Hill, Inc.

Smith, Clive. Thinklabs, Englewood, CO - Private Communication.

Taylor, Kenneth Wayne (1991). *Social Science Research: Theory and Practice*. Scarborough, Ont: Nelson Canada.

Tabachnick, B.G., and L.S. Fidell. (1989) *Using Multivariate Statistics*, 2nd ed. California State University, Northridge: Harper Collins Publishers, Inc.

Thompson, J. (1967). *Organizations in Action*. New York: McGraw-Hill.

Walton, Richard E. and John M. Dutton (1969). "The management of inter-departmental conflict: A model and review". *Administrative Science Quarterly*, 14, 73-84.

Yin, Robert K. (1989). *Case Study Research: Design and Methods*. Newbury Park, CA:
Sage Publishing.

**Appendix A: One Page Summary of Research and Survey sent to
Participating Organizations**

Survey Participants Required:
**Managerial Coordination between Hardware and Software Development
 During Complex Electronic System Design**

Christine Koch
 ckoch@vcn.bc.ca
 (604) 899-2492

Why do some projects that require hardware and software co-development easily meet time, functionality, and cost goals, while others have considerable difficulty? Why do hardware and software development teams often have opposing views? I am writing a Masters thesis in the School of Business at Carleton University that examines these and other issues from a managerial perspective. Part of my thesis research entails a questionnaire study, and I am looking for individuals and / or organizations interested in participating. Some of the companies already participating in my survey are: The Boeing Company, Mitel, Glenayre, Northern Telecom, Hewlett-Packard, Lumonics and Fisher & Paykel.

The Study Thus Far

The hardware and software components of complex electronic systems are typically designed simultaneously, and coordination between the two design processes is crucial to product success in the market place. However, since the two design teams work concurrently, typically have different educational backgrounds, and work with different design tools and design languages, their coordination can be difficult. A completed case study illustrates the differences between hardware and software design and the use of certain coordination mechanisms such as group-based evaluations and use of hardware and software design tools. The case study also shows how other factors such as number of levels to common report, percentage of designers who have both hardware and software design experience, and physical location of design teams affected design project success in terms of design cost, interval, product functionality and quality. Insights gained from the case study have been used to develop hypotheses about the general effectiveness of different coordination mechanisms. The next step is to test these hypotheses with a questionnaire survey.

Questionnaire

I hope to obtain a sample of twenty projects from industries such as telecommunications, computers or aviation. For each project I would like to administer a six-page questionnaire to the project manager and one or two hardware and software developers. Projects need to have been completed within the past year and the hardware and software need to have been designed at the same time. The questionnaire should not take longer than thirty minutes to complete and will cover the issues mentioned above.

Benefits of Participation

After I have conducted the appropriate statistical analyses I will provide each participant with a summary of the findings from their project(s) specifically as well as a report comparing them with other projects in the industry or across industries. All information regarding participants and his / her organization will be kept confidential. The findings of this study will help product development managers better coordinate hardware and software in order to improve their product development process.

If you are interested or know of anyone who may be interested in participating in my survey I can be contacted at (604) 899-2492 or ckoch@vcn.bc.ca

Appendix B: Measures for Project Data Set Variables

Measures for Project Data Set Variables

q1	respondent's professional title
project	identification number for each project (1-18)
q5	length of project in months
q8a	number of managers (from mgr. Perspective) on each project
q8levels	number of management levels (from mgr. Perspective) on each project
q11pm	cost difficulty from the manager's perspective
q12pm	design cycle difficulty from the manager's perspective
q13pm	functionality difficulty from the manager's perspective
q14pm	cost goal success from the manager's perspective
q16pm	design cycle goal success from the manager's perspective
q18pm	functionality goal success from the manager's perspective
q11hs	average hw and software developer response for cost difficulty
q12hs	average hw and software developer response for design cycle difficulty
q13hs	average hw and software developer response for functionality difficulty
q14hs	average hw and software developer response for cost goal success
q16hs	average hw and software developer response for design cycle goal success
q18hs	average hw and software developer response for functionality goal success
pq2	number of people involved on project
pq3	cost of project according to the project manager
pq7	number of developers with both hardware and software design experience on each project according to the project manager
hs8av	average hardware and software developer response (per project) for distance between hardware and software teams
pq17	method of evaluating design teams according to the project manager
hs17av	average hardware and software developer response (per project) for number of design documents shared by both hardware and software design teams
hs20av	average hardware and software developer response (per project) for number of design tools shared by both hardware and software design teams
succpm	average of q14pm, q16pm and q18pm – the overall project success measure from the project manager's perspective
succhs	average of q14hs, q16hs and q18hs – the overall project success measure from the developer's perspective

