UNIVERSITY OF TURKU

ABSTRACT

	Bachelor's thesis
X	Master's thesis
	Licentiate's thesis
	Doctoral dissertation

Subject	Work Informatics / information systems science	Date	8.11.2021
Author(s)	Juha Seppänen	Number of pages	69
Title	Impact of developer–designer collaboration behaviour for ISD project success		
Supervisor(s)	Jani Koskinen, Ph.D.		

Abstract

Information systems development (ISD) has always been a complicated business, and it has been getting more complicated over time. Failure is still commonplace despite advances in project methodologies, techniques, and tools.

In this thesis, two ISD roles, the designer and the developer, were studied. Seven informants were interviewed — four designers and three developers. The research goal was to find out how the designers and the developers collaborated during the ISD project and what impact their collaborative effort may have on the result of the developed information system.

The material was collected via interviews and in-situ observation of the workspaces. A semi-structured interview method was used where a set of structured interview questions was used as a basis for interview sessions but also let the discussion flow into directions where it was naturally heading.

The interviews were transcribed and coded into the NVivo system by interview themes. Emergent themes were added during text analysis, and transcripts were analyzed again to ensure the emerged themes could be analyzed as well. Observation notes were added to the set of codes and mixed with the interview data.

According to the research results, an ISD's success depends on the fluent collaboration of the designer and the developer. Multiple factors were found that lower the risk of ISD failure. Even though their roles are somewhat indeterminate and vary slightly between different companies and teams, these two roles are essential for the success of an ISD project. The role of a customer emerged as the essential stakeholder during an ISD project. All informants mentioned that without clear lines of communications, commonly agreed on goals with milestones, and an overall common understanding of the IS to be developed, the likelihood of failure increases.

Key words	developer, designer, collaboration, tools, roles, success, failure, ISD, SDLC

TURUN YLIOPISTO

TIIVISTELMÄ

	Kandidaatintutkielma
X	Pro gradu -tutkielma
	Lisensiaatintutkielma
	Väitöskirja

Oppiaine	Työinformatiikka	Päivämäärä	8.11.2021
Tekijä(t)	Juha Seppänen	Sivumäärä	69
Otsikko	Kehittäjä–suunnittelija työparin yhteistyön vaikutus tietojärjestelmäprojektien onnistumiseen		
Ohjaaja(t)	filosofian tohtori Jani Koskinen		

Tiivistelmä

Tietojärjestelmäkehitys on aina ollut vaikeaa ja monimutkaista toimintaa, jossa ajan mittaan monimutkaisuus on muuttunut monelta osin kompleksisuudeksi. Epäonnistuneet tietojärjestelmähankkeet ovat edelleen tavallisia huolimatta hankehallinnan menetelmien, tekniikoiden ja työkalujen kehittymisestä.

Tässä tutkielmassa tutkimuskohteeksi valittiin kaksi tietojärjestelmäkehityksen roolia: suunnittelija ja kehittäjä. Seitsemän vapaaehtoista ammattilaista valittiin haastatteluihin, joista neljä oli suunnittelijoita ja kolme kehittäjiä. Tutkimuksen tavoitteena oli saada ymmärrys, miten suunnittelijat ja kehittäjät toimivat tiiviisti yhdessä tietojärjestelmäkehitysprojektin aikana ja millainen vaikutus jos mitään heidän yhteistoimintansa laadulla on tietojärjestelmäprojektin onnistumiseen.

Tutkimusmateriaali kerättiin seitsemän vapaaehtoisen ammattilaisen yksilöhaastattelulla. Tiedonkeruussa käytettiin teemahaastettelumenetelmää.

Haastattelut litteroitiin ja koodattiin NVivo järjestelmään haastattelurungossa mainittuihin teemoihin. Uusien teemojen noustua esiin litteroinnit käytiin läpi uudestaan ja löytyneet koodit lisättiin. Työtilojen havainnoinnin muistiinpanot koodattiin yhdessä haastatteluissa esiin tulleiden kommenttien kanssa.

Tutkimustulosten perusteella tietojärjestelmähankkeen menestys on jossain määrin riippuvaista suunnittelijoiden ja kehittäjien sujuvasta yhteistoiminnasta. Tuloksissa mainitaan useita tekijöitä, joiden huomioonottamisella riskiä voidaan tuntuvasti vähentää. Vaikka roolien määrittely ei ole selkeää voidaan silti sanoa, että tällä työparilla on usein ratkaiseva vaikutus tietojärjestelmähankkeen onnistumiseen. Tämän lisäksi asiakkaan rooli nousi esiin keskeisenä tekijänä suunnittelijoiden ja kehittäjien mielestä.

Avainsanat	kehittäjä, suunnittelija, yhteistyö, työtavat, työkalut, roolit, onnistuminen

UNIVERSITY OF TURKU
Turku School of Economics

IMPACT OF DEVELOPER–DESIGNER COLLABORATION BEHAVIOR FOR ISD PROJECT SUCCESS

A study in collaboration habits of two ISD project roles

Master's Thesis
in Work Informatics

Author:
Juha Seppänen
B.Sc. Computer Science

Supervisor(s):
Jani Koskinen
Ph.D.

8.11.2021
Turku / Rauma

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 Background

Failure in ISD is nothing new. Ackoff (1967) wrote about Management Misinformation Systems and missing their objectives at the end of the 1960s. Brooks cited the "software crisis" in his seminal work "The Mythical Man-Month" in which he explored the reasons for failure for information systems development (Brooks Jr 1995). The term "software crisis" was invented by NASA researchers Naur & Randell (1969).

Information Systems Development (ISD) is a complicated or even complex activity that requires effort from various ISD professionals and stakeholders. Complexity is about living at the edge of chaos (Waldrop 1993). Complexity is evident in both components of the IS: the software component and the human or organizational component. Organizational research has tried to tackle the organizational component (Ackoff 1967). Like Brooks and Kugler stated (1987), there is no silver bullet. The conceptual difficulties in expressing systems remain (Brooks 1986).

ISD project failure rate has stayed relatively constant despite significant advances in project management methodologies, technology, and business management techniques (The Standish Group International 2015; Sauer 1999). Interestingly there was a plateau in the number of reported ISD project failures in the 1980s, suggesting that some problems were solved (Sauer 1993). It proved not to be the case (Sauer 1993). The increasing complexity of software-based systems may offer a partial explanation for this. For example, NASA did a study comparing flight software complexity over time (Dvorak 2009). Software-based functionality in aircraft has increased from less than 10% (the F-4) in 1960 to 80% in the year 2000 (the F-22) (Dvorak 2009). Similar results are reported elsewhere. Standish Group believes big projects are ten times less likely to succeed than small projects (Standish Group International 2015). Complexity increases due to adding software elements to cars were also reported:

"In 1970 the average General Motors automobile had a hundred thousand lines of code, but now a premium-class automobile has close to 100 million lines of code running on 70 to 100 microprocessor-based electronic control units."

- (Charette 2009; Dvorak 2009)

Then again – successes do not exist without failures (Fincham 2002; Sauer 1993). Despite failures, the information systems development and use have been quite successful business over the past decades (Baghizadeh, Cecez-Kecmanovic, and Schlagwein 2020). Software-based systems have allowed multiple advances in societies and businesses in general. Nevertheless, only marginal improvements have been reported despite the improvements in ISD projects' management and the methodologies, tools & techniques. What is there to do?

This thesis aims to deepen the understanding of IS development part of the ISD project success factors through literature review and interviews of two ISD project roles, the designer and the developer. The DeLone and McLean model of IS success is used to evaluate the research findings (DeLone and McLean 2003). As the designer and the developer work is focused on the first two categories of the DeLone and McLean model of IS success – the Systems Quality and the Information Quality, the analysis is limited to those. In the discussion, we expand on that and reflect on User Satisfaction, Individual Impact, and Organizational Impact.

In the analysis part, the informants' narratives are enumerated and reflected upon those that would indicate a success in the "in use" category of a D&L Model (DeLone and McLean 2003). Designers' and developers' individual and collaborative contributions to the overall ISD project are analyzed. Narratives are also projected against the other roles often found in the ISD projects mentioned above.

Alter (2008) lists 20 definitions for an information system. The definitions differ mostly on their emphasis. Most definitions include the technological component, a human component with goal-oriented human activities. Few define the IS in more simple terms. Kroenke et al. (Kroenke, Bunker, and Wilson 2013) simplify the definition to "...a group of components that interact to produce information". Łuba & Rybnik (1992) give us perhaps the most straightforward yet challenging to grasp definition: "An information system is a pair A=(U, A), where U is a non-empty, finite set called the universe, and A is a non-empty, finite set of attributes".

In addition to being technical systems, the information systems are also increasingly organizational and communication systems (Iivari and Koskela 1987). For this thesis, a definition is selected that includes both human components and technological components with human activity. The following definition by Buckingham et al. (Buckingham et al. 1986) is used in this thesis.

> *"A system which assembles, stores, processes and delivers information relevant to an organization (or to society), in such a way that the information is accessible and useful to those who wish to use it, including managers, staff, clients and citizens. An information system is a human activity (social) system which may or may not involve the use of computer systems."* (Buckingham et al. 1986)

Information Systems development lifecycle is thus a description of the various phases, events, and activities put into a linear timeline. It is true regardless of chosen development model or methodology employed in the ISD project.

Looking back to the history of systems development, several distinct eras have been identified. Avison & Fitzgerald list these eras as follows: pre-methodology era, early methodology era, methodology era, and post-methodology era (Avison and Fitzgerald 2003).

In the pre-methodology era in the 1960s and the 1970s, systems were developed without described or formal methodology. Choices in tools and techniques were made on an individualistic basis by the programmers (developers) that often resulted in poor quality of the projects for all stakeholders. User or business requirements were not understood as an important part of the development process. Experience with systems developed without rules or standards led to the development of early systems development methodologies (Avison and Fitzgerald 2003).

Systems Development Lifecycle (SDLC), also known as the waterfall model, was adopted to put structure and control into the systems development process. It introduced the concept of phases that followed each other in linear order. It included the following phases: feasibility study, systems investigation, analysis, design, development, implementation, and maintenance (Royce 1970). A phase had to be completed before one could move on to the next phase. Outputs of a phase had to be inspected and approved to proceed (Avison and Fitzgerald 2006). Despite common misconception, iteration between the phases was part of SDLC methodology, although it was often left unused in practice.

SDLC left still plenty of room for improvement. Multiple new methodologies emerged. A methodology is a recommended collection of practices, procedures, rules, techniques, tools, documentation, management and training used to develop a system (Avison and Fitzgerald 2003). Avison & Fitzgerald also highlight the importance of the underlying philosophy behind the methodology - a set of assumptions and beliefs behind a particular methodology (Avison and Fitzgerald 2003). A set of seven themes emerged:

- Structured,
- Data-oriented,
- Prototyping,
- Object Oriented,
- Participative,
- Strategic and
- Systems (Avison and Fitzgerald 2003).

Many of those above have been utilized together as a best practice mix. It is common to take certain parts of a documented methodology and apply only select parts of it. For example, one can use data flow diagramming from the structured approach. One can use aspects of the prototyping to test the assumptions and designs together with participative techniques where end-user involvement was needed to test the assumptions (Avison and Fitzgerald 2003).

The methodology era did not result in harmonizing different approaches to systems development or the wide adoption of any single methodology. Mixed methodologies became commonplace, and some practitioners abandoned prescribed methodologies altogether. More interest was placed in understanding the underlying thinking and concepts behind collecting a situational and context-suited set of techniques, tools, and procedures (Avison and Fitzgerald 2003). A lack of productivity gains, complexity, and rigidity were listed as reasons for discarding methodologies (Avison and Fitzgerald 2003).

The idea that no detailed ISD methodology fits all situations is central to the PIOCO model (Iivari and Koskela 1987). This idea is good but complicates the ISD for the participants. The participants should all know different ISD methodologies, methods, techniques, and tools. They should use this knowledge to pick a suitable mix for any given situation (Iivari and Koskela 1987).

1.2 Research focus

Roles are used to assigning authority and responsibility and divide responsibilities within organizations and describe the ways of working through role descriptions and role interaction (Zhu, Zhou, and Seguin 2006). The roles that have clear definitions help their holders to collaborate effectively (Ashforth 2000). Unclear or ambiguous role definitions may create dysfunction and conflict in an organization (Bostrom 1980). In collaboration situations, roles help team members to focus on a particular topic assigned to them in the role descriptions. The use of roles clarifies the collaboration (Zhu, Zhou, and Seguin 2006; Biddle and Thomas 1966).

"In fact, in a social environment, roles are taken as a tool to specify human behavior." (Zhu, Zhou, and Seguin 2006)

The role of a designer is ambiguous. A designer may design service, design the interaction of a user interface of a system. A designer may design the visual aspects of a user interface, research human factors. Often a designer specifies and designs the human factors in a user interface or any combination of those mentioned earlier. In addition, the role of a designer may refer to a software designer (Brooks 1986). For this thesis, a designer's role is constrained to aspects where he/she collaborates with the developer role during an information systems development project to produce a functioning IS that serves a business need.

The role of a developer is less ambiguous than the role of a designer. Commonly it is understood as the one who writes the software, codes. However, it can mean other aspects of software design, such as specifying the information systems architecture at various levels. For clarity, the role definition of a developer used in Scrum (Sutherland and Schwaber 2020) is explicitly excluded here. The in-depth analysis of the collaboration between these two roles would be impossible if every role were labeled "a developer". For this thesis, the role of a developer is constrained to aspects of ISD where the developer collaborates with the designer during an ISD lifecycle producing a functioning IS that serves a business need.

Together the roles of designer and developer form a tuple that is a foundational element in the ISD team. Together with other ISD stakeholders, the team is the one tasked to work through the ISD lifecycle and deliver the working software for the IS. These two roles look at the ISD from a slightly different point of view. Whether the difference is significant enough to justify the chosen research focus remains to be seen.

