Facilitating awareness in radically collocated software projects

Amos Ahola

Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Engineering, Espoo December 12th 2008

Supervisor: Professor Reijo Sulonen
Instructor: DI Tuomas Niinimäki
Radical collocation is a method which is used in software development and it means placing the entire software development team for the duration of the entire project in a workspace where are no other people than the ones that are working in the team or are related to the project. Radical collocation has shown good results, and the studies about it present that one of the reasons why it is successful because it increases awareness ("knowledge of what's going on") of the team members.

The goal of this thesis is to find out how radical collocation increases awareness, which in this thesis is defined as up-to-date knowledge about a software project which is necessary for successful coordination and collaboration. In addition to this, the thesis also tries to examine how awareness information is used in radically collocated software projects, and how the workspace could be designed to maximize it.

The research is conducted as a qualitative study through a pilot study with an unstructured interview and observation of a radically collocated workspace. This is in addition to the actual case study consisting of eight interviews and an observation of a workspace, which is a dedicated part of an open-office plan.

The most important finding is a causality chain, where radical collocation is a strong catalyst for social identification, which is a prerequisite for abundance of communication, which itself is necessary for high level of awareness.

The results meet the goals of the thesis well, and are significant in understanding why radical collocation is successful. In addition to this, improvement suggestions are given to workspace design, and other mechanisms that concern awareness in radically collocated software projects are discussed.

**Keywords:** Awareness, workspace, software development, radical collocation
Yhteensijoittaminen ("Radical collocation") on metodi jota käytetään ohjelmistokehityksessä ja se tarkoittaa sitä, että sijoitetaan koko ohjelmistokehitystiimi koko projektiin ajaksi työtilaan, joka on eristetty projektin ulkopuolisista henkilöistä. Yhteensijoittaminen on osoittautunut toimivaksi, ja sitä käsittelevät tutkimukset esittävät, että yksi syy siihen on se, että se lisää tiimin jäsenten tietoisuutta ("Awareness") siitä mitä projektissa tapahtuu. Tämän diplomityön tavoite on selvittää miten yhteensijoittaminen lisää tietoisuutta (joka on tässä diplomityössä määritelty ajan tasalla olevana tietona, joka on tarpeen jotta yhteistyö ja koordinaatio voivat onnistua ohjelmistoprojektissa). Tämän lisäksi diplomityö yrittää selvittää miten tätä tietoisuutta käytetään hyväksi yhteensijoitetuissa ohjelmistoprojekteissa, ja miten työtila voidaan suunnitella tietoisuuden maksimoimiseksi.

Tutkimus toteutettiin laadullisena ja siihen sisältyi pilottitutkimus, joka koostui strukturomattomasta haastattelusta ja yhteensijoitetun tiimin tiimin. Tämän lisäksi toteutettiin cäse-tutkimus, joka koostui kahdeksasta puolistrukturoidusta haastattelusta, sekä projektiin käytetyn työtilan tarkkailusta.

Tärkein yksittäinen havainto oli syy-seuraussuhde, missä yhteensijoittaminen on vahva katalyyti sosiaaliselle identifiikaatiolle, joka on ennakkovaatimus runsaalle kommunikaatiolle, joka taas on tarpeen, jotta korkea tietoisuuden taso voidaan saavuttaa.

Tämä työ täyttää asetetut tavoitteet hyvin, ja tulokset ovat merkittäviä radikaalin kollokaation menestyksen ymmärtämisessä. Tämän lisäksi työssä esitetään ehdotuksia miten työtiloja voitaisiin suunnitella, ja käsitellään mekanismeja, joilla tietoisuutta voidaan lisätä yhteensijoitetuissa ohjelmistoprojekteissa.

Avainsanat: Tietoisuus, työtila, ohjelmistokehitys, yhteensijoittaminen
Acknowledgements

This master’s thesis was written during the year 2008, and the people who are indirectly responsible for this are:

Peitsa Hynninen - you were invaluable in weaving this mesh of data and mixed topics to a concrete thesis

Tuomas Niinimäki - for answering all my questions that needed to be answered, and persistently reading through the ever so incomplete work

Shosta Sulonen - for telling me that preaching is not enough in all occasions, but substance is sometimes required too

My family - for support in all that I do, and especially Tuomas Ahola for giving me academic but brotherly advice on my thesis

In Espoo, 12th of December 2008

Amos Ahola
# Table of contents

1 Introduction .................................................................................................................................................................................. 1

1.1 Definitions ............................................................................................................................................................................. 2

1.2 Motivation .............................................................................................................................................................................. 2

1.3 The objective of the research ................................................................................................................................................ 2

1.4 Research questions ............................................................................................................................................................. 3

1.5 Scope ..................................................................................................................................................................................... 3

1.6 Structure of the thesis .......................................................................................................................................................... 4

2 Theoretical framework ................................................................................................................................................................. 5

2.1 Introduction ........................................................................................................................................................................... 5

2.2 Software development .......................................................................................................................................................... 6

2.2.1 The characteristics of software development .................................................................................................................. 6

2.2.2 Plan-driven software development ................................................................................................................................ 7

2.2.3 Agile software development ........................................................................................................................................... 9

2.2.4 Distributed software development .................................................................................................................................. 10

2.2.5 Radical collocation ........................................................................................................................................................... 10

2.3 Awareness ............................................................................................................................................................................. 13

2.3.1 What is awareness? ........................................................................................................................................................ 13