Appendix C: Project Manager, Hardware and Software Developer Questionnaires

Project Manager Survey

Name of company you work for: _____

Name of project specified for this survey: _____

(this information will be kept confidential)

General Background Information

Project Manager

1. What is your current job title? _____
2. What is the highest level of education that you have attained? _____
3. How old are you? _____

Your Organization

4. Approximately how many people are employed in your organization within the city in which you are located? _____
5. In which industry or industries does your organization work? (If more than one please specify)

1. Consumer	_____
2. Industry	_____
3. Military/Defence	_____
4. Government	_____
5. Other	_____

Project specified for this survey

6. From product conception to full-scale production, how long did this project take? _____ Years _____ Months
7. Approximately how many people (including hardware and software developers as well as project managers and any other individuals) in your organization were involved on this project? _____
8. The product which was developed for this project was targeted to which of the following markets? (If more than one please specify)

1. Consumer	_____
2. Industry	_____
3. Military/Defence	_____
4. Government	_____
5. Other	_____

9. "Initially, the project goals were straightforward and unambiguous in that everyone involved agreed on the problems and 'how solutions could be achieved.'" 1 2 3 4 5 6 7
strongly disagree strongly agree
10. Approximately how much money did your organization invest in this project? \$ _____

Number of levels to common report

11. Here are three examples of what a system development group may look like. Please draw a similar diagram which represents the system development group of the project which was selected for this survey, indicating which position you held and the hardware and software groups you were responsible for.
- Please draw diagram here.
- mg = manager
sw = software team
hw = hardware team
pmg = project manager

12. How often were you involved in the decision to make design changes? 1 2 3 4 5 6 7
never always
13. How often did the hardware developers ask for your assistance during the design process? 1 2 3 4 5 6 7
never always
14. How often did the software developers ask for your assistance during the design process? 1 2 3 4 5 6 7
never always

Hardware and software design experience

15. How many developers had both product hardware and software design experience for the project specified for this survey ? _____
16. Of the designers who had both product hardware and software design experience, how many of them were **hardware** developers? _____
17. Of the designers who had both product hardware and software design experience, how many of them were **software** developers? _____

Physical distance

18. How many **hardware** teams were there on the project specified for this survey? _____
19. Can you tell me where each **hardware** team was located? (i.e. city and province or state)
- HW1: _____
- _____
- HW2: _____
- _____
- HW3: _____
- _____
20. How many **software** teams were there on the project specified for this survey? _____

SW2: _____

SW3: _____

- 1 2 3 4 5 6 7 8
never always n/a

23. On average, how often did you organize group meetings between hardware and software developers during this project? (please circle the appropriate response)

1. every day
2. at least once a week
3. between 1-2 times a month
4. less than once a month
5. never

- 1 2 3 4 5 6 7
never always

- 1 2 3 4 5 6 7
never always

- 1 2 3 4 5 6 7
never always

27. When you made evaluations, did you evaluate individuals, the hardware and software teams alone, or the hardware and software teams combined?

1. Individuals
2. Hardware and software teams alone
3. Hardware and software teams combined
4. A combination of the above? (please specify)

28. When dealing with this project, did you reward individuals, the hardware and software teams alone, or the hardware and software teams combined?

1. Individuals
2. Hardware and software teams alone
3. Hardware and software teams combined
4. A combination of the above? (please specify)

5. None of the above

Task Difficulty

29. Can you please identify how the project for this survey fits into the following matrix. (put an X)

design process	technology new to the firm		
	technology existing in the firm		
		existing market	new market
		customer	

30. "Initially, I felt that it would be very easy to meet the project's cost goals."