The research question for this thesis is: What factors the designer – developer collaboration could affect to increase the likelihood of ISD success?

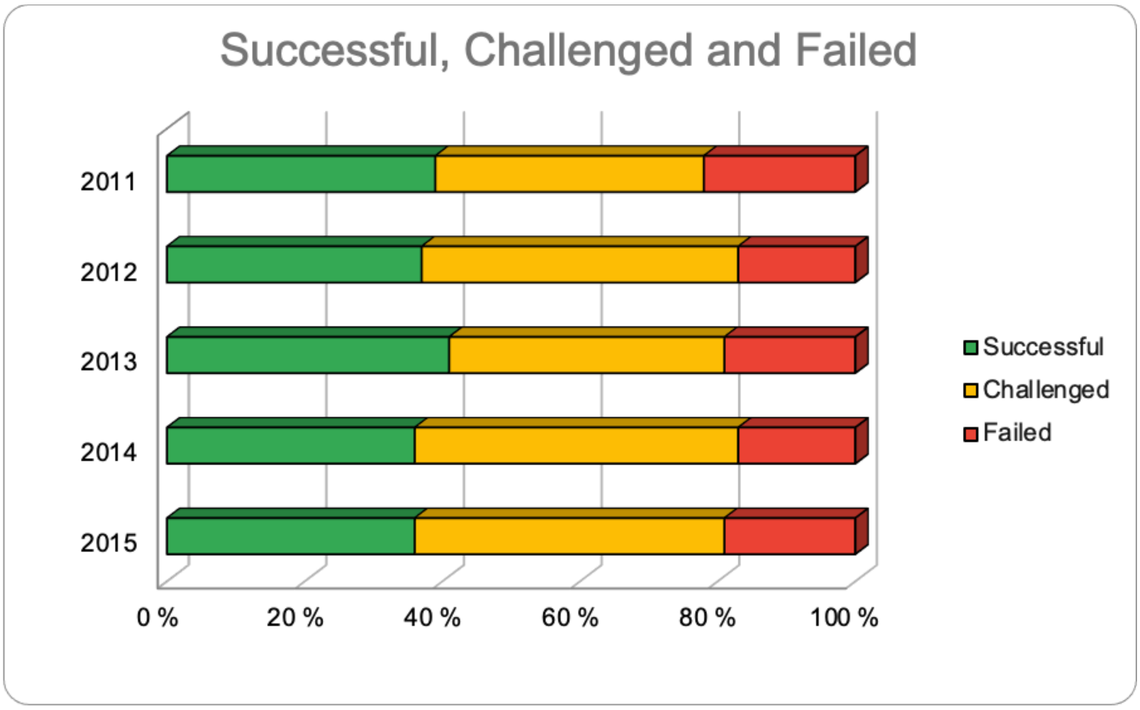
This thesis is structured in the following way. The first chapter gives an introduction and background to the topic as well as states the research question. The second chapter gives an overview of the relevant literature within the field of study. The third chapter specifies the research methods. In chapter four, the findings are reported in detail. Chapter five discusses the research findings in context with DeLone & McLean's IS Success Model. Finally, in chapter six, the conclusions are drawn.

2 LITERARY REVIEW

ISD project failures have been documented extensively throughout the IS development and systems development history (Baghizadeh, Cecez-Kecmanovic, and Schlagwein 2020). Failures of ISD projects have been documented with such rigor that ISD failure is now expected. If not by the professionals but at least by the public. The public's interest is mainly in ISD project failures due to the extensive amount of money spent and lost in ISD failures (The Standish Group International 2015).

The Chaos Report 2015 reports that during 2015 only 29% of ISD projects were considered a success (The Standish Group International 2015).

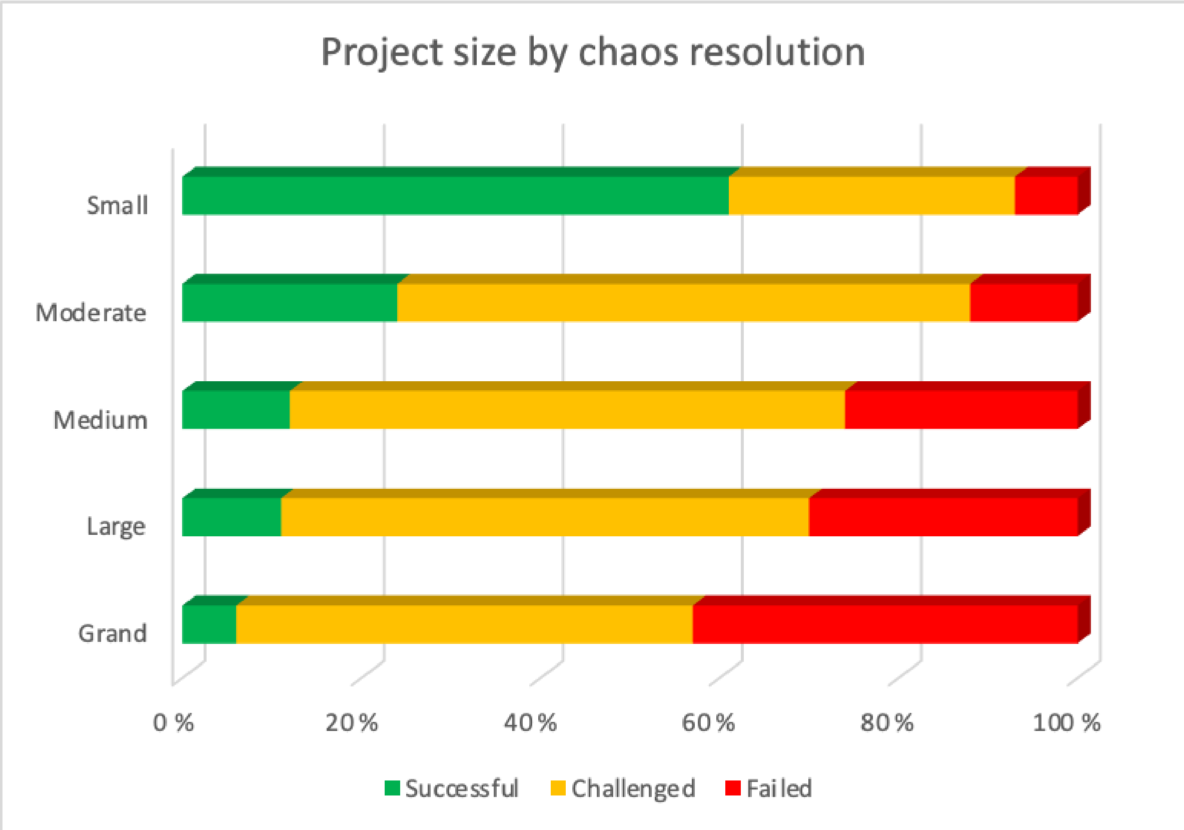
Figure 1 Success and failure of ISD project 2011-2025 (The Standish Group International 2015)



According to the original categories of "Successful", "Challenged" and "Failed" classification Standish Group used in the original Chaos Report from 1995. When adding together "Challenged" and "Failed" projects, the picture of ISD project failure gets quite grim. In 2015 somewhat failed projects amounted to 64 % of all projects.

It gets interesting when project size is inspected. Standish Group believes big projects are ten times less likely to succeed than small projects (Standish Group International 2015). It is shown in Figure 2 Project size (The Standish Group International 2015). Curtis et al. determined that large projects may succeed, but that requires a single exceptional individual to coordinate the project, with deep domain knowledge and excellent software development knowledge (Curtis, Krasner, and Iscoe 1988).

Figure 2 Project size (The Standish Group International 2015)



ISD project failures are still commonplace, as stated in the Standish Chaos Report 2015 (The Standish Group International 2015). An extensive review of ISD project failure research was done by Baghizadeh, Cecez-Kecmanovic & Schlagwein (Baghizadeh, Cecez-Kecmanovic, and Schlagwein 2020). As ISD projects are increasingly complex, there is a need to gain more impactful knowledge about reasons for failure to avoid common pitfalls.

The Chaos Report (The Standish Group International 2015) has used simplistic and determinate success attributes in its reporting. The original attributes of an ISD project were OnBudget, OnTime, and OnTarget. They expanded these definitions in 2015 to include three new attributes to success: OnGoal, Value, and Satisfaction (The Standish Group International 2015). This reflects the need to have more actionable and fine-grained explanations of the success or failure of ISD projects.

However, The Chaos Report (The Standish Group International 2015) and literary review done by Baghizadeh et al. (Baghizadeh, Cecez-Kecmanovic, and Schlagwein

2020) indicate that there is no clear consensus of the definition of what constitutes success or failure. This makes it hard or even impossible to make comparisons of cases in order to analyze them and to find a recipe for success (Baghizadeh, Cecez-Kecmanovic, and Schlagwein 2020). Baghizadeh et al. (2020) further found cases where an ISD project was deemed to be in an indeterminate state – a failure and a success at the same time (Baghizadeh, Cecez-Kecmanovic, and Schlagwein 2020).

2.1 Categories of ISD failures in literature

In order to be able to analyze failures of ISD there has been research done to find categories for the ISD project failures. Fincham (Fincham 2002) has divided ISD project failures into three major categories: the rationalist, the process, and the narrative / interpretative. His categories follow Sauer (Sauer 1999).

The rationalist category focuses on organizational goals. It looks mainly into managerial and organizations as structures for activities. It uses simple causal logic where B is directly caused by A. In it the success or failure is binary. This approach has received critique for providing a static and deterministic view to the ISD project failure and gives little guidance on how to get to the root cause (Fincham 2002; Baghizadeh, Cecez-Kecmanovic, and Schlagwein 2020).

The rationalistic approach has received criticism for abandoning the contextual details and intricacies of ISD projects to have more general applicability over projects with different contexts (Baghizadeh, Cecez-Kecmanovic, and Schlagwein 2020 These abstractions do not necessarily help answer the only relevant question: Why do ISD projects fail (or succeed) (Fincham 2002).

The *process category* focuses on the processes of the organization and looks at socio-technical interaction. The outcome is according to the organizational process. Process category offers a better view into the complexities of ISD projects. It can offer better explanations for why projects fail or succeed. Then again, its focus on the processes often offers simplistic explanations and still expects binary failure/success of the ISD project (Fincham 2002; Baghizadeh, Cecez-Kecmanovic, and Schlagwein 2020).

The *narrative category* focuses on the narratives, stories, and plots. It uses interpretation and sense-making methods to give a non-binary end state of the ISD project. There is no objective success or failure, but the actors within the ISD project socially construct the reality. This approach has the potential to get deeper into the root causes, but as it lacks comparable results due to different contexts between ISD projects, this approach has not gained much support among scholars (Fincham 2002; Baghizadeh, Cecez-Kecmanovic, and Schlagwein 2020).

2.2 Reasons for ISD success or failure found in the literature

Kraut et al. (Kraut and Streeter 1995) raised the impact of coordination as a crucial aspect in ISD project success or failure. They state that in addition to having formal communication in place in projects, there is a need for informal communication processes as well (Kraut and Streeter 1995). Nidumolu (Nidumolu 1995) touched similar topic, looking at the ISD project coordination from both formal coordination and informal coordination viewpoints. There the coordination is looked at from a project performance and risk perspective. Coordination between the user and the IS staff (horizontal coordination) was found to positively impact the quality of the system (Nidumolu 1995). The project management role (vertical coordination) was seen as most important when the risks increased. Vertical coordination did not have a direct effect on the project performance, however (Nidumolu 1995). The combined effects of horizontal and vertical coordination need to be present to have a high-performing ISD project (Nidumolu 1995).

Reasons for failure can be embedded in the roles working in the ISD project. Multiple issues constrain the work of the designers. When designing interactivity, one hurdle is the lack of proper tools (Myers et al., 2008). Multiple different tools and methods to describe the interaction have been tried out. Tools such as Photoshop and Illustrator come up short of allowing designers to describe the interaction of the user interface. In order to add interactivity, one needs to possess skills in coding, whether it is JavaScript or ActionScript (Myers et al. 2008). There is a need to try out different interaction options. With current tools, this process is laborious and time-consuming (Myers et al. 2008).

Failure modes can emerge from the difference in work styles. Designers and developers are not a homogenous bunch of professionals by any means. Work styles of designers vary, and so do the work styles of developers (Campos and Nunes 2007). Nevertheless, they are always expected to work together in a team setting and be effective in producing working software. Campos and Nunes (Campos and Nunes 2007) found that designers and developers employ several different workstyles and switch between them frequently. They call these workstyle transitions (Campos and Nunes 2007). Workstyle transitions each have different costs. Campos and Nunes' research (Campos and Nunes 2007) showed that the costliest transition was the transition from problem to solution space, followed closely by a jump between nonfunctional to fully functional.

Collaboration in groups is often about making decisions (Stasser and Titus 1985). It can also be used to reach a consensus on a topic being discussed. Collaboration can also increase participants' knowledge and expertise on the matter (Stasser and Titus 1985). It provides a way to test and verify assumptions made by another professional from the other expert discipline. These positive outcomes from collaboration can happen, but Strasser and Titus (1985) state that information shared during a collaboration session frequently does not affect the initial preferences of the stakeholders unless the information has been shared before the session so that the participants have been allowed to internalize the new information (Stasser and Titus 1985).

Inter-organizational collaboration has been written about by Paasivaara & Lassenius (Paasivaara and Lassenius 2003). Communication is in a central position in all collaboration practices (Paasivaara and Lassenius 2003). In the context of distributed projects, they identified a set of collaboration practices that are found to be necessary for the context of inter-organizational collaboration. They are:

1) Synchronization of main milestones,
2) Frequent deliveries,
3) Establishment of peer-to-peer links,
4) Problem-solving practices,
5) Informing and monitoring practices and
6) Relationship building practices (Paasivaara and Lassenius 2003).

As this research is focused on the collaboration of designer–developer pair, the inter-organizational aspects seem less relevant. However, it is prudent to look closer at some of the identified practices and investigate them in the context of peer-to-peer collaboration and their effect on the likelihood of success or failure.