2.3.2 Awareness and software development .......................................................................................................................... 15

2.3.3 The workspace awareness framework .......................................................................................................................... 16

2.4 Physical workspace..................................................................................................................................................................... 19

2.4.1 The office ......................................................................................................................................................................... 19

2.4.2 The effect of close proximity on communication ........................................................................................................ 20

2.4.3 Project artifacts .............................................................................................................................................................. 22

2.4.4 Social identification ........................................................................................................................................................ 22

2.5 Summary................................................................................................................................................................................ 23

3 Research methodology .................................................................................................................................................................. 26

3.1 The pilot study ......................................................................................................................................................................... 26

3.1.1 Overview of the pilot group ........................................................................................................................................... 26

3.1.2 Observation ...................................................................................................................................................................... 26

3.1.3 Interview .......................................................................................................................................................................... 27

3.1.4 Limitations ...................................................................................................................................................................... 27

3.2 The case study .......................................................................................................................................................................... 28

3.2.1 Overview of the case project ........................................................................................................................................ 28

3.2.2 Observation ...................................................................................................................................................................... 29

3.2.3 Interviews ........................................................................................................................................................................ 29

3.2.4 Limitations ...................................................................................................................................................................... 30

3.3 Analysis of the gathered data ................................................................................................................................................ 30

4 Results ....................................................................................................................................................................................... 32

4.1 The pilot study results ............................................................................................................................................................ 32

4.1.1 The radically collocated workspace .................................................................................................................................. 32

4.1.2 The communication patterns ......................................................................................................................................... 33

4.1.3 Ad-hoc pair and group programming ............................................................................................................................ 34

4.2 The case study results ............................................................................................................................................................ 35

4.2.1 The dedicated workspace ................................................................................................................................................ 35

4.2.2 Previous experiences on a radically collocated workspace ........................................................................................... 38

4.2.3 Opinions on the current dedicated workspace ........................................................................................................... 40

4.2.4 Instant messaging ........................................................................................................................................................... 44

4.2.5 Expectations about the upcoming radically collocated workspace ................................................................................ 45
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 Discussion</td>
<td>48</td>
</tr>
<tr>
<td>5.1 The effect of radical collocation on awareness</td>
<td>48</td>
</tr>
<tr>
<td>5.2 The communication cycle</td>
<td>49</td>
</tr>
<tr>
<td>5.2.1 Information pull</td>
<td>50</td>
</tr>
<tr>
<td>5.2.2 Information push</td>
<td>51</td>
</tr>
<tr>
<td>5.3 Maximizing the benefits of radical collocation in software projects</td>
<td>53</td>
</tr>
<tr>
<td>5.3.1 Display projection</td>
<td>53</td>
</tr>
<tr>
<td>5.3.2 Seating</td>
<td>54</td>
</tr>
<tr>
<td>5.3.3 Communication protocol</td>
<td>55</td>
</tr>
<tr>
<td>5.4 Individual learning versus team learning</td>
<td>57</td>
</tr>
<tr>
<td>5.5 Forcing awareness</td>
<td>58</td>
</tr>
<tr>
<td>6 Summary and conclusions</td>
<td>59</td>
</tr>
<tr>
<td>6.1 How well were the goals of this thesis met</td>
<td>59</td>
</tr>
<tr>
<td>6.2 Limitations</td>
<td>59</td>
</tr>
<tr>
<td>6.3 Implications for further research</td>
<td>60</td>
</tr>
<tr>
<td>6.4 Conclusions</td>
<td>61</td>
</tr>
<tr>
<td>References</td>
<td>62</td>
</tr>
<tr>
<td>Appendixes</td>
<td>69</td>
</tr>
<tr>
<td>Appendix A: The communication log sheet</td>
<td>69</td>
</tr>
<tr>
<td>Appendix B: The interview guide</td>
<td>70</td>
</tr>
</tbody>
</table>
1 Introduction

Software industry is going through radical change. Software engineering is no longer bound to traditional engineering ways, as plan-driven software development is giving way to agile methods. In addition to this, distributed software development is increasing drastically, as software companies wish to reap the benefits of globalization.

These trends are not without problems. One major contributor to difficulties in software projects is the lack of coordination. In software development, different people working on the same project must coordinate their work so it gets done and fits together, no work is done more than once, and that all the parts of the software are delivered quickly. (Kraut, Streeter 1995)

This requires awareness, ‘knowing what is going on’ (Endsley 1995), which is also critical to successful collaboration in teamwork (Dourish, Bellotti 1992). Teasey et. al (2000) introduced a method called radical collocation, in which they placed the whole software project team in the same physical workspace for the duration. Studies about it (Teasley et al. 2000, Teasley et al. 2002) show, that it increases productivity and employee satisfaction, but at some level in the expense of privacy and concentration. Teasley et al. (2002) suggest that one of the reasons why radical collocation is successful is because the common physical workspace increases the awareness of the team members. That process is called ‘Workspace awareness’ (Gutwin, Greenberg & Roseman 1996).