1 2 3 4 5 6 7
strongly disagree strongly agree

31. "Initially, I felt that the design cycle schedules that were set would be very easy to meet."

1 2 3 4 5 6 7
strongly disagree strongly agree

32. "Initially, I felt that it would be very easy to meet all of the product's functionality goals."

1 2 3 4 5 6 7
strongly strongly
disagree agree

Project Success

33. "In the end, the cost goals were easily met."

1 2 3 4 5 6 7
strongly strongly
disagree agree

34. "The cost goals were crucial to the project's success"

1 2 3 4 5 6 7
strongly strongly
disagree agree

35. "In the end, all of the design cycle schedules were easily met."

1 2 3 4 5 6 7
strongly strongly
disagree agree

36. "Meeting all of the design cycle schedules was crucial to the project's success"

1 2 3 4 5 6 7
strongly strongly
disagree agree

37. "In the end, the product's functionality goals were easily met."

1 2 3 4 5 6 7
strongly strongly
disagree agree

38. "The project's functionality goals were crucial to the project's success"

1 2 3 4 5 6 7
strongly strongly
disagree agree

39. After the product's initial system design specifications were established, how much functionality originally allocated to hardware was in the end implemented in software?

1 2 3 4 5 6 7
none more
at all than
half

40. Why was functionality from hardware design implemented into software?

1. Because of time constraints
2. Because of product's performance
3. Not applicable
4. Other reasons (please specify)

41. After the product's initial system design specifications were established, how much functionality originally allocated to software was in the end implemented in hardware?
1	2	3	4	5	6	7	110
none							more
at all							than
							half
42. Why was functionality from software design implemented into hardware?
1. Because of time constraints
 2. Because of product's performance
 3. Not applicable
 4. Other reasons (please specify)
- _____
- _____
43. During this project, how much functionality had to be removed from the product?
1	2	3	4	5	6	7
none						more
at all						than
						half
44. During the project, how much functionality had to be added to the product?
1	2	3	4	5	6	7
none						more
at all						than
						half
45. "I really enjoyed working on this project"
1	2	3	4	5	6	7
strongly						strongly
disagree						agree

Hardware Developer Survey

Name of company you work for: _____

Name of project specified for this survey: _____

(this information will be kept confidential)

General Background Information

Hardware developer

1. What is your current job title? _____
2. For how many years have you been a hardware developer? _____
3. What is the highest level of education that you have attained? _____
4. What other degrees or areas of study have you had?

5. How old are you? _____

Your Organization

6. In which industry or industries does your organization work?
(If more than one please specify)

1. Consumer	_____
2. Industry	_____
3. Military/Defence	_____
4. Government	_____
5. Other	_____

Project specified for this survey

7. From product conception to full-scale production, how long did this project take? _____ Years _____ Months
8. How long were you involved on this project? _____ Years _____ Months

9. Approximately how many hardware developers in your organization were involved on this project?	First quarter of project _____ Second quarter of project _____ Third quarter of project _____ Fourth quarter of project _____																					
10. The product which was developed for this project was targeted to which of the following markets? (If more than one please specify)	1. Consumer _____ 2. Industry _____ 3. Military/Defence _____ 4. Government _____ 5. Other _____																					
11. "Initially, the project goals were straightforward and unambiguous in that everyone involved agreed on the problems and 'how solutions could be achieved.'"	<table border="0"> <tr> <td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td> </tr> <tr> <td>strongly</td><td></td><td></td><td></td><td></td><td></td><td>strongly</td> </tr> <tr> <td>disagree</td><td></td><td></td><td></td><td></td><td></td><td>agree</td> </tr> </table>	1	2	3	4	5	6	7	strongly						strongly	disagree						agree
1	2	3	4	5	6	7																
strongly						strongly																
disagree						agree																

Number of levels to common report

12. Here are three examples of what a system development group may look like. Please draw a similar diagram which represents the system development group of the project which was selected for this survey, indicating which position you held.