Even though the designer–developer pair can quite effortlessly agree on the main milestones, they seldom work in a project that does not need any synchronization of the deliverables between the stakeholders (Paasivaara and Lassenius 2003). At least the main milestones between the stakeholders should be enough. However, it is not always necessary to use the same processes (Paasivaara and Lassenius 2003).

Frequent deliveries are one aspect that certainly can improve communication, thus allowing collaboration to be better with designer-developer context as well. Paasivaara &

Lassenius (2003) found that having frequent deliveries lead to less rework due to difficult or impossible integration of code modules, thus lowering the risk of failure.

Relationship-building practices are also relevant in designer–developer collaboration. Building a good relationship between the two roles can be a key to effective collaboration. Paasivaara & Lassenius (2003) state further that face-to-face contact is required in distributed teams at least some of the time.

Iivari & Koskela (1987), in their seminal article on the PIOCO Model, state that even though the IS specification should be separated from the technical implementation, this is not usually the case. Further, they state that prototyping and evolutionary approaches instead of traditional SDLC or waterfall methods create additional difficulties in project management (Iivari and Koskela 1987). They further suggest that the PIOCO model integrates different and possibly conflicting methods (Iivari and Koskela 1987). The key aspects that they mention are:

1) Decision making orientation – ISD process is an inquiry process that supports decisions to be made concerning the IS to be developed,
2) Contingency approach – flexibility to select the methodologies, methods and tools as situation demands,
3) A balanced organizational, conceptual and technical view – not only the specification, not only the technology but also the organizational view are important aspects to consider when doing ISD,
4) The dynamics of ISD process – a non-linear structure of main phases together with flexible process-mode of project planning instead of blueprint, linear mode for ISD project and
5) IS assessment – IS is assessed through effectiveness, user satisfaction and efficiency criteria (Iivari and Koskela 1987).

Dealing with two specific roles also has risks involved. Bostrom (Bostrom 1980) discussed role conflicts and divided the role conflict or breakdown into three types: person-role conflict, intrasender conflict, and role overload (Bostrom 1980). Person-role conflict is how role expectations are inappropriate with the person's orientations, standards, or values with the role (Bostrom 1980). Intrasender conflict is the extent to which role requirements are incompatible with the capabilities of resources of the person with the role (Bostrom 1980). Role overload is about the role expectations communicated to the person, exceeding the amount of time available for their accomplishment (Bostrom

1980). Intersender conflict is when different messages and pressures from a single member of the role set are incompatible (Bostrom 1980). In addition, there can also be the ambiguity of the role. It is a situation where the expectations sent to the role holder are vague, ambiguous, and unclear, making it difficult for the role holder to fulfill the requirements (Bostrom 1980). Bostrom based his work on Kahn et al., House & Rizzo, and Miles (Kahn et al. 1964; House and Rizzo 1972; Miles 1974).

Even though it is common to talk about ISD failures for decades, there has been a bold attempt by DeLone & McLean (DeLone and McLean 1992) to talk about success. The DeLone & McLean Model of IS Success (DeLone and McLean 1992) uses six categories for IS quality: *system quality, information quality, use, user satisfaction, individual impact and organizational impact*.

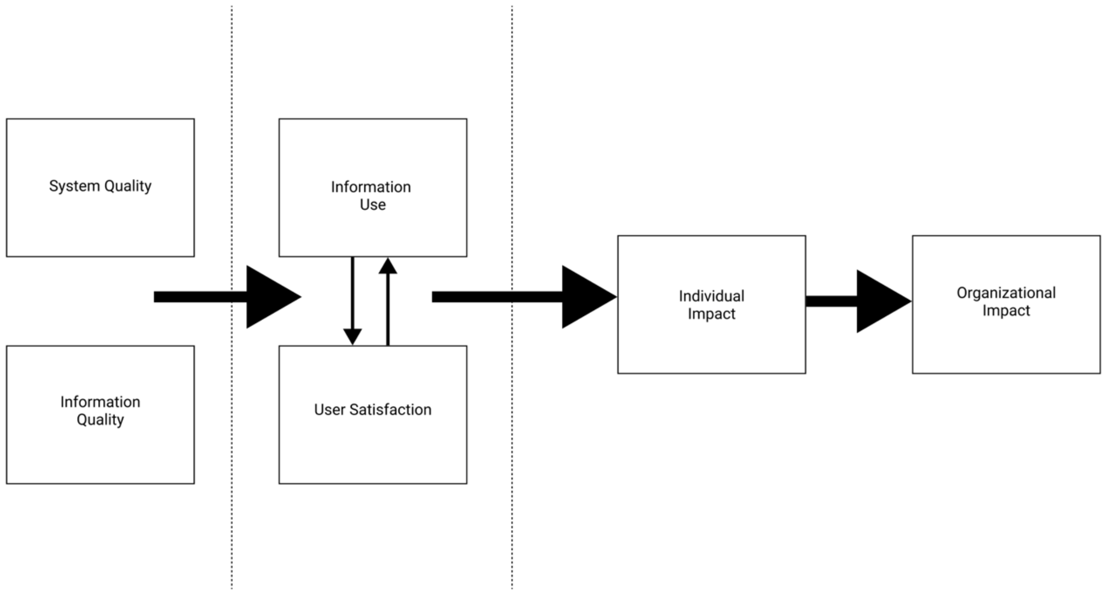


Figure 3 - IS Success Model, based on DeLone & McLean (DeLone and McLean 1992)

DeLone & McLean state that there needs to be a success in all listed categories to consider a IS a success. *System Quality* is concerned with the aspects such as data accuracy and ease of use (DeLone and McLean 1992). *Information Quality* is concerned with the aspects such as importance, relevancy, clarity, format, and appearance (DeLone and McLean 1992). These two are needed for *Information Use & User Satisfaction* to be evaluated (DeLone and McLean 1992). *Individual Impact* follows these two, and in the end, the *Organization Impact* is realized as a sum of individual impacts (DeLone and McLean 1992).

Based on the communications research of Shannon and Weaver (Shannon 1948) and the information "influence" theory of Mason (Mason 1978), as well as empirical management information systems (MIS) research studies from 1981–87, a comprehensive, multidimensional model of IS success was postulated by DeLone and McLean (DeLone and McLean 2003). Shannon and Weaver (Shannon 1948) defined the technical level of communications as the accuracy and efficiency of the communication system that produces information. The semantic level is the success of the information in conveying the intended meaning. The effectiveness level is the effect of the information on the receiver. In the D&M IS Success Model (DeLone and McLean 1992), "systems quality" measures technical success; "information quality" measures semantic success; and "use, user satisfaction, individual impacts," and "organizational impacts" measure effectiveness success.

3 RESEARCH METHODS

3.1 Study design

A qualitative research approach was selected. There are multiple categories for qualitative research, but they all emphasize the meaningful relationships of social phenomena and the need to consider when describing, interpreting, or explaining communication, culture, or social activity. The aim is to understand the significance of human action in professional collaboration. Qualitative research favors humans as data collecting instruments due to human's ability to adapt to various situations during research. This method also allows the voice of the professional to be recorded as such. The final reason for choosing a qualitative approach allows the research plan to form and change during the research process itself (Hirsjärvi et al., 2018).

Different aspects of collaboration have been studied, but little or no research was available on the specific research area where collaboration was studied to understand the impacts to success or failure to ISD. It was essential to approach the information gathering as a holistic process when collecting the data in situ and better understand the revealed topics. The field of study is also such that although quantitative data is available, this data does not adequately offer possibilities to explore the nature of the collaboration (Järvinen and Järvinen 1997; Hirsjärvi et al. 2018).

3.2 Data collection

In order to get a deep understanding of how professionals act in the roles of designer and developer, a set of individual interviews were conducted. The advantages of the interview method are many. It is a relatively flexible data collection method and can adapt to the informants' various situations and individual styles. The interview also allows for digging deeper into topics. Narratives from two distinct yet collaborative roles allow for a comparison of the ISD project from their perspectives. The interview also gives a direct voice to the informant – it is his or her subjective opinion that matters (Hirsjärvi et al., 2018).

Often the advantages of the interviews are also their disadvantages. Interviews need to be planned very carefully, and adequate time needs to be set aside for the interview session. The questions need to be designed to minimize the informant's tendency to give socially suitable answers. Informants may also be intimidated by the situations and feel that their professional competence or knowledge is being tested (Hirsjärvi et al. 2018). Several techniques can alleviate this. The participation of the informants was voluntary. It limited the number of informants available for the study but eliminated persons who had reservations about the research motivation. Another technique used to alleviate the concerns was to spend some time at the beginning of the interview sessions to go over the research goals and explain that they are not the focus of the interviews as persons but their ways of working and collaborating with them the other roles.

In a semi-structured interview, some of the questions or themes are selected before conducting the interviews (Järvinen and Järvinen 1997). The structured interview is selected when seeking statistical generalization. The open-ended interview is used when theoretical generalization is needed (Järvinen and Järvinen 1997).

A semi-structured interview method was selected due to the above-mentioned reasons. It combines specific questions with open-ended questions, allowing unexpected insights to emerge from the discussions with the informants (Seaman 1999; Hove and Anda 2005). In a semi-structured interview, the aim is to gather information broadly from a set of focused topics (Järvinen and Järvinen 1997). With this, a list of themes or questions was drafted that was used as a guide in the interview sessions and discussions with the informants. Few key questions were identified as critical in order to be able to collect a balanced and comparable data set. The semi-structured questions are listed in Appendix 1. Open-ended interview questionnaire template.

The group interview is an efficient method for collecting information from several informants at once. The group may fill in data that an informant might otherwise ignore or overlook. As a downside, the group members may influence the group members in ways that the researcher cannot control or foresee. They may leave out topics or issues that are not socially acceptable to discuss (Hirsjärvi et al., 2018). In this thesis, the group interview was not selected as the data collection method for that reason. The intent was to get the informant's personal stories collected without the influence of peers. However, it would have been interesting to have another session as pair of interviews with the same semi-structured question set. Unfortunately, this was not possible due to the schedule constraints of the informants.

In addition to the interviews, in-situ observation was employed to gather information about the context of the collaboration. Observation is a delicate way to gather information (Hirsjärvi et al., 2018). It is regarded as the necessary means to collect information but is often replaced by questionnaires and interviews as it is a pretty labor-intensive method

(Hirsjärvi et al., 2018). Observation has also received a critique that it would even alter the behavior of the people being observed (Hirsjärvi et al., 2018). Finally, for the information collected through observation, there would need to be a second independent observer with whom notes could be compared (Hirsjärvi et al., 2018). For master thesis purposes, this would have been overkill (Hirsjärvi et al., 2018).

For this thesis, observation was used in addition to the interview method. Unfortunately, permission was not given to observe teams of developers and designers working on an actual project. This would have given more information and allowed for deeper analysis of the working collaboration in real situations.

The observation took place in three different office spaces before and after the interview sessions. Notes were taken that describe the workspaces in detail. Work arrangement was researched further during the in-person interviews by asking questions about the work environment. This was done to get a deeper understanding of why the workspace was set up and how it serves the work that both the developer and the designer do.

3.3 Categorization

The in-person interview is a labor-intensive data collection method. Even though the actual interview sessions can be relatively short, the planning and preparations take a few days to a few weeks. Time is spent to get all interviews agreed and scheduled, and location arranged. Interviews took between 92 to 132 minutes. Transcribing the audio recordings took 60 minutes for every 15 minutes of audio recording, with approximately 160 hours in total.

The transcripts were fed into the NVivo system that is a commonly used qualitative data analysis tool. Transcripts were coded into nodes of information (themes). A set of themes was created based on the semi-structured interview questions in "Appendix 1. Open-ended interview questionnaire template". The set of themes was expanded during the analysis stage as new themes arose from the data. The codes were created to represent the common themes that arose from the narratives. A general quote theme was used to highlight aspects that were mentioned during the interviews.

3.4 Ethical issues

The ethical principles of research with human participants published by the Finnish National Board on Research Integrity (Tutkimuseettinen neuvottelukunta) (Iina Kohonen, Arja Kuula-Luumi 2019) were followed in the design of this research.

Interview subjects were chosen from volunteers from two Finnish software development companies. A companywide message requesting for volunteers for interviews in both roles was sent out. The goal was to get at least four designers and four developers to volunteer for the interviews. Eventually, four designers and four developers were selected, and interviews were scheduled with each one separately. Later one developer canceled the interview session due to a scheduling conflict.

At the beginning of the interview, the informant was informed about the purpose and goals of the research, as suggested by Seaman (1999). The fact was stressed that the focus of the research was not the individual informant or his/her behavior but his/her collaboration with the designer/developer role. Each of the seven interviews were recorded with the permission of the informants. The recordings were transcript into separate documents. These documents were stored together with the audio recordings to be used for analysis, comparison, and interpretation. A chance to review the interview transcript was given to each informant. Minor alterations were made to the transcripts as a result of the review comments. In one case, there was a need to refer to the audio recording of the interview with the informant to make sure his words were valid as transcribed.

In the reporting, attention has been paid in order to keep the informant's identities a secret. Direct quotes, when used, have been paraphrased to hide any idiosyncrasies a particular informant might be recognized. Each informant has been given a label in the form [designer | developer | <number>] that does not represent their interview order or any other recognizable fact. In the final reporting section, the informants were referred simply by their role, either a designer or a developer.

4 REPORTING (RESEARCH RESULTS)

The structured questions formed the basis for discussion, and the research results were structured accordingly.

The first set of questions were about the background of the informants: the age, education, and how did they end up in that role. The informants were asked to define their role and name it – this formed a basis for comparison between the informants on how different their roles were in detail and what titles were used.

Informants were asked to describe their work environment and the roles of their colleagues they work with. Then the discussion was guided towards the collaboration aspects. All informants were asked whether their collaboration was prescribed or voluntary. This question was designed to check their worldview – whether they see themselves as free-thinking individual professionals or as part of bigger machinery.