Awareness has been an increasingly popular topic in computer supported cooperative work (CSCW), and for the past 15 years, several researchers have pondered the true nature of awareness tried to explain what it is. This has lead to ambiguous and even contradictory use of the word, and “it is becoming increasingly clear, that the term ‘awareness’ does not denote a set of practice. In fact, it is hardly a concept any longer.” (Schmidt 2002)

In this thesis, the term “awareness” is used as a bridge between software development and the physical workspace in a pragmatic approach – the author wishes to study how the radical collocation increases it, how the awareness is used and how the workspace could be designed to increase the total awareness of the team members. The answers to these questions can prove to be useful in understanding the nature of collaboration in software development, and provide concrete
advice on how to design a workspace in a way it will meet the needs of a software development team.

1.1 Definitions

Awareness - to cope with the ambiguity of the term awareness, in this thesis, awareness is defined as *up-to-date knowledge about a software project, which is necessary for successful coordination and collaboration.*

Collaboration - *the process of working together in pursuit of common objectives.*

Coordination - *managing dependencies between activities.* (Malone, Crowston 1994)

Software project - *a unique venture with a beginning and an end, undertaken by people to create software within defined constraints of scope, time, resources, and quality.*

Up-to-date knowledge - *knowledge about an issue, which has not changed since the knowledge was acquired.*

1.2 Motivation

Software projects are difficult to coordinate due to the inherent characteristics of the field, which are discussed further in the chapter 2.2.1 **Characteristics of software development.** This combined with the notion that current day software development is a social process, where collaboration is important (Booch, Brown 2003), creates the need for awareness in software projects. This thesis focuses on the physical workspace design in fulfilling that need.

1.3 The objective of the research

The first thing in discovering how the physical workspace should be designed so it would enable coordination and collaboration in a software project is to discovering which mechanisms and features in the common physical workspace increase (or decrease) the amount of awareness among team members. Second, it must be found out how team members actually use that knowledge in their daily work. Third, improvement suggestions to the existing workspace can are given.
The three objectives of this thesis can be presented as follows: how does radical collocation increase awareness in software projects, how does the radical collocation enhance coordination and collaboration in software projects, and how could the benefits of radical collocation be increased.

1.4 Research questions

RQ1: How does radical collocation increase awareness in software projects?

RQ2: How does radical collocation enhance coordination and collaboration in software projects?

RQ3: How should the physical workspace be arranged to increase the benefits of radical collocation in software projects?

1.5 Scope

The scope of this thesis is limited to the coordination and collaboration of an intra-organizational and intra-national small (10 persons or less, so that they could fit in the same workspace) radically collocated software development teams. This thesis focuses on the issue in the viewpoint of software development. The software process used and the state of the project can influence the need for coordination and collaboration (and therefore awareness), but this thesis will not address these issues (when the need for awareness is highest, and when it is lowest), but instead will discuss the matter in terms of maximizing awareness. This thesis will give only little thought to spatial factors that are not related to coordination and collaboration (such as aesthetics).
1.6 Structure of the thesis

The structure of the thesis is presented in Figure 1.

Chapter 1 Introduction presents the motivation for the study and based on this, the goal of the study and the research questions are formulated. In addition to this, the term awareness is defined and the scope of the thesis is given.

Chapter 2 Theoretical framework is constructed from the previous work included in the scope of this thesis.

Chapter 3 Research methodology presents the research methods used and an overview of the case projects.

Chapter 4 Results produces the actual results from the case studies.

Chapter 5 Discussion discusses the implications that can be derived from the case study.

Chapter 6 Summary and conclusions is self-explanatory. In addition to this, the applicability of the used methods and the reliability of the study are discussed, and the suggestions for future studies are presented.
2 Theoretical framework

2.1 Introduction

Three discourses can be derived from the research questions: software development, awareness, and physical workspace. The discourses are presented in Figure 2. The purpose of this chapter is to discuss the relevant aspects of previous work. Each dimension is unfolded under the corresponding topic and previous work that overlaps two or more dimensions is considered relevant and is presented in the most corresponding section.
2.2 Software development

The focus on this chapter is on discussing the elements in software development relevant in terms of coordination and collaboration. This is achieved by presenting the characteristics of software engineering affecting coordination and collaboration, how the software development processes has evolved in the course of the past decades, and what is the method called radical collocation.

2.2.1 The characteristics of software development

Kraut et al. (1995) identify and describe four characteristics of software development, which make coordination problems common in software development, and therefore increase the need of awareness (Dourish, Bellotti 1992). These characteristics are scale, uncertainty, interdependence and informal communication. In turn, Cockburn (2002) describes software development as a thinking-intensive and communication-intensive activity, and Brooks (1979) took notice of the major difference in programmer productivity.

**Scale**

Software systems are often so large, that comprehending them in detail is impossible for a single person, or even for a group of people. This leads to division of work, which in turn creates boundaries. These boundaries can be geographic, organizational or social, and they hinder coordination. (Kraut, Streeter 1995)

**Uncertainty**

Many software systems are one-of-a-kind, since they are tailored to fill the needs of a new customer, or the new needs of an existing customer. This means that software development is a non-routine action, and because of that, the results are uncertain, unlike in a manufacturing process, where all products are alike. In addition to this, the scale issue makes software projects time-consuming, so in the duration of the projects, the customer’s needs may change. The uncertainty makes the problems created by the scale worse. (Kraut, Streeter 1995)
Interdependence
Software components depend very much from each other, so the individual work of a developer must be integrated to the work of others. This compounds the problems created by the scale and the uncertainty, since it distributes the effect of one change through the whole project. (Kraut, Streeter 1995)