Please draw diagram here.

mg = manager
sw = software team
hw = hardware team
pmg = project manager

A.

B.

C.

13. How often was the project manager with authority over both hardware and software development involved in the decision to make design changes?	1	2	3	4	5	6	7	8
	never						always	not applicable
14. How often would you say you could have made design changes, regarding the technical aspects of the project, without a co-worker or manager's authority?	1	2	3	4	5	6	7	
	never						always	
15. When you felt that design changes needed to be made, how often did these changes cause a delay or disruption of work during the hardware development process?	1	2	3	4	5	6	7	
	never						always	

Hardware and software design experience

16. Do you have software design experience? Yes No (go to question 20)

17. If yes, how much? (please circle the appropriate response)
1.	Less than one year
2.	1 - 2 years
3.	3 or more years

18. Did your software design experience help you with the design process on the project specified for this survey?	Yes	No

19. During what part of the design process did your software design experience prove helpful in the project specified for this survey?
(please circle the appropriate response(s))
1. first quarter of design process
 2. second quarter of design process
 3. third quarter of design process
 4. last quarter of design process

Physical distance

20. How many **hardware** teams were there on the project specified for this survey?

21. Where were you located during the project specified for this survey? (i.e. city and province/state – if you were located at more than one site please specify and put the dates of when you were at each site).

22. How often on average did you have informal contact, such as impromptu meetings in hallways, with the software developers during the project?
1. At least once a day
 2. Every few days
 3. Once a week
 4. Two to four times per month
 5. Less than once a month

23. How far was the software development team that you interacted with most often from your work station?
1. In the same building
 2. A short car ride
 3. A long car ride
 4. A short flight
 5. A long flight

24. How often was the project manager willing to provide support (e.g. time or money) for hardware developers when they needed to meet with software developers during this project?
- 1 2 3 4 5 6 7 8
 never always not applicable

25. During this project, how often did you use the following methods to communicate with the software team?

	never						always
written letters or memos	1	2	3	4	5	6	7
electronic mail	1	2	3	4	5	6	7
telephone	1	2	3	4	5	6	7
teleconferencing	1	2	3	4	5	6	7
face-to-face	1	2	3	4	5	6	7

Group-based evaluations

26. How often did the project manager organize group meetings between hardware and software developers during this project?
1. every day
 2. at least once a week
 3. between 1-2 times a month
 4. less than once a month
 5. never
27. During these group meetings, how often would you say hardware and software developers agreed on the project's overall goals and priorities, such as costs, quality, and time schedules?
1	2	3	4	5	6	7
never						always
28. On average, how often would you say were hardware developers able to see issues and problems from a software developer's perspective?
1	2	3	4	5	6	7
never						always
29. "The hardware and software teams felt like one united team on this project"
1	2	3	4	5	6	7
totally agree						totally disagree

Design Documents and Tools

30. How often did you use design documents shared by both hardware and software developers during the project specified for this survey?
1	2	3	4	5	6	7
never						every day

31. What type of design documents were they? (for example, functional analysis documents, detailed specification documents, etc.)

32. Did you use design tools (i.e. simulators, emulators, etc.) for the project specified for this survey?

Yes

No (go to question 35)

33. Please list all of the design tools you used during the project specified for this survey and indicate which ones were new to you (i.e. first time you used them) and which ones were shared by both hardware and software developers.

Design Tool	new to you (put an X)	shared
<hr/>	<hr/>	<hr/>
<hr/>	<hr/>	<hr/>
<hr/>	<hr/>	<hr/>
<hr/>	<hr/>	<hr/>
<hr/>	<hr/>	<hr/>

34. During what part of the design process did you use design tools that were **shared** by both hardware and software developers? (please circle the appropriate response(s))

1. first quarter of design process
2. second quarter of design process
3. third quarter of design process
4. last quarter of design process
5. the entire design process

Task Difficulty

35. "Initially, I felt that it would be very easy to meet the project's cost goals."

1	2	3	4	5	6	7
strongly disagree					strongly agree	

36. "Initially, I felt that the design cycle schedules that were set would be very easy to meet."