A sizable portion of the interview sessions was spent discussing how the informants work – what methods (if any) are being employed, what tools they use to accomplish their work, who their stakeholders are, and how they work with them.

The inputs and outputs were also discussed – what artifacts did they create, which of them were shared, to whom and why. Informants were also asked to place their activities in a linear timeline for the ISD project to find out where they see themselves in that timeline. The informants were asked to tell what they do not do in the context of an ISD project.

In the following, the research results are reported from the interviews. The workplaces are also described in detail.

4.1 Demographics of the designers

The designer informants have a wide variety of professional backgrounds. Educational background ranges from M.Sc. in engineering, B.Sc. in business administration, and being self-educated without any formal education in the field. By contrast, the Design Census survey ("Design Census" 2019) reports that most designers have a bachelor's degree, and only 6% have a master's degree. Category in the Design Census ("Design Census" 2019) "I was born this way" fits into one designer informant who reported to be self-taught. A designer reports:

"In my previous job there were no designers - developers did the design as they pleased. At some point I realized that although I loved to code, I loved designing things even more. I've always been interested in design and read a lot about design before I got a job as a designer. Eventually I got hired as a designer even though I did not have any hands-on experience on the matter." – a designer

The ages of the informants fit into two brackets – 30-40 and 40-50. For comparison, the Design Census ("Design Census" 2019) reports the average age of designers to be 35 years.

All informants had grown to their roles through experience ranging from 6–15 years. All of them had worked professionally in other roles within IS development. Previous roles included a developer, an art director, a marketer, a service designer, and a software test engineer.

4.2 Demographics of the developers

Developer informants few had formal education and had earned degrees in software engineering or computer science (both M.Sc. and B.Sc.). By contract, Stackoverflow's 2021 developer survey found that 80% of the professional developers hold a formal higher education degree, with B.Sc. being the most common among the survey participants (Stackoverflow 2021). A developer reports:

"I was dabbling with C64 for as a kid and did some "Hello World" experiments. Then my father showed me a hot-air balloon moving across the screen, sprite, and I thought that it was so cool thing that I need to learn how to do those things." – a developer

The ages of the informants fit into two brackets – between 25-34 years old and between 35-44 years old. The StackOverflow[1] 2021 survey reports that majority (69%) of the professional developers are between 25 to 34-year-old (Stackoverflow 2021).

[1] StackOverflow developer survey is a yearly global survey of professional developers and is highly cited as the authoritative source of data on the global developer community ((Stackoverflow 2021)

The informants have worked full-time as developers between 9–15 years. More than 50% of the Stackoverflow 2021 survey participants have been working between 11 and 17 years by comparison (Stackoverflow 2021).

4.3 Role of the designer

The role of a designer role is ambiguous or multifaceted. The informants used names such as "application designer", "digital designer", "UX designer", "UI designer". Names such as "visual designer", "interaction designer", "graphical artist," and "concept designer" are also specifiers that are often added to the role of a designer.

> *"This is slightly complicated - in bigger companies there often is separate and clear-cut roles such as business analyst who deals with requirements, UI designer, front-end developer and such. Here at A these roles are somewhat folded into the role of Application Designer." – a designer*

A designer informant hinted that the size of a company has something to do with the range of responsibilities given to the different roles. "Visual designer" designation hints that the person in that role designs something visual – how things look like. "UX designer" or user experience (UX) designer might deal more broadly with the experience, considering not just how things look but also what kinds of interaction there is. Some of the designations are just titles used to communicate to stakeholders and set up expectations.

The designers mentioned several activities they perform while holding the role of a designer. The designer role is genuinely multifaceted, even though not all designers do everything mentioned below in all ISD projects. Many of them are something that one would expect – to do with designing artifacts. Some can be a little surprising such as making lists. Lists have nothing to do with visuals or User Interfaces. It is one technique employed by a designer to perform the "sense-making" activity mentioned below. The design is not just drawing mockups or pixel-perfect views. Many of the activities designer role encapsulates today used to belong to other roles such as system analyst, business analyst, technical writer, etc.

In the table, the activity names and the descriptions are from the informants. The author augments some descriptions in order to fill the gaps in the transcripts.

Table 1 - Activities that designers perform

Activity name	Brief description
Requirements	Gathering and documenting the requirements from the customer or from the end users
Specifications	Writing specifications for the system
Designing workflows (of a system)	Part of the specifications, specific workflows of the user activity with the computer system or the information system on the whole
ER-charts	Entity-relationship charts that depict the various entities of a system and their relationships
Lists	Listing things out, a planning technique or a design technique
Views	Designing the views of a UI
Wireframes	Creating a conceptual “map” of the views and how they fit together, how the activities flow from one view to another
A theme	Theme or a style of a user interface of a information system
A style	Like theme, may include different level of abstraction
Domain model	The model of a information system domain that defines the boundaries of the system
Business analysis	Doing analysis of the business, part of the requirements gathering and analysis
Concept design	A mockup of an IS to be designed, on higher level, mainly to test out assumptions and to allow the designers and developers to learn about the intended use. Could also be a longer document that describes the IS to be created in it’s entirety on a high level that may include some of the critical functionalities. This often includes also the activites of humans interacting with the computer system.
Sales support	Expert advice to the sales team during the bidding process
Estimating	Creating work estimates for various design activities
Sense-making	Iterative process where a designer aims to understand the “whole” to be designed and the reasons and assumptions around it.
“Designing”	A designer referred this as the “actual design work” but includes many above and below mentioned actitivies.
Prototypes	Creating a prototype of an information system for purposes of validation of the design or to map out different design alternatives
Layouts	Part of the UI view design where the view is broken up into layout pieces
Customer relationship management	Joint activity within IS project to deal with various customer-initiated activities such as frequent communication, reviews and decisions
UI design	The design of how the user interface of an IS looks like

Activity name	Brief description
UX design	The design of how the user interface of an IS looks & feels like
Project Management	Informants mentioned that sometimes they have to do project management in addition to their “design” role
Planning	Part of the planning process for an ISD project
Decisions	Doing decisions with various stakeholders on the designs.
Information gathering	Part of the business analysis process
Sketching	Doing draft designs to map out the problem area
Designing the flow of use	Similar to the view design but for a whole flow of the UI
Reviews	Design reviews of various sorts
System testing	Final testing of the working IS in order to review or accept the implemented designs

4.4 Role of the developer

Traditionally the developer is the one who writes the software, codes. During the interviews, this common assumption was confirmed. Developers are primarily interested in turning the requirements and abstract concepts into working software through code written in a programming language.

Developers also deal with aspects of the software architecture. It is the case especially with developers who can be categorized as senior developers. The software architecture deals with functional partitioning and the organization of code blocks so that the overall system can be managed and altered as needed. Developers are interested in the accuracy of both function and data.

Developers also take part in the overall project management. Developers in Company A were usually in charge of task management and defect management. Developers also work with sales to do project estimation and costing.

Table 2 - Activities that developers perform

Activity name	Brief description
Coding	Writing the code of the system in a computer programming language
Testing	Various testing categories were mentioned: unit, integration, UI, system and acceptance testing. A lot of the testing involves writing the tests in the code and some developers considered testing as part of the coding (unit tests).

Activity name	Brief description
Backend coding	The implementation of the functionality needed for the data creation, reading, updating and deleting. Some backends also do other activiries than basic data storage.
Front-end coding	The implementation of the functionality needed for the user interface, the views, the styles. The “look & feel” of a system.
Full-stack coding	Combination of the front-end and backend implementation, some developers defined themselves as either front-end or full-stack.
Architecture design	The arrangement of the technical implementation pieces, separating the front-end functionality from backend functionality.
Project management	Almost all developer informants mentioned that they. performed at least some project management activities.
Work breakdown	Splitting of the project into work packages that can be estimated and costed. Part of project costing.
Estimating	Activity related to work breakdowns and costing. Estimation of how many man hours, man days or man months the information system might take to implement. This is used to give a customer a quote of the price.
Customer communication	Similar to the customer relationship management but deals only with communication between the developers and the customer. Oftentimes seen as project management activity, as part of monitoring and control of a project.
Defect management	Sub-activity of quality control and testing – management of defects in the code so that the desired quality level is achieved that the customer expects.
Bug hunting	Part of defect management, finding out errors in the implemented code and fixing them.
Reviews	Activity where a piece of code or a design is reviewed. Code reviews are within the developers, design reviews is either with the designers or with designers and the customer project owner.
Version control	In order to manage the different iterations and work products of the several developers a version control system is used which contains the “outputs” of the both developers and designers. That includes code, graphical assets, configuration data etc.

4.5 Tools that designers use

In order to understand the collaboration context in detail, the informants were asked about with what tools they perform their activities. Further questions were asked whether the tools were adequate for their job. Particular focus was put on the tools that the designers and the developers shared.

Designers use multiple different tools for their work – both for individual work and for collaborating with developers.

From the interviews, some designers used Evernote[2]. For them, This tool was irreplaceable for them as it holds his history of notes from various projects and designs. They used Evernote as a personal tool and did not use its collaboration features to share with others. For sharing with others, they used other tools and copy-pasted the material they wanted to share to applications such as Slack[3] or Google Docs[4].

All designers mentioned a whiteboard. Whiteboard was seen to be an ideal tool for collaborating when everyone was co-located. Drawing, writing, organizing, and reorganizing ideas was common theme that all designers mentioned. One designer also mentioned that he likes to use sticky notes[5] and a whiteboard - drawing around stickies, organizing stickies to reshape the whole to communicate one's thoughts. Both designers and developers used it.

One designer informant had experience with a software-based whiteboard tool, the Miro[6]. Miro allows multiple people to collaborate on a single canvas or a virtual whiteboard. Miro mimics real-world objects such as sticky notes, felt pens, allowing real-world style freeform drawing, sketching, and jotting. It also features typical chart and diagram tools and allows users to create stencils. It also has tools for text entry and editing. The informant mentioned that they usually use Miro both with developers and with customers. It was the tool of choice when stakeholders were not co-located.

The Smartboard[7] was mentioned once. It combines a physical whiteboard with a projector-camera combination, allowing remote teams to collaborate on the "same" whiteboard space. As it requires all participants to have a Smartboard, it was deemed too expensive and restrictive for modern collaboration needs. When it was used, the experience was good for each participant. The resolution of the device did receive critical comments from a few informants, in any case.

All informants used Slack. Slack is an instant messaging platform that separates discussions into chat rooms or "channels". The chat rooms can be private or public. Private chat rooms are invite-only, whereas the public rooms are visible in the channel directory

[2] Evernote, https://www.evernote.com
[3] Slack, https://www.slack.com
[4] Google Docs (now Google Workspace), https://workspace.google.com
[5] PostIt, https://www.post-it.com/3M/en_US/post-it/products/
[6] Miro, https://miro.com/features/
[7] Epson Smartboard, https://epson.com/brightlink-interactive-displays-whiteboard-projectors

and can be joined by anyone within the company. When an ISD project is formed, a corresponding Slack chat room is created for the team, including other stakeholders such as the customer. Many informants mentioned that Slack chat rooms are used even if the team is fully co-located. Here, the discussion may be face-to-face or in Slack, but people often prefer to communicate in the chat room to get the message. Chat rooms allow the room members to view the entire chat history to check their discussions later. Slack is an invaluable tool for teams that are not co-located. Even though Slack offers video conferencing, only a few informants mentioned that they used that feature as they had other tools..

All informants used Google's GSuite (later renamed to Google Workspace) in both companies. It is a productivity suite of applications for everyday office work that is a fully cloud-native platform. It was used to write documents, keep notes, do video conferencing and webinars, and share documents created with other applications. It also hosted the company intranet with Company A.

Figma[8] and InVision[9] are tools for doing collaborative designs concurrently. They are both cloud-native applications. Company A had used InVision previously but was moving all its collaborative designs to Figma. Company B had little experience with Figma but had used InVision extensively over the past few years. With these tools, the designers can create visual designs from wireframes to pixel-perfect mockups. They both allow all views to be drawn out separately, and both can do click-through prototypes of the interaction. Company A designers especially liked Figma's capability to create their component library with full styling parameters that could be re-used with other designs. Figma also allows multiple designers to work on a design simultaneously when dealing with large design works. It also allows for styling parameters to be exported as cascading style sheet code for UI developers. However, the informants did not use this feature due to a lack of full support for all code aspects needed. The designs in InVision or Figma did serve as a tool for the quality assurance phase when the working system was compared against the detailed designs. This practice was mentioned only by designers from Company B.

The trend with visual tools is towards tools that allow team members to collaborate on the artifacts simultaneously. They facilitate prototyping, reviews, styling & theming, sketching, version control, and design asset management, which previously meant that

---

[8] Figma, https://www.figma.com/
[9] InVision, https://www.invisionapp.com/

various tools had to be used and documents generated for intermediary work products. They are inclusive – the designers can work on the designs, and the same tool can be used with the developers for design reviews and other formal collaboration practices. The tools for team communication are just as essential. They facilitate geographically separated teams to collaborate via text, voice, and video. They also lower the cost of the projects as no travel expenses accrue. Some tools combine visual groupwork with instant communication for methods such as brainstorming and workshops. The tools are fast moving towards a stage where the collaboration between the designers and developers is finally becoming seamless.

4.6 Tools that developers use

As with designers, the developers use various tools that they use to perform their activities. Questions were asked whether the tools were adequate for their job, and particular focus was put on the tools shared by the developers and the designers.

Developer tools can be divided into a couple of categories. There are tools for writing code, tools for managing a project, tasks, storing exchanging information, tools for revision control, tools for writing and maintaining documentation, tools for reviewing designs, and tools for communicating with stakeholders.

Most often cited IDE[10] was IntelliJ IDEA[11]. Some developers also used other editors, such as Microsoft Visual Studio Code[12], which is especially popular with front-end developers.

Revision control systems included BitBucket[13], GitHub[14] and GitLab[15]. All these are code repositories for GIT[16] version control system originally developed by Linus Torvalds.