Informal communication
Formal communication methods, such as written documents, status review meetings, automated reporting and error tracking do not fulfill all the communication needs in software development due to the uncertainty inherent to software projects and therefore informal communication is necessary. Informal communication is described as personal, peer-oriented and interactive communication. (Kraut, Streeter 1995)

Thinking-intensive and communication-intensive activity
Software development is an activity, where one needs to balance between concentrating on personal work and communicating with others. At some points, a programmer must focus solely on the task at hand, in order to complete it well. On the other hand, communication between team members is necessary too. (Cockburn 2002)

Major differences in programmer productivity
In software development, due to individual characteristics and skills, the productivity differences between programmers can be tenfold. This results in difficulties in project planning and implementation, because resources are not completely interchangeable. This means that one can not necessarily complete every programming task in the project, or it will take considerably longer for him than for someone else. (Brooks 1979)

2.2.2 Plan-driven software development
The traditional way to build software is called plan-driven software development, and it can be characterized by a systematic engineering process, which has several input-output phases. Figure 3 presents the phases in the classic waterfall life cycle model (Ghezzi, Jazayeri & Mandrioli 1991). Each phase has input, against which the output can be validated. This puts significant value for the completeness of the documentation in every phase, since it is the main source for verification, and
important for the requirement traceability throughout the whole project. (Boehm, Turner 2003, Chau, Maurer & Melnik 2003)

Figure 3: The waterfall model of the software life cycle (Ghezzi, Jazayeri & Mandrioli 1991)

The problem in this constant documentation lies in the ever increasing workload of creating new documents, and keeping the old ones up-to-date (Chau, Maurer & Melnik 2003). Still, the abundance of documentation is not without benefits, as it makes it possible to use documentation as means of communication and diminishes the loss of information if a key member of the team leaves the project (Boehm, Turner 2003).

Chau et al. (2003) emphasize that plan-driven development uses documentation as the primary medium in knowledge sharing, which makes knowledge sharing time-and space-independent. This is the opposite of the preference for face-to-face communication in agile methodology, which is presented next.
2.2.3 Agile software development

The cost of changes grows through the software development life cycle (Boehm 1981). Conforming this, plan-driven software development tries to reduce the cost by stopping the changes early in the software projects by collecting a complete set of requirements in the beginning of the project. This approach is too slow in many cases in the ever changing business environment of today. To tackle the problems created by the changing business environment, agile software development methods emerged. (Highsmith, Cockburn 2001, Boehm, Turner 2003)

The essential core of these methods was formally introduced by the Agile Alliance in 2001. The Agile Manifesto (Fowler, Highsmith 2001) stated: “We are uncovering better ways of developing software by doing it and helping others do it. We value:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation.
- **Customer collaboration** over contract negotiation.
- **Responding to change** over following a plan.” (Fowler, Highsmith 2001)

Agile methods can be characterized as lightweight processes with short iterative cycles, user involvement and tacit knowledge sharing. The iterative cycles mean that the whole product is not delivered at once, but incrementally built in short periods called cycles, and between each cycle, the product is delivered to the customer. The end users of the software are actively involved in the process, establish, prioritizing and verifying requirements. Excessive documentation is seen as burden, because the change is constant. To manage the project without a complete set of documentation, information is tacitly shared inside the team. (Boehm, Turner 2003)

An example of an agile practice is pair programming. It is a setting where two developers work at the same workstation at the same time, solving problems and programming together by taking turns. This has proven to enhance the technical skills of the participants and improve team communication, and it is considered more enjoyable than working in solitude. (Cockburn, Williams 2001, Williams, Kessler 2002)
Recently, Turk et al. (2005) have depicted that there are assumptions behind the principles of the Agile Alliance, which are often unstated in the work relating to agile processes. One of these assumptions is as follows: "Developers are located in a time and space such that they are able to have frequent, intensive communication with each other." If this assumption does not hold, agile processes are less likely to be applicable. (Turk, France. R 2005)

Highsmith et al. (2001) emphasize that agile methods suggest using social activities efficiently in knowledge sharing. They perceive face-to-face communication as a better and faster way to transfer ideas than documentation, and group work is considered to produce better results in software projects, than individual work.

### 2.2.4 Distributed software development

In the beginning of the 21st century, software had become one vital component in most industries as success in the market depends even more on using software as a competitive edge. Even before that companies had developed software in remote locations, as they aimed for cheaper and skilled labor, but in the 21st century the phenomenon exploded. Now software development is increasingly globally distributed. (Herbsleb, Moitra 2001)

The definition of "Virtual" — "not sitting together" (Cockburn 2002) can be used to describe distributed software development. The actual setting for distributed software projects can vary in the amount of geographical locations, team size in each location, business models, software processes. The experiences on distributed software development indicate that they are more time consuming than comparable collocated projects, and this is due to challenges in communication and coordination. These challenges originate from the distance, difference in time, cultural differences, organizational resistance and technical issues. (Cockburn 2002, Herbsleb, Moitra 2001)

### 2.2.5 Radical collocation

The concept of radical collocation is to put an entire software development team in the same room for the duration of the project. The purpose for this is to overcome communication delays and breakdowns. Radical collocation increases productivity and employee satisfaction, but this is not without tradeoffs: it wrecks privacy, quiet and possibly concentration. (Teasley et al. 2000)
The difference between distributed, collocated and radically collocated software development is illustrated in Figure 4.