1 2 3 4 5 6 7
strongly strongly
disagree agree

37. "Initially, I felt that it would be very easy to meet all of the product's functionality goals."

1 2 3 4 5 6 7
strongly strongly
disagree agree

Project Success

38. "In the end, the cost goals were easily met."

1 2 3 4 5 6 7
strongly strongly
disagree agree

39. "The cost goals were crucial to the project's success."

1 2 3 4 5 6 7
strongly strongly
disagree agree

40. "In the end, all of the design cycle schedules were easily met."

1 2 3 4 5 6 7
strongly strongly
disagree agree

41. "Meeting all of the design cycle schedules was crucial to the project's success."

1 2 3 4 5 6 7
strongly strongly
disagree agree

42. "In the end, the product's functionality goals were easily met."

1 2 3 4 5 6 7
strongly strongly
disagree agree

43. "The product's functionality goals were crucial to the project's success."

1 2 3 4 5 6 7
strongly strongly
disagree agree

44. After the product's initial system design specifications were established, how much functionality originally allocated to hardware was in the end implemented in software?

1 2 3 4 5 6 7
none more
at all than
half

45. Why was functionality from hardware design implemented into software?
1. Because of time constraints
 2. Because of product's performance
 3. Other reasons (please specify)
-
-
-
46. After the product's initial system design specifications were established, how much functionality originally allocated to software was in the end implemented in hardware?
1	2	3	4	5	6	7
none						more
at all						than
						half
47. Why was functionality from software design implemented into hardware?
1. Because of time constraints
 2. Because of product's performance
 3. Other reasons (please specify)
-
-
48. During this project, how much functionality had to be removed from the product?
1	2	3	4	5	6	7
none						more
at all						than
						half
49. During this project, how much functionality had to be added to the product?
1	2	3	4	5	6	7
none						more
at all						than
						half
50. "I really enjoyed working on this project"
1	2	3	4	5	6	7
strongly						strongly
disagree						agree

Software Developer Survey

Name of company you work for: _____

Name of project selected for this survey: _____
(this information will be kept confidential)

General Background Information

Software developer

1. What is your current job title? _____
2. For how many years have you been a software developer? _____
3. What is the highest level of education that you have attained? _____
4. What other degrees or areas of study have you had?

5. How old are you? _____

Your Organization

6. In which industry or industries does your organization work?
 (If more than one please specify)
 1. Consumer _____
 2. Industry _____
 3. Military/Defence _____
 4. Government _____
 5. Other _____

Project specified for this survey

7. From product conception to full-scale production, how long did this project take? _____ Years _____ Months
8. How long were you involved on this project? _____ Years _____ Months

9. Approximately how many software developers in your organization were involved on this project? Please indicate the number of software developers involved at every quarter of the project.
- First quarter of project _____
- Second quarter of project _____
- Third quarter of project _____
- Fourth quarter of project _____
10. The product which was developed for this project was targeted to which of the following markets? (If more than one please specify)
1. Consumer _____
2. Industry _____
3. Military/Defence _____
4. Government _____
5. Other _____
11. "Initially, the project goals were straightforward and unambiguous in that everyone involved agreed on the problems and 'how solutions could be achieved.'"
- 1 2 3 4 5 6 7
- strongly strongly
- disagree agree

Number of levels to common report

12. Here are three examples of what a system development group may look like. Please draw a similar diagram which represents the system development group of the project which was selected for this survey, indicating which position you held.