[10] IDE – integrated development environment
[11] IntelliJ IDEA, https://www.jetbrains.com/idea/
[12] Visual Studio Code, https://code.visualstudio.com/
[13] BitBucket, https://bitbucket.org/
[14] GitHub, https://github.com/
[15] GitLab, https://about.gitlab.com/
[16] GIT, https://git-scm.com/

For project management, multiple tools were mentioned. A standard tool for all was Jira[17]. Whiteboard was also mentioned as a tool for managing project information. Company B teams used the Kanban method[18] in most projects and used both whiteboard and Jira to keep track of the project.

Documents were written with Google Docs. Some teams used Confluence[19] to write and maintain documentation of the project. Confluence is more a publishing platform than a text editor system and can organize large number of documents and pages of distinct information.

Developers do not use design tools for creating designs. Instead, they are used to review the designs. Sketch[20], InDesign and Figma were mentioned as tools where developers do pair- or team reviews of the designs.

For stakeholder communication, Google Hangouts[21] and Skype[22] were mentioned. Both facilitate video & audio calls with good video and audio quality. Some developers also mentioned Slack as a video calling and mentioned that it is mainly used for chat room purposes. Skype was also used as an instant messaging platform, but Slack in both companies largely replaced it.

The trend with the developer tools is also towards facilitating collaboration between the various ISD project roles. The tools facilitate pair or even group coding sessions, allow for project management integration, and are easy to use even for the designers. This defect has created an unnecessary division between these two roles. Naturally, there are still tools that are needed only by the developers. In summary, the toolset needed to implement an IS successfully converges towards a few good tools for collaboration. The ecosystem of tools seems to expand without constantly breaking the good collaboration practices.

4.7 Collaboration spaces

Factors that may have an impact on collaboration performance in a workplace include lighting, temperature, air quality, and acoustics (Choi and Moon 2017; Wagner et al. 2007; Abbaszadeh et al. 2006; Kosonen et al. 2011; Fisk 2002). Therefore, it is relevant

---

[17] Jira, https://www.atlassian.com/software/jira
[18] Kanban, https://en.wikipedia.org/wiki/Kanban_(development)
[19] Confluence, https://www.atlassian.com/software/confluence
[20] Sketch, https://www.sketch.com/
[21] Hangouts, https://hangouts.google.com/
[22] Skype, https://skype.com/

to consider the workplaces as part of this research. In order to find out what, if any, impact the spaces had for designer – developer collaboration three spaces were also observed before the interview sessions.

Three workspaces in two different buildings were observed. Space A was in the third floor of a mid-20th century industrial building converted to open space for office use in 2000. Space B was in the same building as Space A but on the second floor. Space C was in a different building, on the top floor of a mid-20th century office building in a busy downtown area.

4.7.1 Space A

Space A is situated on the third floor of the building. The space where designers and developers work is an ample open space with structural columns to support the roof and desk arrangements used as space dividers. Desks are arranged side-by-side in three desks, with another set of three desks facing the first row. There were nine sets of these grids of 6 workstations in the room, allowing up to 30 people to be working in the space.

The open space has tall windows facing the street and adequate lighting and ventilation systems. The temperature is controlled by an air conditioning system. The temperature of the space was mentioned to be a bit too cold during the summer but comfortable during fall, winter, and early spring. Parts of the space were reported to be drafty as the air vents were close to the desks.

In addition to the open space, there are also three conference rooms and two "phone-booth"-style soundproof cubicles. The conference room setup included a conference table, six to eight chairs, a 60" TV for presentations, a conference speaker with an inbuilt 360° video camera, an entire wall size whiteboard. Each of the conference rooms has tall windows facing the same street as the open space. Doors have large windows so that people can see in from the corridor. Space A at site 1 is visualized in Figure 3 – Space A.

For intense collaboration, this space was said to be a bit poor. As it is an open space shared by tens of people, a loud conversation would disturb. There are three meeting rooms for intense collaboration, but they are not exclusively available for the 3rd-floor workers. Frequently designers and developers need to limit their chat to digital channels. This can also be an asset if the ISD project team is geographically distributed. In that case, all the team communication should be kept in the digital channel. A good feature of this space is that desks are arranged to form a sense of team around the tables and use non-verbal communication cues to convey a meaning that is not possible entirely through digital channels.

4.7.2 Space B

Space B is located on the second floor of the building. The team workspaces are in three separate rooms, and all are set up differently according to the team's preferences that occupy the space. Space B also has a big conference room to host companywide meetings, seminars, and presentations requiring ample space. Space B hosts a fully equipped kitchen. The space also includes a WC and space for storing jackets and outdoor shoes.

The team rooms all have tall windows facing the parking lot with adequate lighting and ventilation systems. An air conditioning system controls the temperature. Similarly, the temperature of the space was mentioned to be a bit too cold during the summer but comfortable during fall, winter, and early spring. No one mentioned the air vents causing a draft as they did in Space A.

The workstations are set up side by side manner along the walls of the workspaces. Separate space for each team allows the team to have meetings and remote video conferences in that space without reserving a meeting room. This creates certain flexibility for timing the meetings and allows for ad-hoc conversations with the whole team. In-team communication has little or no obstacles, and communication with stakeholders can be arranged with ease.

4.7.3 Space C

Space C is on a 100-year-old building's top floor that has been converted for office use. The floor is divided into smaller workspaces that are used either as meeting rooms or for teamwork. In the corridor, there are common areas with coffee machines. On one end of the corridor is a large kitchen and lounge area.

Workspaces usually seat six to ten people. This is seen as limiting as seating requirements vary. For this reason, part of the team needs to alternate between being at the office and working from home. Workspaces have two long desks facing each other and seat 4-5 people on each side. Workstations have space for a laptop, 1-3 external displays, a keyboard, and a pointing device (either a mouse or a trackpad). The workspace also has a wall-to-wall whiteboard that is used as a Kanban board and a drawing area for architecture diagrams, UI sketches, or team notes. A big-screen TV (70") can be used as a video conference display. It is also possible to connect it to any workstation to use it instead of the whiteboard.

The workspaces all have windows facing either the street or the courtyard. The lighting fixtures and the windows provide good lighting conditions. The temperature and ventilation in Space C were found to be comfortable.

Workspace has a glass door and a glass wall so that people walking by can see in, and people inside can see who is passing by. In other team spaces, the glass wall was used as a whiteboard. Some rooms had a separate whiteboard on wheels that could be moved around.

4.7.4 Virtual collaboration spaces

With virtual spaces, it meant electronic meeting/chat rooms, e-mail message chains, shared notepads, and electronic whiteboards together with a voice and video connection.

Company A had a long history of utilizing electronic meeting rooms, instant messaging platforms, and e-mail. Skype and later Slack were the chosen tools for companywide and in-team collaboration. E-mail is also available, but it is mostly used as a tool to communicate with external stakeholders.

Company B has similar tools for virtual spaces as company A. The Slack is utilized for companywide and in-team collaboration but also for collaborating with customers and other external stakeholders.

Slack allows for both public and private virtual chat rooms to be set up. By default, all new chat rooms are public, and one-to-one rooms are private. Slack as a platform allows for various notifications to be sent to the people in the chat rooms. One can also post permanent messages to pin them to the room to highlight and stay visible for all people in the chat rooms. One can also send "pings" to a chat room or a set of persons within a chat room. One can also use "mentions" within new messages sent to a chat room by using special @-annotation.

There are multiple other solutions for virtual or electronic chat rooms. In addition to Skype, there are tools such as Flowdock[23], Mattermost[24] and Microsoft Teams,[25] to name

---

[23] Flowdock, https://www.flowdock.com/
[24] Mattermost, https://mattermost.com/
[25] Teams, https://www.microsoft.com/fi-fi/microsoft-teams/group-chat-software

a few. All these facilitate in-team and companywide collaboration with the ability to invite external collaborators into chat rooms.

4.8 Ways to reduce risk of ISD failure – what the informants suggested

Interview narratives provided ideas and good practices on how the risk of ISD failure can be mitigated. It emerged as a common topic that informants saw as an essential aspect of their work. Ideas span from preferred personal practices to team practices to companywide prescribed processes.

Both designers and developers mentioned most if not all mitigation strategies in one form or another. The same was true also between both companies – the differences seem minute. However, at the same time, one needs to be cognizant that subtle differences may yield a significant competitive advantage for the companies, which in turn is an advantage for the customers.

Communication was raised as a crucial aspect to ensure that the ISD project succeeds. Especially in the early phase, the project must be set up correctly. Communication is vital in every aspect of transferring knowledge. One developer informant mentioned that he especially likes when designers annotate their design drawings with notes about actions and flow of the views. Agreeing on what to do and how to do it was an important part of the communication process. Tools such as brainstorming, or workshops were mentioned for that.

"At this phase [initiation] it is constant communication. You need to be truly transparent so that you do not base things on false assumptions" – a designer

Almost all the informants cared deeply about the **collaboration** between their respective roles. Designers should understand at least some aspects of software engineering. This, according to the informant, increases the likelihood that things will work out well when creating an IS.

“We want to create design that are actually implementable”

– a designer

The feedback loop between all the stakeholders should be kept as short as possible. It was mentioned that it changes slightly when moving from project initiation towards implementation. First, the feedback loop needs to be short between the designer and the customer. This ensures that a common shared understanding is formed and trust is established between the stakeholders. Further on in the development lifecycle, the focus shifts from customer-designer feedback to customer-designer-developer, designer-developer, and in the end, towards customer-developer.

“Ideally we’d have a functional prototype that can be tested out [by the customer] and walked through. This would help to reduce the amount of misunderstandings.” – a developer

One informant mentioned that in his projects, they prefer to do daily meetings with the customer initially. If that is not possible, they tend to insist on doing dailies every second day. This is also to keep the feedback loop as short as possible. Highly effective teams can progress quickly, and with that velocity, the risk of going back and doing rework increases exponentially. Having short feedback loop has the added benefit of keeping the customer engaged with the ISD project.

Another informant mentioned that they refuse to accept the project if the customer does not supply **a full-time product owner** to the ISD project. An informant working in the other company mentioned that this would be their preference but also said that they do not insist on it. All the informants said that they needed to know something about the business case and why they were doing it.

"Quite often the business level requirements are not explicitly communicated, and the customer jumps straight into solution space. They skip sharing of the background information which would be really important to know and to understand. After that one just needs to ask about details." – a designer

Team size was mentioned as an essential aspect by both designers and developers. Keeping teams as small as possible allows for less management overhead, fewer handovers, and in a small team, the communication is efficient. Being co-located as a team was mentioned to be one way to increase the success rate.

One developer mentioned mutual respect. He had experienced projects where they lacked a dedicated designer and felt that it was challenging to ensure the customer's quality without a designer.

"There are definitely advantages [to having a designer in the team]. [It means that] someone has actually thought out the usability of the system when you contrast this to the situation where there is just a developer, and the end result pretty much looks like it – [something that is] not very well thought out" – a developer

4.9 Collaboration events

The informants said that the collaboration among the designers and the developers happens most frequently at the beginning of a project and the end of the project. One designer mentioned that it is all collaborative work when bidding [project proposal] to the customer.

At first, the collaboration is about sense-making. The designer-developer tuple needs to create a shared understanding of what they need to work together to satisfy the business case their customer needs.

"Often it starts with me explaining the domain, drawing the domain model to the wall and then we start thinking how to model it with technical models. With this we have had very close collaboration with [the developer]." – a designer

All informants mentioned styling as part of the ISD project where designers and developers need intense collaboration. It was emphasized that it is a deep collaboration to get the styling right.

"When the UI shell starts to look good then a developer can just take the project add it to the rest of the technical project" – a designer

A designer mentioned that the intense collaboration fades away and becomes as-needed cooperation once the raw coding starts. Another designer mentioned that the designers activate again when the coding is nearly done, and they can start testing the final product. At this stage, intense collaboration happens that ensures the quality the customer is expecting.

Sometimes the collaboration is also about agreeing to disagree. A developer tells:

"When you are in early talks with the customer and the designers you oftentimes get to 'kill' ideas even before they are fully designed as they would be costly to implement" – a developer

In one aspect, the designer informants and the developer informants differed significantly in their narratives. The designers felt that the collaboration happens almost throughout the project, whereas the developers stated that the collaboration starts at a pretty late phase. Developers mentioned that they do not always have anything to comment on, especially when working with designers that can design so that there are no issues with the implementation.

"I start communication with the designers when I get first designs for review" – a developer

All informants said that they enjoy the collaboration and the "challenge" they get from looking things from different angles.

4.10 Collaboration breakdowns

Breakdowns in teamwork are common as teamwork deals with interactions among humans. The informants reported several breakdowns as part of their narratives that dealt with breakdowns in designer–developer collaboration and breakdowns with other roles in the mix.

Breakdowns happen most often due to defects in communication. Defects happen primarily at the semantic level, as most of the communication is sent and received. The most often cited reason for rework due to communication defects was "we were in a hurry".

> *"Developers were in such a hurry that they did not take the time to internalize the comments in the wireframes – either they did not care or they did not understand them." – a designer*

In another comment a developer said that the static designs are just not enough due to different contexts.

> *"Our context [as developers] is different, we have different understanding even if we are all at the same wireframe design. It is just a static picture. It provides few clues how it would look like or how would it need to behave in the end." – a developer*

Sometimes the breakdown occurs not between the designer and the developer but between the customer and the team. Both designers and developers cited this. The customer may have good reasons for change or just react to what they see that induces a change. Frequently as the customer learns more about the system, they change their mind. Also, as mentioned above, the different context for the customer and the team distorts the communication and may lead to breakdown.

"Myself, the designer and the customer can have long conversations [together] using the same words and look at the same designs but in the end we may discover that we understand it [all] differently and it did not get done as it should have been done. This leads to rework."
– a developer

Being co-located gives no guarantees of all going well. Even if the team, the customer, and other stakeholders sit together for the ISD project, various forms of defects or breakdowns in collaboration happen. A designer told a story about a project that was a constant risk of failing:

"This all happened despite everyone sitting in a same room."