<table>
<thead>
<tr>
<th>Distributed</th>
<th>Collocated</th>
<th>Radically collocated</th>
</tr>
</thead>
<tbody>
<tr>
<td>team members reside in</td>
<td>team members reside in the</td>
<td>team members reside in</td>
</tr>
<tr>
<td>different geographical</td>
<td>same building, but in</td>
<td>the same room</td>
</tr>
<tr>
<td>locations</td>
<td>different rooms</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 4: Different settings in software development**

Experiences on radical collocation show that the problems related to concentration and privacy can be mostly overcome as the team members learn to phase in and out of the simultaneous work, and get used to the increased familiarity with their teammates. The main problem identified was not related to the noise in the workspace, but to the individual recognition of work. Developers were worried that the management could see their individual contribution to the project. (Teasley et al. 2000)

Teasley et al. (2002) present that radical collocation is successful because the team members like working in team rooms, it enhances communication and coordination, creates awareness among the team members, decreases disturbances that were not related to the project to the minimum, is perceived as a good learning environment, contributes to team unity and group mentality, and decreases the time people on non-work related activities.

Even though radical collocation has been proven to be a beneficial method in software development, it still isn’t “the silver bullet”, since a manageable project scope and small team sizes might be prerequisites for reaping the most benefits out of radical collocation. (Teasley et al. 2002)

Two room layouts experimented in radical collocation by Teasley et al. (2002) are presented in Figure 5. Teasley et al. did not argument why these setting were used, nor did they explain why two different settings were used.
Figure 5: Two room layouts used in radical collocation (Teasley et al. 2002)
2.3 Awareness

This chapter discusses the previous work on awareness, illustrating how ambiguous the term "awareness" actually is. The aspects of awareness which are important in the focus of this thesis, meaning it concerns software development or the physical workspace, are dissected further under the topics awareness and software development and the workspace awareness framework.

2.3.1 What is awareness?

In general, awareness is the concept of ‘knowing what is going on’ (Endsley 1995), or “An understanding of the activities of others which provides a context for your own activity” (Dourish, Bellotti 1992). Gutwin et al. (2004) described awareness with four basic characteristics:

1. Awareness is knowledge about the state of a particular environment.
2. Environments change over time, so awareness must be kept up to date.
3. People maintain their awareness by interacting with the environment.
4. Awareness is usually a secondary goal – that is, the overall goal is not simply to maintain awareness but to complete some task in some environment.

The concept of awareness has been studied by many researchers in the past, resulting in a variety of ways awareness is perceived. The identified awareness types are presented in Table 1.
<table>
<thead>
<tr>
<th>Awareness type</th>
<th>Description</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group awareness</td>
<td>what everyone else in the same team is doing, their state and attributes</td>
<td>(Kantor, Zimmermann &amp; Redmiles 1997, Hinds, Kiesler 2002)</td>
</tr>
<tr>
<td>Informal awareness</td>
<td>the state of others in the work community</td>
<td>(Gutwin, Greenberg &amp; Roseman 1996)</td>
</tr>
<tr>
<td>Group-structural awareness</td>
<td>the group’s organization and of the working relationships in it</td>
<td>(Gutwin, Greenberg &amp; Roseman 1996)</td>
</tr>
<tr>
<td>Team awareness</td>
<td>other team member’s attributes</td>
<td>(Hinds, Kiesler 2002)</td>
</tr>
<tr>
<td>Casual awareness of others in work groups</td>
<td>the social state of others in the work group</td>
<td>(Borning, Travers 1991)</td>
</tr>
<tr>
<td>Social awareness</td>
<td>social connections and roles in the group/knowledge of the social situation of someone</td>
<td>(Gutwin, Greenberg &amp; Roseman 1996, Prinz 1999)</td>
</tr>
<tr>
<td>Background awareness</td>
<td>who is around you and what is happening</td>
<td>(Bly, Harrison &amp; Irwin 1993)</td>
</tr>
<tr>
<td>Concept awareness</td>
<td>how new information fits to the previous knowledge</td>
<td>(Gutwin, Stark &amp; Greenberg 1995)</td>
</tr>
<tr>
<td>Peripheral awareness</td>
<td>what happens in the peripheral view</td>
<td>(Bly, Harrison &amp; Irwin 1993, Cadiz et al. 2002, Baecker et al. 1993)</td>
</tr>
<tr>
<td>Workspace awareness</td>
<td>what happens in the same workspace, and how it relates to the workspace</td>
<td>(Gutwin, Greenberg 2002, Vertegaal, Velichkovsky &amp; van der Veer, G. 1997)</td>
</tr>
<tr>
<td>Situational awareness</td>
<td>what happens around you</td>
<td>(Endsley, Garland 2000)</td>
</tr>
<tr>
<td>Passive awareness</td>
<td>information transferred without the need for attention</td>
<td>(Dourish, Bellotti 1992)</td>
</tr>
<tr>
<td>Project awareness</td>
<td>the progress of the project in the viewpoint of tasks and personnel</td>
<td>(Anderson, Bouvin 2000, Kantor, Zimmermann &amp; Redmiles 1997)</td>
</tr>
<tr>
<td>Activity awareness</td>
<td>project work that supports group performance in complex tasks</td>
<td>(Carroll et al. 2003)</td>
</tr>
<tr>
<td>Reciprocal awareness</td>
<td>understanding of what someone else is doing</td>
<td>(Schmidt 1994)</td>
</tr>
<tr>
<td>Mutual awareness</td>
<td>shared understanding of the state of the environment</td>
<td>(Schmidt 1994)</td>
</tr>
<tr>
<td>Conversational awareness</td>
<td>who is communication with whom</td>
<td>(Vertegaal, Velichkovsky &amp; van der Veer, G. 1997)</td>
</tr>
<tr>
<td>Availability awareness</td>
<td>status of others availability for an activity</td>
<td>(Lai et al. 2003)</td>
</tr>
<tr>
<td>Process awareness</td>
<td>where tasks fit in the stages of the process</td>
<td>(Dustdar 2002)</td>
</tr>
<tr>
<td>Collaboration awareness</td>
<td>a feature enabling collaboration of different software</td>
<td>(Lauwers, Lantz 1990)</td>
</tr>
<tr>
<td>Organizational awareness</td>
<td>association to people, reasons for being together and shared knowledge</td>
<td>(Gutwin, Greenberg &amp; Roseman 1996)</td>
</tr>
<tr>
<td>Historical awareness</td>
<td>how software artifacts have evolved in the course of history</td>
<td>(Nutter, Boldyreff 2003)</td>
</tr>
<tr>
<td>General awareness</td>
<td>who is around, what they are doing, if they are busy of free, meeting or alone, receptive to communication or not</td>
<td>(Gaver 1991)</td>
</tr>
</tbody>
</table>
In the scope of this thesis, the relevant parts of previous work can be presented in two parts: what awareness is necessary in software development, and the process of how the common physical workspace creates awareness. These are covered below under the topics **Awareness and software development** and the **workspace awareness framework**.