Please draw diagram here.

mg = manager
sw = software team
hw = hardware team
pmg = project manager

13. How often was the project manager with authority over both hardware and software development involved in the decision to make design changes?
1	2	3	4	5	6	7	8
never						always	not
14. How often would you say you could have made design changes, regarding the technical aspects of the project, without a co-worker or manager's authority?
applicable						
1	2	3	4	5	6	7
never						always
15. When you felt that design changes needed to be made, how often did these changes cause a delay or disruption of work during the software development process?
1	2	3	4	5	6	7
never						always

Hardware and software design experience

16. Do you have hardware design experience?

Yes

No (go to question 20)

25. During this project, how often did you use the following methods to communicate with the hardware team?

	never						always
written letters or memos	1	2	3	4	5	6	7
electronic mail	1	2	3	4	5	6	7
telephone	1	2	3	4	5	6	7
teleconferencing	1	2	3	4	5	6	7
face-to-face	1	2	3	4	5	6	7

Group-based evaluations

26. How often did the project manager organize group meetings between hardware and software developers during this project?
1. every day
 2. at least once a week
 3. between 1-2 times a month
 4. less than once a month
 5. never
27. During these group meetings, how often would you say hardware and software developers agreed on the project's overall goals and priorities, such as costs, quality, and time schedules?
1	2	3	4	5	6	7
never						always
28. On average, how often would you say were software developers able to see issues and problems from a hardware developer's perspective?
1	2	3	4	5	6	7
never						always
29. "The hardware and software teams felt like one united team on this project"
1	2	3	4	5	6	7
totally agree						totally disagree

Design Documents and Tools

30. How often did you use design documents shared by both hardware and software developers during the project specified for this survey?
1	2	3	4	5	6	7
never						every day

31. What type of design documents were they? (for example, functional analysis documents, detailed specification documents, etc.)

[illegible]

32. Did you use design tools (i.e. simulators, emulators, etc.) for the project specified for this survey?

Yes

No (go to question 35)

33. Please list all of the design tools you used during the project specified for this survey and indicate which ones were new to you (i.e. first time you used them) and which ones were shared by both hardware and software developers.

Design Tool

new to you
(put an X)

shared

34. During what part of the design process did you use design tools that were **shared** by both hardware and software developers? (please circle the appropriate response(s))

1. first quarter of design process
2. second quarter of design process
3. third quarter of design process
4. last quarter of design process
5. the entire design process

Task Difficulty

35. "Initially, I felt that it would be very easy to meet the project's cost goals."

1 2 3 4 5 6 7
strongly strongly
disagree agree

36. "Initially, I felt that the design cycle schedules that were set would be very easy to meet."	1 2 3 4 5 6 7
strongly
disagree	strongly
agree
37. "Initially, I felt that it would be very easy to meet all of the product's functionality goals."	1 2 3 4 5 6 7
strongly
disagree	strongly
agree

Project Success

38. "In the end, the cost goals were easily met."	1 2 3 4 5 6 7
strongly
disagree	strongly
agree
39. "The cost goals were crucial to the project's success."	1 2 3 4 5 6 7
strongly
disagree	strongly
agree
40. "In the end, all of the design cycle schedules were easily met."	1 2 3 4 5 6 7
strongly
disagree	strongly
agree
41. "Meeting all of the design cycle schedules was crucial to the project's success."	1 2 3 4 5 6 7
strongly
disagree	strongly
agree
42. "In the end, the product's functionality goals were easily met."	1 2 3 4 5 6 7
strongly
disagree	strongly
agree
43. "The product's functionality goals were crucial to the project's success."	1 2 3 4 5 6 7
strongly
disagree	strongly
agree
44. After the product's initial system design specifications were established, how much functionality originally allocated to hardware was in the end implemented in software?	1 2 3 4 5 6 7
none
at all	more
than
half

45. Why was functionality from hardware design implemented into software?
1. Because of time constraints
 2. Because of product's performance
 3. Other reasons (please specify)
-
-
-
46. After the product's initial system design specifications were established, how much functionality originally allocated to software was in the end implemented in hardware?
1	2	3	4	5	6	7
none						more
at all						than
						half
47. Why was functionality from software design implemented into hardware?
1. Because of time constraints
 2. Because of product's performance
 3. Other reasons (please specify)
-
-
48. During this project, how much functionality had to be removed from the product?
1	2	3	4	5	6	7
none						more
at all						than
						half
49. During this project, how much functionality had to be added to the product?
1	2	3	4	5	6	7
none						more
at all						than
						half
50. "I really enjoyed working on this project"
1	2	3	4	5	6	7
strongly						strongly
disagree						agree