– a designer

Having team distributed in an unbalanced way leads into different kinds of breakdowns.

"That [ISD] project we had one developer in [site B and rest were sitting together at site T]. The communication went bad because I think the developer was quite junior and he did not know how to ask for help. A lot of time was spent and the developer was doing something on his own weeks at a time." – a designer

Scheduling the design and the development too far apart can also cause breakdowns quite late in the ISD project. Paasivaara & Lassenius mention explicitly that synchronization of the main milestones together with frequent deliveries are essential for efficient collaboration (Paasivaara and Lassenius 2003).

"They implemented their part in a way that was not as it was originally intended. This made it impossible for [the designers] to style the [user interface]. [The designers] had to throw the ball back to the devs with instructions to modify the [existing implementation] such that made styling possible. This led into a tremendous amount of re-work and back-and-forth communication on fixing things that were already implemented" – a designer

Finally, human nature was mentioned as a possible cause for breakdowns in the collaboration. Both the designers and the developers mentioned and admitted this. Below is a quote from a designer:

"I feel that the designers have big egos, to the level it can be a problem. Some need to perform in front of others. I am now not sure if this [is] just about designers or is it generally a human [personality] trait. I've not seen that behavior with developers, but I am sure there are [such traits] with developers as well. "– a designer

5 DISCUSSION

DeLone & McLean's MIS Success Measures are divided into six categories: system quality, information quality, information use, user satisfaction, individual impact, and organizational impact (DeLone and McLean 1992). DeLone & McLean enumerated the success factors (see Figure 5 - MIS Success Factors). In the following, the informant narratives are reflected against the above-mentioned categories.

5.1 Impact on System Quality

The designer – developer collaboration can impact data accuracy, ease of use, ease of learning, the convenience of access, human factors, the realization of user requirements, system accuracy, system efficiency, response time, and turnaround time.

Ease of use is something that is usually a responsibility of a designer. Based on the narratives, this proved to be true to some extent, but developers said they want to use their skills to ensure that the overall software architecture makes it cost-effective to create the software in a way that the design specifies. Designers focus on measures such as the convenience of access, human factors, and ease of learning. Developers can impact data accuracy, system accuracy, system efficiency, and performance measures like response time and turnaround time.

Narratives indicate that the overall system quality is created together through designer – developer collaboration in the development.

5.2 Impact on Information Quality

The designer – developer collaboration can impact usableness, understandability, clarity, format, appearance, content, accuracy, conciseness and completeness.

Both designers and developers said that they both participate on measures such as clarity, format, conciseness, and appearance. Narratives revealed that especially styling the system is often a challenging part of the system development. Collaboration is needed to ensure that 1) it can be done 2) it can be done right.

Narratives revealed that designers have a bigger impact on measures such as usableness, understandability, and content. These measures are closely related to system quality measures. Designers often obsess over the details that developers overlook. Designers aim to communicate the importance of these details to the developers as leaving them to a later phase may require rework in the code level. Designers are also more involved in the early discussions with the customer to discover the “thing” to be built.

Based on developer narratives, the information quality measures that have the most impact are accuracy and completeness. Again, software architecture design is the key to these.

5.3 Impact on Information Use & User Satisfaction

The designer – developer collaboration can impact overall satisfaction, information satisfaction, enjoyment, and decision-making satisfaction.

Information Use and User Satisfaction (see Figure 3) follow the previous success categories of System Quality and Information Quality. The narratives indicate that overall satisfaction is a goal of both roles and that their collaborative work aims to succeed in all measures..

5.4 Impact on Individual & Organizational Impact

The designer – developer collaboration can impact individual measures such as learning, information awareness, information recall, problem identification, decision effectiveness, improved productivity, task performance, quality in plans, and personal valuation of IS.

The designer – developer collaboration can impact organizational measures such as operating cost reductions, overall productivity gains, increased work volume, product quality, contribution to achieving goals, and service effectiveness.

In the end, the business case of an ISD project rests on realizing the impact on the individual and organizational levels. Designers said they are heavily involved in ISD projects and are introduced early on to the business goals. Some informants with Company B indicated that they perform the acceptance testing based on goals for the project.

5.5 Other ways to ensure ISD project success

In his seminal article “No Silver Bullet”, Brooks (Brooks 1986) states that there is no silver bullet to slay the werewolf (the software complexity). Many technologies and methodologies mentioned in the “NBS” article have become standard in most ISD projects.

Meanwhile, the IS industry is full of salespeople that claim to hold a silver bullet and one needs only to adopt their pre-packaged product or methodology. Yet Brooks argues that:

1. "The difficulties of making big software systems consist of essential difficulties and accidental (or incidental) difficulties,
2. The great leaps of progress in the past were accomplished by eliminating accidental difficulties, for example by high-level languages, time-sharing and workstations,
3. Of the remaining difficulties, at least half seem to be essential, the very inherent complexity of what we build,
4. Therefore, no attack on accidental difficulties can bring an order-of-magnitude improvement – indeed, more than a factor of 2,
5. And yet, most of the proposed "radical improvements" proposed continue to address only accidental difficulties." (Fraser et al. 2007)

In the retrospective to the "NSB" (Fraser et al. 2007), researchers looked back to the original postulates listed above. Fowler (Fraser et al. 2007) agrees mostly that the silver bullet is nowhere to be found. However, some advances in the craft, such as iterative development, object orientation, and pre-built software packages in packages or libraries, have taken the industry further. He states that no single technology or methodology can make the fundamental difficulty of writing code go away (Fraser et al., 2007). Interestingly the narratives from the interviews do not raise the difficulty of writing code at all. The difficulties reported by the developers revolve around making the right architectural decisions early in the project. This requires more than just good coding skills. As stated in one narrative, there needs to be constant communication between the designer and the developer at the beginning of the ISD project.

Lopez ((Fraser et al. 2007) disagrees with Brooks (Brooks 1986). He claims that there exists a silver bullet. He states that the pursuit of personal and professional excellence, when achieved, results in order of magnitude improvements in software productivity (Fraser et al., 2007). This view is implicitly present in the narratives. All informants take pride in their chosen career path and aim to excel in it. They emphasize not only their skills within a role but also their skills to work together – to collaborate.

Namioka (Fraser et al. 2007) highlights that creating products that do not meet expectations is the biggest problem in the industry. Writing requirements, doing quality assurance, and having product and project managers in the ISD project are all instances of futile attempts to represent the customer's interests (Fraser et al., 2007).

Lack of on-site customer in the ISD can be mitigated but not necessarily fully (Fraser et al. 2007). The informants mentioned several times the importance of having clear communication lines with the customer. There are several ways to achieve this. The most often cited method was to co-locate the ISD project team in customer premises. This was mentioned as a rule by informants in Company B. Other means mentioned were daily meetings with the development team and the customer. This can work, but special attention needs to be paid to the customer representative. He/she must be able to make decisions for the project or have a straightforward and fast process to make the decision.

Essential difficulties with the ISD remain (Brooks 1986). One aspect common with the informants was that they all recognized the need for both roles – the designer and the developer. An interdisciplinary perspective is needed (Fraser et al. 2007) and a healthy dose of mutual respect. Working closely together, spending time to make sure that all stakeholders within and without the team have a shared common understanding of the artifact to be built is no silver bullet but a necessary “lead bullet” that, when followed, could yield predictable positive results.

Parnas (Fraser et al. 2007) proposes a solution out of the software “crisis”. That is to establish a level of formalism to ISD common with civil engineering projects such as bridge-building. This approach would put a barrier to entry into the ISD profession – only people with proper credentials would be allowed to do the design and development work. This, together with regulatory supervision and audit, would weed out the people who are not up to the job (Fraser et al., 2007). His approach seems a bit harsh at a time when the barrier of entry is perhaps the lowest it has ever been. Only three out of seven professionals interviewed for this thesis would have had a chance to practice if Parnas’ postulates were effective. Even if we would not accept his way as the only road forward, we should embrace a constant improvement mindset and reject the “silver bullets” as they are offered.

Even though each informant from both roles described their ways of working as not prescribed, there still was evidence of specific techniques that usually are associated with prescribed process frameworks or management prescribed rules. Further, the role of the management was present implicitly. Both companies had specific rules on how to approach ISD projects, yet 6 out of 7 informants said that they are fully self-organizing and that management is there to coach and support them generally. Narratives tell a story where contemporary ISD teams do not use prescribed processes and methodologies. This is consistent with what Avison & Fitzgerald have stated earlier (Avison and Fitzgerald

2003). Even though some informants did imply certain companywide processes are in place, there seems to be a freedom for ISD teams to adopt and adapt the ways of working that work for them in the context of a particular ISD project.

The role of the customer emerged as a critical one. Implicitly this has been discovered from previous research. The customer is not always entirely on board with the ISD project and only involves contract negotiation. This can be tackled with a working practice that investigates the customer's capability and willingness to commit to the ISD project before further work is done. Ideally, this approach reduces the risk of early ISD project failure significantly. It was suggested as good practice to keep the feedback loop with the customer as short as possible. Designers and developers said that it is crucial to understand the big picture – what is the business case the customer is after. Unfortunately, an ISD project is not a static entity. The ISD project occupies the same space and time as the world around it. Therefore, there needs to be a set of capabilities within each ISD project that allow the project to respond to its surrounding world. The customer of the development team has a business case, but significantly the designer's role can add even more value to the process if he/she has access to the user. This often means that SDLC is extended beyond the formal requirements specification so that the requirements are jointly developed with the customer or the business case owner. Several informants mentioned this practice.

In designer-developer collaboration, it is beneficial to do “frequent deliveries” in the form of artifacts produced by either the designer or the developer (Paasivaara and Lassenius 2003). This is similar to the idea in Scrum of Inspect and Adapt (Sutherland and Schwaber 2020). Scrum adds another idea to this: transparency (Sutherland and Schwaber 2020). All these three are based on empiricism – the notion that only through a continuous cycle of build-inspect-adapt can collaborate effectively, and assumptions are kept at a minimum between all collaboration parties.

Are we back in the pre-methodology era (Avison and Fitzgerald 2003)? The short answer to this would be no, we are not. The plethora of methodologies, tools and prescribed processes is part of IS development's current state. The narratives tell the story that they have specific mental frameworks and rules in place when they begin. Those give them a starting point and a structure to do a successful collaboration within the team. This is true despite their inability to verbalize and explain in detail how they work.

The idea of contingency is implicitly present in the narratives. It seems that is the basis for their ways of working – adapting to a situation at hand. Implicitly the management of both companies recognizes this as well. They certainly do have a set of processes and rules that are written down and communicated as part of the induction of new professionals, but they are not imposed as is to the teams working on an ISD project.

5.6 Communication within collaboration

All informants stated that the collaboration of the designer and the developer was essential for the success of the ISD project. The collaboration itself is a combination of communication and working together on the IS artifact to be developed. Shannon & Weaver (1948) split the communication into three facets: technical communication, semantic communication, and communication effectiveness.

The technical level of communication deals with accuracy and efficiency. Based on the narratives the communication between the designers and the developers, there are several obstacles that may hinder the accuracy of the communication.

The first obstacle that was discussed was related to the ways of working and terminology. The new members of the team need to understand the “lingo” of the team before they can work efficiently as a member of a team regardless of the role. The way of working is another communication hindrance. Most designers thought that the ways of working were not prescribed. The ways of working were agreed upon within the team. The narratives of the designers are slightly conflicting with the narratives of the developers. Developers had a relatively strong sense of having a prescribed process to run an ISD project. The narratives on how to involve the customer further revealed that there are commonly agreed rules on how to handle certain aspects of the ISD project.

Another obstacle that was discussed had to do with the team was working location. Based on the narratives, the best way to ensure the technical level of communication is to have the team co-located in the same space, preferably in the same enclosed space. Also, efficient and accurate communication was possible in Space A that was part of the bigger open space but separated sections per project team. Considering the overall communication accuracy and efficiency, the best option was to be co-located in customer premises where the customer was also present and available for the development team. This was standard practice with Company B. Company A also successfully used this setup with businesswise feasible projects. Whether to have the ability to be co-located with the customer is again heavily dependent on the customer's capability, and level of their understanding of how being co-located lowers the risk of ISD project failure.

The worst setup for team communication was mentioned to be a situation where most of the team was co-located, and one team member was remote. The efficiency of the communication of the team was heavily impacted as many aspects of the project were discussed only between the co-located team members. Despite the team agreement that all team communication should happen within a standard Slack tool that provides a dedicated chat room and document sharing. It was suggested that a fully remote team works better than partially co-located just for this reason.

Knowledge creation is an essential part of the designer and developer collaboration. Nonaka (1994) proposed a paradigm for the knowledge creation process where knowledge creation is a continuous dialogue between two or more professionals – turning tacit into explicit knowledge. This is in line with what has been written about collaboration on page 15.