### 2.3.2 Awareness and software development

The few existing studies which concern awareness and software development are related to the use of awareness tools (software that provide awareness through visualizing human activities (Storey, Ćubranić & German 2005)) in distributed software development, emphasizing the necessity of group awareness (e.g. Gutwin, Penner & Schneider 2004, Hupfer, Ross & Patterson 2004) and the awareness of relationships (e.g. Nutter, Boldyreff 2003, Damian et al. 2003, Herbsleb et al. 2001, Magnusson, Asklund & Minör 1993, Froehlich, Dourish 2004, Sarma, Noroozi & van der Hoek, A. 2003) in distributed software projects.

Group awareness consists of knowing who are working on the project, if they are present and available now, what they are doing or working with now, what are their roles and responsibilities, what are their skills and knowledge, and how their work affects the work of others (Gutwin, Penner & Schneider 2004, Storey, Ćubranić & German 2005, Hinds, Kiesler 2002). In distributed software development, group awareness is often maintained through mailing lists and chat systems (Gutwin, Greenberg 2004). One example of a more advanced group awareness tool is “Jazz”, a collaborative development environment (Hupfer, Ross & Patterson 2004), which bundles communication tools and software development tools.

Tools providing awareness of relationships, e.g. Augur (Froehlich, Dourish 2004), try to provide insight of the software product as a whole, through combining and presenting relevant information from different sources, such as the version control system (CVS), the source code or the bug tracking system (Storey, Ćubranić & German 2005)
2.3.3 The workspace awareness framework

This sub-chapter presents the workspace awareness framework, which is constructed by (Gutwin, Greenberg 2002). Workspace awareness consists of up-to-the minute knowledge of what everyone else in the same workspace is doing related to the dimensions and tools in the workspace. This is made possible by the media-rich verbal and nonverbal interaction (Daft, Lengel 1986) and information flow between the people in the same space, such as speech, gestures, usage of a certain tool, physical location of a team member in the workspace. (Gutwin, Greenberg & Roseman 1996).

The workspace awareness framework describes what information is transferred in the workspace, how the information is used, and how the workspace eases the gathering and interpretation of the information. The workspace awareness framework is presented in Figure 6.

![Figure 6: The workspace awareness framework (Gutwin, Greenberg 2002)](image)

The essential up-to-date knowledge that is maintained through the workspace awareness process can be defined as an answer to “Who, Where, What, When, and How” questions. This means that we
know with whom we are working with, where are they working, what are they doing, when did something happen and how.

There are multiple factors that maintain that knowledge in a workspace: consequential communication, intentional communication, artifacts and feedthrough.

**Consequential communication**
The content of consequential communication can be described as information, which is transferred automatically through observing and hearing someone doing something in the workspace. This communication is unintentional, and it is not directed at anyone.

**Intentional communication**
Intentional communication can be described as communication, which is initiated for the purpose of transferring information. This communication can be directed towards a single person or a group. Intentional communication is discussed further under the topic 2.4.2 the effect of close proximity on communication.

**Artifacts and feedthrough**
In the context of workspace awareness, artifacts are physical objects in the workspace. When we see someone working on them, we gain information through feedthrough. Artifacts and feedthrough will be discussed further under the topic 2.4.3 Project Artifacts.

In collaboration, the workspace awareness can be used to reduce effort, increase efficiency and reduce errors. This is done through five activities: Management of coupling, Simplification of communication, Coordination of action, Anticipation, and Assistance.

**Management of coupling**
In group work, some tasks can be done alone, but some require more than one person. Workspace awareness enables easy transition between these two work types by making it possible to shift focus from one to another and back without delay.
Simplification of communication
The common physical workspace makes communication more efficient by simplifying it. The methods through the communication is simplified are presented later under the topic 2.4.2 the effect of close proximity on communication in Table 2.

Coordination of action
In group work, work should be done in the right time and in the right order. Workspace awareness eases this by providing information on what people are doing, without the need for communication.

Anticipation
Workspace awareness makes predicting other peoples' actions and activity possible in multiple timescales, meaning that the information we acquire about a person can be used to predict his actions in the present and in the future.

Assistance
Assisting others in their tasks is important in collaborative work. Workspace awareness eases this by making it easier to determine if help is needed and if it can be provided.
2.4 Physical workspace

"The provision of workspace is, or at least should be, a direct response to the considered needs of people (individually and collectively) in supporting them in their work endeavors (current and future)." (McGregor 2000)

This subchapter discusses how the physical workspace we know as the office has evolved in the past, and how it affects the work done in it and discusses the previous work on the advantages of a common physical workspace, which are relevant in the scope of this thesis. These advantages are depicted under the topics the effect of close proximity on communication, project artifacts and social identification.

2.4.1 The office

The two extremes in office design can be described as the traditional office and the open office. The characteristics of the traditional office are private rooms for the employees, whereas the open office can be characterized by the common open space, which is somehow distributed between the individuals working in it. The traditional office was challenged by the concept of open offices in the 1950s and this trend reached its peak of popularity in 1970s. The claimed benefits that were used in promoting open offices included flexibility of space, meaning that the space could be customized to meet the need of the organization, and the notion that getting rid of physical barriers between the employees increased communication. In addition to this, it was perceived that open offices reduced workspace related expenses. (Brennan, Chugh & Kline 2002)

The research on open offices has been controversial, providing both positive results, such as increased communication (Allen, Gerstberger 1973), as well as negative results, like increased disturbances and distractions (Oldham, Brass 1979). Recent studies (Brennan, Chugh & Kline 2002, Banbury, Berry 2005) tend to emphasize the negative results of open offices, such as decrease in satisfaction with the physical environment (in comparison to traditional offices), increased physical stress, decreased team member relations, lower perceived job productivity and deterioration of concentration.

According to DeMarco et al. (1999), aesthetics are often stressed too much in workspace design. This can cause problems if the space is designed to look good, instead of fulfilling the needs of the people working in it. The management can also hinder the use of the workspace by enforcing strict
order in the interior decoration, even if the decisions of individuals could beneficial to the individual or to someone else. The strict order can e.g. forbid leaving papers on the table overnight, or setting up posters or diagrams on the wall, which could be used for sharing information. (DeMarco, Lister 1999)

2.4.2 The effect of close proximity on communication

Allen (1979) illustrated that the decrease in communication over distance is exponential and reaches asymptote in 30 meters. In addition to maximizing the frequency of communication, being physically in the same workspace has two effects on communication: it lowers the effort to initiate communication and it enables overhearing. (Kraut et al. 2002, Gutwin, Penner & Schneider 2004)

When people are in the same workspace, initiating communication takes little effort and it is done whenever needed. As the frequency of communication is high, team members have many opportunities to exchange information about their personal competence, availability or about the task at hand. (Kraut et al. 2002)

In close proximity, intentional communication has a side-effect called overhearing. Communication between two peers turns, intentionally or unintentionally, into one-to-many communication when there are people close by. This makes communicated information available to everyone in the proximity. (Gutwin, Greenberg 2004)

The key characteristics of collocated synchronous interaction are presented in Table 2.
<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Description</th>
<th>Implication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rapid feedback</td>
<td>As interactions flow, feedback is as rapid as it can be</td>
<td>Quick corrections possible when there are noticed misunderstandings or disagreements</td>
</tr>
<tr>
<td>Multiple channels</td>
<td>Information along participants flows in many channels, voice, facial expressions, gesture, body posture and so on</td>
<td>There are many ways to convey a subtle or complex message; also provides redundancy</td>
</tr>
<tr>
<td>Personal information</td>
<td>The identity of contributors to conversation is usually known</td>
<td>The characteristics of the source can be taken in to account</td>
</tr>
<tr>
<td>Nuanced information</td>
<td>The kind of information that flows is often analog or continuous, with many subtle dimensions (e.g., gestures)</td>
<td>Very small differences in meaning can be conveyed; information can easily be modulated</td>
</tr>
<tr>
<td>Shared local context</td>
<td>Participants have a similar situation (time of day, local events)</td>
<td>A share frame on the activities; allows for easy socializing as well as mutual understanding about what is on each others' minds</td>
</tr>
<tr>
<td>Informal “hall” time before and after</td>
<td>Impromptu interactions take place along subsets of participants on arrival and departure</td>
<td>Opportunistic information exchanges take place, and important social bonding occurs</td>
</tr>
<tr>
<td>Coreference</td>
<td>Ease of establishing joint reference to objects</td>
<td>Gaze and gesture can easily identify the referent of deictic terms</td>
</tr>
<tr>
<td>Individual control</td>
<td>Each participant can freely choose what to attend to and change the focus of attention easily</td>
<td>Rich, flexible monitoring of how all of the participants are reacting to whatever is going on</td>
</tr>
<tr>
<td>Implicit cues</td>
<td>A variety of cues as to what is going on are available in the periphery</td>
<td>Natural operations of human attention provide access to important contextual information</td>
</tr>
</tbody>
</table>
| Spatiality of reference              | People and work objects are located in space                                                                                                                                                               | Both people and ideas can be referred spatially; “air boards”                                                                Adam148154964
2.4.3 Project artifacts