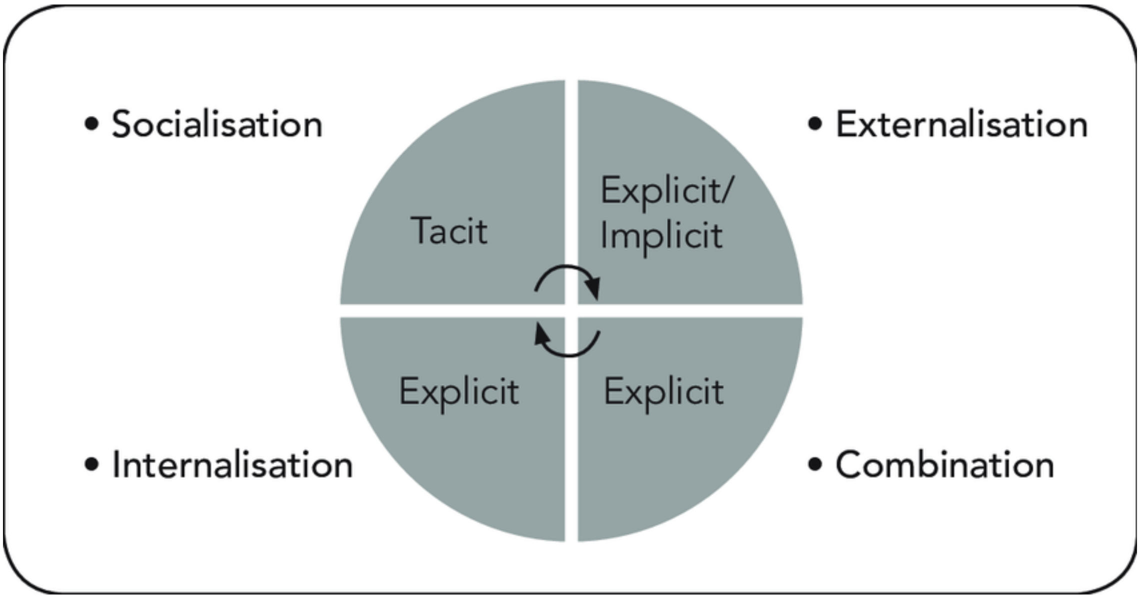


Figure 4 - The SECI Model of Knowledge Creation (Nonaka 1994)

6 CONCLUSIONS

This thesis adds to the narrative–interpretative approach where the emphasis is put on sense-making and finding possible root causes for specific categories of ISD failures. ISD distress concept was introduced during literary review and utilized in the analysis of the interviews. Designer–developer collaboration proved to be a good focus for the research.

There is not a silver bullet that would remove all ISD risks and guarantee success. Some practices are not all that common and re-iterates the importance of those followed mainly by the industry. Essential difficulties remain until the role of the customer is adequately understood. At the same time, both the designers and the developers need to step further out of their comfort zones. The designers need to go even further into the business domains and develop working practices that add the customer to the collaboration process. The developers need to step further out of the technical and coding tasks and to develop ways to make sure that their skills can be best applied to solving the right problem that the customer has. Blurring the role boundaries may be needed.

It is interesting to recognize how the narratives revealed somewhat contradictive information on the interaction between the designer and the developer and the team and the customer. Informants stated clearly several times that they do not have a prescribed process that tells how they should perform their work and that agreement on the work practices was made within the project team. Then again, they acknowledged that they follow specific rules on how the interaction with the customer and other stakeholders happens. When one thinks of the ISD project lifecycle, there is a stage where marketing and sales "own" the process and a handover from sales to development at a certain point within the overall lifecycle. The sales process implicitly impacts the ISD project and how the ISD team will define its internal process and rules. The prescribed process does exist, but perhaps due to the seniority of the informants, they did not acknowledge it. Nonetheless, they do work based on prescribed processes and rules. The ability to decide the ways of working within the team does give the team members a sense of mastery.

The details such as team setup, including location, tools, and ways of working, may seem too minor details to learn anything from or to use to reduce the risk of ISD failure. It is quite the opposite. The details provide a cornerstone for success as these can be considered in the project setup phase. They can also be considered during the lifecycle of a project when adaptations need to be made to these good practices.

ISD team works best when co-located with a customer. The most significant impact of this is in the communication within the team and with the most important stakeholder, the customer. The technical quality of communication can be assured with ease. The defects in the communication process can be readily noticed and corrected. Ways of working can be adapted to suit both the ISD team and the customer. Collaboration between the designers and the developers is most efficient when co-located. Collaboration between the customer (a product owner or project manager) and the designer works best when co-located.

As stated in 1.1, many methods, methodologies, tools, and techniques aim to lower the risk of failure. Not everyone is up to the job. The interviewed developers and designers have chosen their professional focus for good personal reasons. There is a silver bullet, however. Lopez states:

> *"There is a "silver bullet" – it is the pursuit of personal and professional excellence – this when achieved, easily gives us an order of magnitude improvement in software productivity. There is no "silver bullet" from without – it must come from within".* (Fraser et al. 2007)

Lopez (Fraser et al., 2007) has a point. Aiming for excellence is always a good goal. In addition, the ISD project should consider managing the customer – not just the communication but also setting the expectations for the customer and allocating resources to manage those expectations as part of the overall project. Further research is needed to uncover the state of customer management and to create best practices for it.

REFERENCES

Abbaszadeh, S, L Zagreus, D Lehrer, and C Huizenga. 2006. "Occupant Satisfaction with Indoor Environmental Quality in Green Buildings." https://escholarship.org/uc/item/9rf7p4bs.

Ackoff, Russell L. 1967. "Management Misinformation Systems." *Management Science* 14 (4): B-147-B-156. https://doi.org/10.1287/mnsc.14.4.b147.

Alter, Steven. 2008. "Defining Information Systems as Work Systems: Implications for the IS Field." *European Journal of Information Systems* 17 (5): 448–69. https://doi.org/10.1057/ejis.2008.37.

Ashforth, Blake. 2000. *Role Transitions in Organizational Life: An Identity-Based Perspective*. Routledge.

Avison, D. E., and G. Fitzgerald. 2006. *Information Systems Development : Methodologies, Techniques & Tools*. Book. Edited by Guy Fitzgerald. 4th ed. London: McGraw-Hill.

Avison, D.E, and G. Fitzgerald. 2003. "Where Now for Development Methodologies?" Communications of the ACM. 2003. https://dl.acm.org/doi/fullHtml/10.1145/602421.602423.

Baghizadeh, Zeinab, Dubravka Cecez-Kecmanovic, and Daniel Schlagwein. 2020. "Review and Critique of the Information Systems Development Project Failure Literature: An Argument for Exploring Information Systems Development Project Distress." *Journal of Information Technology* 35 (2): 123–42. https://doi.org/10.1177/0268396219832010.

Biddle, Bruce J, and Edwin J Thomas. 1966. "Role Theory: Concepts and Research."

Bostrom, Robert P. 1980. "Role Conflict and Ambiguity," 88–115. https://doi.org/10.1145/800085.802734.

Brooks, Frederik P. 1986. "No Silver Bullet - Essence and Accidents of Software Engineering." In *Information Processing '86: Proceedings of the IFIP 10th World Computer Congress*, edited by Hans-Jürgen Kugler, 10–19. Elsevier B.V.

Brooks Jr, Frederick P. 1995. *The Mythical Man-Month: Essays on Software Engineering*. Pearson Education.

Buckingham, Richard A, Rudy A Hirschheim, Frank F Land, and Colin J Tully. 1986. *Information Systems Education: Recommendations and Implementation*. Cambridge University Press.

Campos, Pedro, and Nuno Jardin Nunes. 2007. "Practitioner Tools and Workstyles for User-Interface Design." *IEEE Software* 24 (1): 73–80. https://doi.org/10.1109/MS.2007.24.

Charette, Robert N. 2009. "This Car Runs on Code." *IEEE Spectrum* 46 (3): 3. http://spectrum.ieee.org/green-tech/advanced-cars/this-car-runs-on-code/0.

Choi, Joon Ho, and Jehyun Moon. 2017. "Impacts of Human and Spatial Factors on User Satisfaction in Office Environments." *Building and Environment* 114 (March): 23–35. https://doi.org/10.1016/J.BUILDENV.2016.12.003.

Curtis, Bill, Herb Krasner, and Neil Iscoe. 1988. “A Field Study of the Software Design Process for Large Systems.” *Communications of the ACM* 31 (11): 1268–87.

DeLone, William H, and Ephraim R McLean. 1992. “Information Systems Success: The Quest for the Dependent Variable.” *Information Systems Research* 3 (1): 60–95.

DeLone, William H., and Ephraim R. McLean. 2003. “The DeLone and McLean Model of Information Systems Success: A Ten-Year Update.” In *Journal of Management Information Systems*, 19:9–30. M.E. Sharpe Inc. https://doi.org/10.1080/07421222.2003.11045748.

“Design Cencus.” 2019. 2019. https://designcensus.org.

Dvorak, Daniel L. 2009. “NASA Study on Flight Software Complexity.” In *AIAA Infotech at Aerospace Conference and Exhibit and AIAA Unmanned...Unlimited Conference*. https://doi.org/10.2514/6.2009-1882.

Fincham, Robin. 2002. “Narratives of Success and Failure in Systems Development.” *British Journal of Management* 13 (1): 1–14. https://doi.org/10.1111/1467-8551.00219.

Fisk, William J. 2002. “How IEQ Affects Health, Productivity.” *ASHRAE Journal* 44 (5). https://www.osti.gov/servlets/purl/821649.

Fraser, Steven D, Frederick P Brooks, Martin Fowler, Ricardo Lopez, Aki Namioka, David L. Parnas, David Thomas, and Linda Northrop. 2007. “‘No Silver Bullet’ Reloaded - A Retrospective on ‘Essence and Accidents of Software Engineering.’” In .

Hirsjärvi, S, Pirkko Remes, Paula Sajavaara, and E Sinivuori. 2018. *Tutki Ja Kirjoita (15.-22. Painos)*. 22nd ed. Tammi.

House, Robert J, and John R Rizzo. 1972. “Role Conflict and Ambiguity as Critical Variables in a Model of Organizational Behavior.” *Organizational Behavior and Human Performance* 7 (3): 467–505.

Hove, Siw Elisabeth, and Bente Anda. 2005. “Experiences from Conducting Semi-Structured Interviews in Empirical Software Engineering Research.” In *Proceedings - International Software Metrics Symposium*, 2005:10–23. https://doi.org/10.1109/METRICS.2005.24.

Iina Kohonen, Arja Kuula-Luumi, Sanna-Kaisa Spoof. 2019. “Guidelines for Ethical Review in Human Sciences.” *Publications of the Finnish National Board on Research Integrity 3/2019* 3.

Iivari, Juhani, and Erkki Koskela. 1987. “The PIOCO Model for Information Systems Design.” *MIS Quarterly: Management Information Systems* 11 (3): 401–19. https://doi.org/10.2307/248688.

Järvinen, Pertti, and Annikki Järvinen. 1997. *Tutkimustyön Metodeista*.

Kahn, Robert L, Donald M Wolfe, Robert P Quinn, J Diedrick Snoek, and Robert A Rosenthal. 1964. “Organizational Stress: Studies in Role Conflict and Ambiguity.”

Kosonen, Risto, Mervi Ahola, Kirsi Villberg, and Tarja Takki. 2011. “Perceived IEQ Conditions: Why the Actual Percentage of Dissatisfied Persons Is Higher than Standards Indicate?” *Sick Building Syndrome*, 75–88. https://doi.org/10.1007/978-3-642-17919-8_4.

Kraut, Robert E., and Lynn A. Streeter. 1995. “Coordination in Software Development.” *Communications of the ACM* 38 (3): 69–82. https://go.gale.com/ps/i.do?p=AONE&sw=w&issn=00010782&v=2.1&it=r&id=GALE%7CA16764439&sid=googleScholar&linkaccess=fulltext.

Kroenke, David, Deborah Bunker, and David Wilson. 2013. *Experiencing MIS*. Pearson Higher Education AU.

Łuba, Tadeusz, and Janusz Rybnik. 1992. “Rough Sets and Some Aspects of Logic Synthesis.” In *Intelligent Decision Support*, 181–99. Springer Netherlands. https://doi.org/10.1007/978-94-015-7975-9_13.

Mason, Richard O. 1978. “Measuring Information Output: A Communication Systems Approach.” *Information and Management* 1 (4): 219–34. https://doi.org/10.1016/0378-7206(78)90028-9.

Miles, Robert Henry. 1974. *ROLE-CONFLICT AND AMBIGUITY IN BOUNDARY AND INTERNAL ROLES: A FIELD STUDY USING ROLE-SET ANALYSIS AND A PANEL DESIGN.* The University of North Carolina at Chapel Hill.

Myers, Brad, Sun Young Park, Yoko Nakano, Greg Mueller, and Andrew Ko. 2008. “How Designers Design and Program Interactive Behaviors.” *Proceedings - 2008 IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC 2008*, 177–84. https://doi.org/10.1109/VLHCC.2008.4639081.

Naur, P, and B Randell. 1969. “Software Engineering: Report of a Conference Sponsored by the NATO Science Committee, Garmisch, Germany, 7-11 Oct. 1968, Brussels, Scientific Affairs.” https://dl.acm.org/doi/abs/10.5555/1102020.

Nidumolu, Sarma. 1995. “The Effect of Coordination and Uncertainty on Software Project Performance: Residual Performance Risk as an Intervening Variable.” *Information Systems Research* 6 (3): 191–219. https://doi.org/10.1287/isre.6.3.191.

Nonaka, Ikujiro. 1994. “A Dynamic Theory of Organizational Knowledge Creation.” *Organization Science* 5 (1): 14–37. https://doi.org/10.1287/orsc.5.1.14.

Paasivaara, Maria, and Casper Lassenius. 2003. “Collaboration Practices in Global Inter-Organizational Software Development Projects.” *Software Process Improvement and Practice* 8 (4): 183–99. https://doi.org/10.1002/spip.187.

Royce, Winston W. 1970. “MANAGING THE DEVELOPMENT OF LARGE SOFTWARE SYSTEMS.” In *IEEE WESTCON*.

Sauer, Chris. 1993. "Why Information Systems Fail : A Case Study Approach." *Why Information Systems Fail : A Case Study Approach*. Information Systems Series. Henley-on-Thames: Waller.

Sauer, Chris. 1999. "Deciding the Future for IS Failures: Not the Choice You Might Think." *Rethinking Management Information Systems*, 279–309.

Seaman, Carolyn B. 1999. "Qualitative Methods in Empirical Studies of Software Engineering." *IEEE Transactions on Software Engineering* 25 (4): 557–72. https://doi.org/10.1109/32.799955.

Shannon, C E. 1948. "A Mathematical Theory of Communication*."

Stackoverflow. 2021. "Stackoverflow Developer Survey 2021." 2021. https://insights.stackoverflow.com/survey/2021.

Stasser, Garold, and William Titus. 1985. "Pooling of Unshared Information in Group Decision Making: Biased Information Sampling During Discussion." *Journal of Pereonality and Social Psychology*. Vol. 48.

Sutherland, Jeff, and Ken Schwaber. 2020. "The 2020 Scrum Guide." 2020. https://www.scrumguides.org/scrum-guide.html.

The Standish Group International, Inc. 2015. "Chaos Report 2015."

Wagner, A, E Gossauer, C Moosmann, Th Gropp, and R. Leonhart. 2007. "Thermal Comfort and Workplace Occupant Satisfaction-Results of Field Studies in German Low Energy Office Buildings." *Energy and Buildings* 39 (7): 758–69. https://doi.org/10.1016/j.enbuild.2007.02.013.

Waldrop, MM. 1993. "Complexity: The Emerging Science at the Edge of Order and Chaos." https://www.google.com/books?hl=fi&lr=&id=VP9TWZtVvq8C&oi=fnd&pg=PA9&ots=_1ZDSfMvpL&sig=EL16U_0pnKWvNvcOy98Ab0BrKCA.

Zhu, Haibin, Meng Chu Zhou, and Pierre Seguin. 2006. "Supporting Software Development with Roles." *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans* 36 (6): 1110–23. https://doi.org/10.1109/TSMCA.2006.883170.

APPENDICES

Appendix 1. Open ended interview questionnaire template

Q: What is your role? Please describe it.

Q: Why is there such a role?

Q: Who do you work with?

Q: Talking about collaboration among the other role – is that collaboration voluntary or is it prescribed by rules or guidelines?

Q: How do you work?

Q: What methods do you employ in your work?

Q: What tools do you use?

Q: What artefacts do you create or deliver?

Q: Who uses these artefacts?

Q: Thinking about a linear time systems development lifecycle – where in it your work happens? Where in that timeline you are most active?

Q: What do you not do?

Appendix 2. MIS Success Factors

System Quality	Information Quality	Information Use	User Satisfaction	Individual Impact
Data accuracy	Importance	Amount of use / duration of use	Satisfaction with specifics	Information understanding
Data currency	Relevance	Number of inquiries	Overall satisfaction	Learning
Database contents	Usefulness	Amount of connect time	Single-item measure	Accurate interpretation
Ease of use	Informativeness	Number of functions used	Multi-item measure	Information awareness
Ease of learning	Usableness	Number of records accessed	Information satisfaction: difference between information needed and received	Information recall
Convinience of access	Understandability	Frequency of access	Enjoyment	Problem identification
Human factors	Readibility	Frequency of report requests	Software satisfaction	Decision effectiveness: decision quality
Realization of user requirements	Clarity	Number of reports generated	Decision-making satisfaction	Improved decision analysis
Usefulness of system features and functions	Format	Charges for system use		Correctness of decision
System accuracy	Appearance	Regularity of use		Time to make decision
System flexibility	Content	Use by whom?		Confidence in decision
System reliability	Accuracy	Direct vs. chauffeured use		Decision-making participation
System sophistication	Precision	Binary use: use vs. nouse		Improved individual productivity
Integration of systems	Conciseness	Actual vs. reported use		Change in decision
System efficiency	Sufficiency	Nature of use: use for intended purpose		Causes management action
Resource utilization	Completeness	Appropriate use		Task performance
Response time	Reliablity	Type of information used		Quality of plans
Turnaround time	Currency	Purpose of use		Individual power or influence
	Timeliness	Levels of use: general vs. specific		Personal valuation os IS
	Quniqueness	Recurring use		Willingness to pay for information
	Comparability	Institutionalization / routinization of use		
	Quantitaviness	Report acceptance		
	Freedom from bias	Percentage used vs. opportunity for use		
		Voluntariness of use		
		Motivation to use		

Figure 5 - MIS Success Factors (DeLone and McLean 1992)

6.1 Appendix 3. Workspace layouts

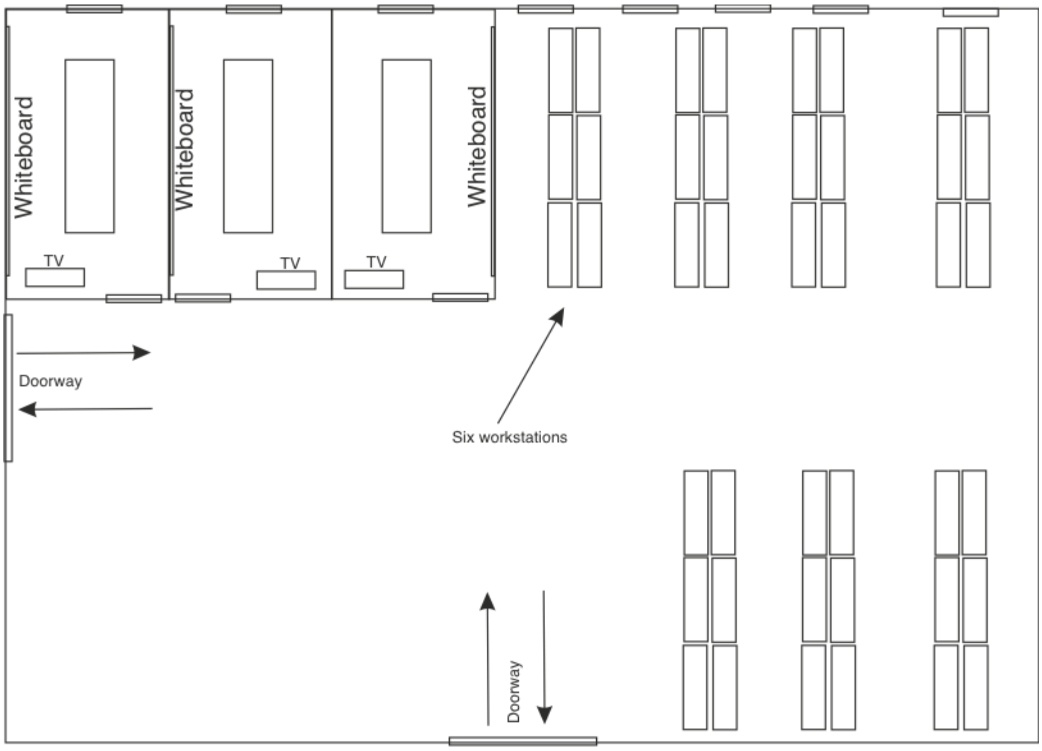


Figure 6 – Space A layout

Figure 7 – Space A meeting room with wall-to-wall whiteboard

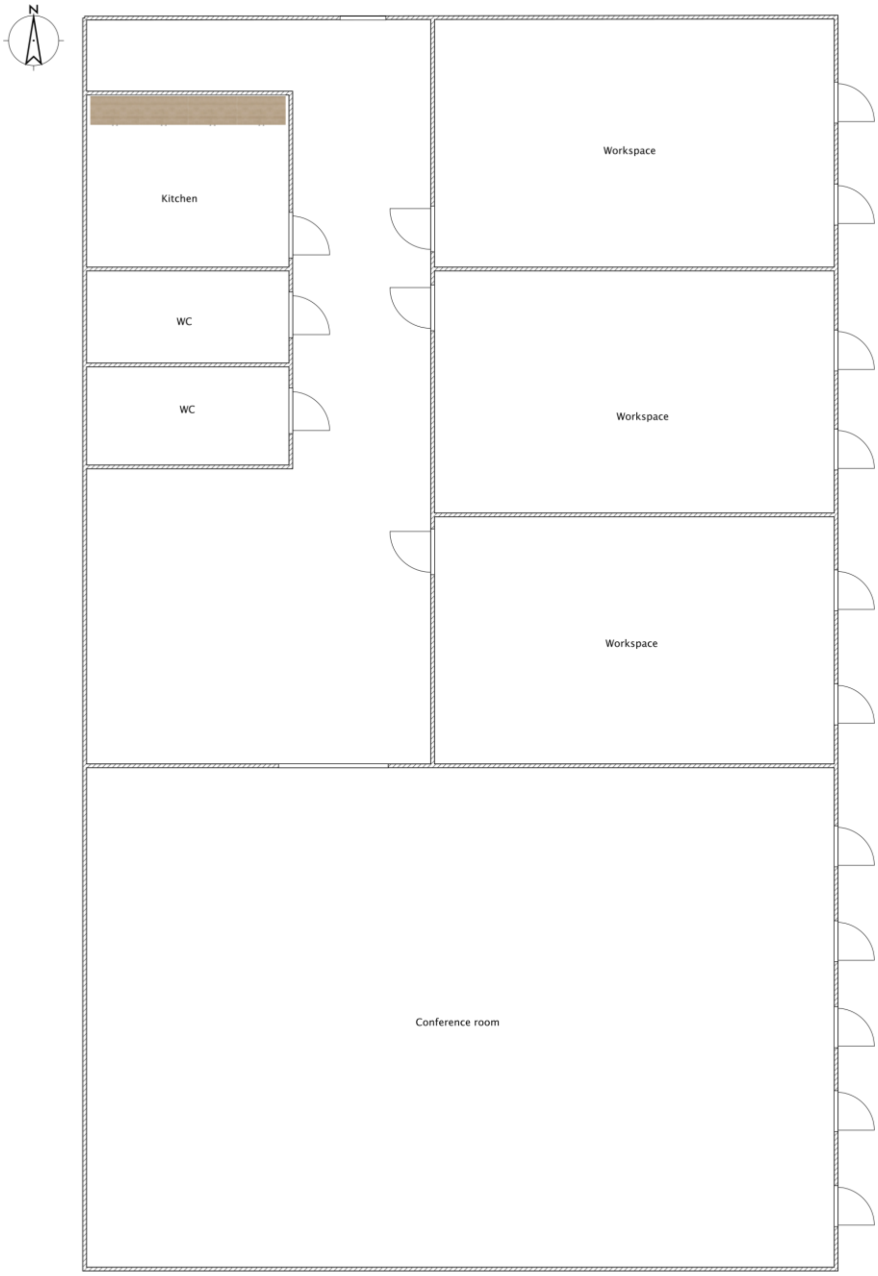


Figure 8 - Space B layout

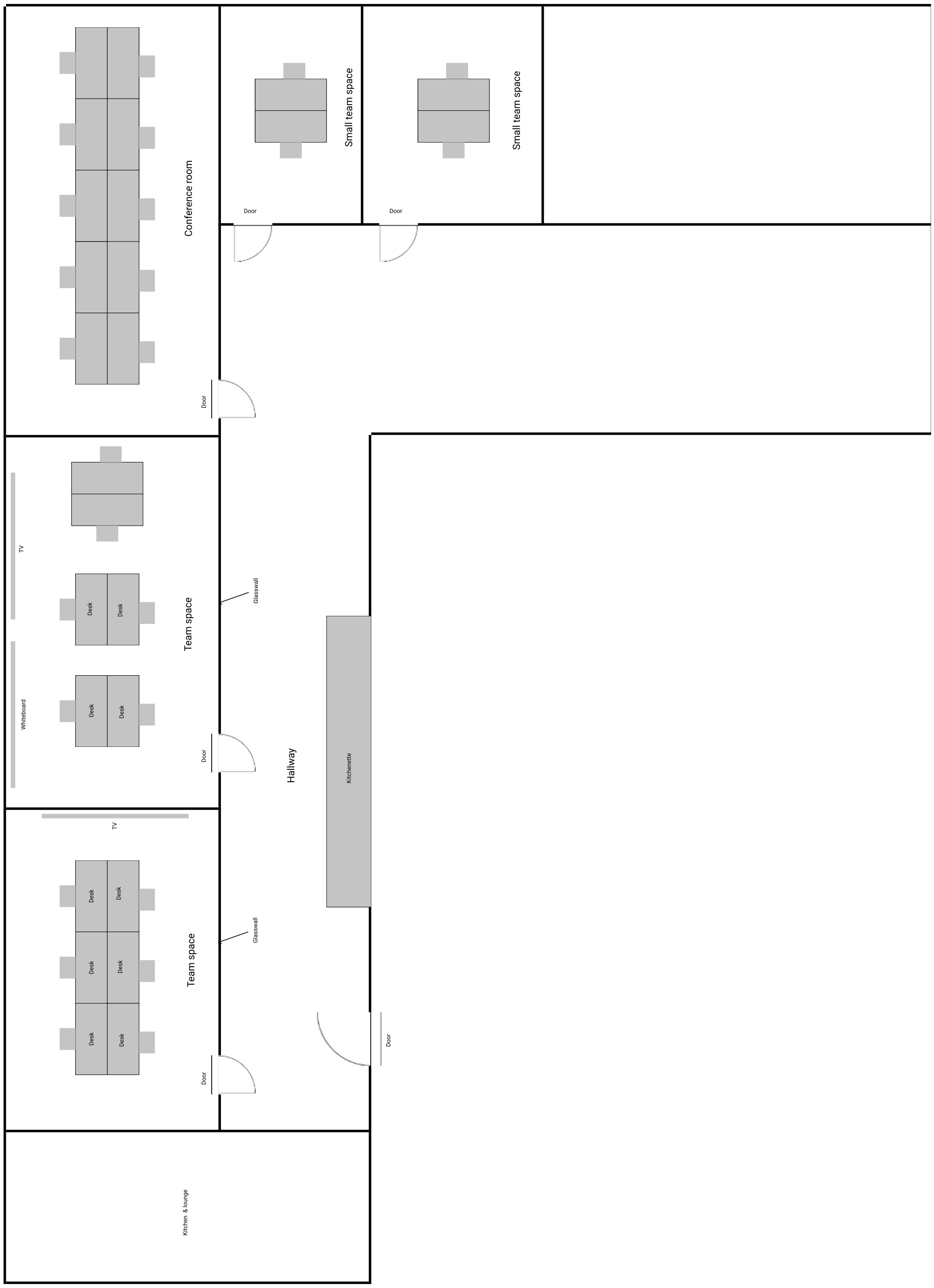


Figure 9 - Space C layout