Hutchins (1995) describes artifacts as ‘The Crystallization of Knowledge and Practice’; they allow externalization and representation of ideas. Commonly used artifacts in project work are e.g. sketches on whiteboards, posters and miniature models. Three types of artifacts are identified: design artifacts, procedural artifacts and motivational artifacts. (Perry, Sanderson 1998, Olson et al. 1998)

Design artifacts are plans, models, prototypes, or visualizations of a problem (Perry, Sanderson 1998). In software projects, these artifacts could be use cases, code interfaces, module designs and such. Artifacts that concern the process itself are called procedural artifacts (Perry, Sanderson 1998). They include forms, change requests, office memos, schedules and Gantt charts. The author perceives that procedural artifacts in software projects most likely do not distinguish much from the procedural artifacts in other projects, since process issues such as to-do lists and deadlines are common to most projects. Motivational artifacts are the items that promote team spirit and motivate team members, such as photographs of the workgroup or company logos. (Olson et al. 1998)

When a team is working together in a physical workspace, the artifacts are constantly visible to everyone. In addition to this, it makes editing, clustering, and moving the artifacts possible for all team members at all times. The editing of artifacts is a source of information as well, as it increases the knowledge about the content of the artifact, as well as about the editor. This is called feedthrough. (Olson et al. 1998, Gutwin, Greenberg 2002)

2.4.4 Social identification

The social identity theory (Tajfel, Turner 1986) concerns intergroup behavior. It suggests that a person creates a feeling of belonging to a work-group through categorization (of people), identification (association to groups), comparison (with other groups) and psychological distinctiveness (we want to be distinct and positively compared with other groups). Hinds et al. (2002) present, that social identification to a work-group is necessary for people to cooperate well in distributed work settings. Hakonen et al. (2007) state, that geographical boundaries form the essential identification boundaries.
2.5 Summary

The three discourses derived from the research questions were software development, awareness, and physical workspace.

Software development is a distinct field of engineering, which can be characterized by scale, uncertainty, interdependence, informal communication, major difference in programmer productivity, and as thinking and communication intensive activity. In the past decades, agile practices have evolved from the previously rigorous plan driven software development process, and at the same time, software development has become increasingly distributed. The emerging practice of radical collocation is promising in increasing the productivity and work satisfaction of the team members in software projects.

Awareness has been a widely but incoherently studied issue, leading in ambiguous results. Studies concerning awareness in software development focus on tools providing awareness of two kinds: Group awareness and the awareness of relationships. The workspace awareness framework illustrates the content, creation, and usage of awareness in a common workspace.

Physical workspace should be the space designed to fulfill the needs of the people and the work process, and recent evidence shows that open offices do not fill those need very well, but instead attention is often given to the aesthetics. The close proximity can increase the awareness of the people working in it by enriching communication, and by enabling the use of common project artefacts. In addition to this, social identification is strong with people who are located physically close to you.

The up-to-date knowledge about a software project, which is necessary for successful coordination and collaboration, can be summarized from the literature review. The components of awareness are presented below in question form.

Who am I working with?

In terms of coordination and collaboration, this is the question that needs to be answered before many others can be presented. We must know the actors that need to coordinate, and with whom we can collaborate.
What are the skills and knowledge of others?
The skills and knowledge of others working with us are important in three ways. First, to coordinate work well, we must know who should do what work, meaning who has the skills and knowledge best suited to do the correspondent work. Secondly, to ask for assistance, we must know who to contact. Thirdly, we must know when someone’s knowledge is outdated in order to help him.

What is the activity of others?
Knowing on what someone else is currently working enables us to assist him in his work. In addition to this, if the person is not present, we will know when to expect him. This eases the decision when selecting a source of information for something that we need, and knowing that the person not present is most likely not up-to-date on all issues.

What is the availability of others?
We must know if a person we need to communicate with is currently available, meaning that we know if the communication will disturb and interrupt the work of the person we are trying to contact. This is necessary for collaboration to be productive, instead of counter-productive.

What are the relations between work objects and people?
When we are working on a work object (e.g. a code file), we must let the person responsible for it know what we are doing. In addition, the person is often someone who knows about the current object more than we do, so he can provide assistance when necessary. Furthermore, if our work on a work object influences other work objects, we must let the people responsible for them and people currently working on them know. For all of this to work, we must know the relations between work objects and people.

What are the practices that need to be followed?
Some software development practices can be effective in increasing the awareness of team members, such as the daily scrum meetings (Schwaber 2007). The team members must know what practices to follow, in order to collaborate and coordinate well.

The components of awareness in software projects are presented in Figure 7.
Figure 7: Components of awareness in radically collocated software projects

<table>
<thead>
<tr>
<th>Awareness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Who am I working with?</td>
</tr>
<tr>
<td>What are the skills and knowledge of others?</td>
</tr>
<tr>
<td>What is the activity of others?</td>
</tr>
<tr>
<td>What is the availability of others?</td>
</tr>
<tr>
<td>What are the relations between work objects and people?</td>
</tr>
<tr>
<td>What are the practices that need to followed?</td>
</tr>
</tbody>
</table>
Research methodology

This chapter describes the different phases of the research, and the research methods used in this thesis. The research consisted of three phases, and is done as a qualitative study. The first phase was a literature review, which was the basis for observations and the interviews in the qualitative (Miles, Huberman 1984) studies, which consisted of the pilot study and the actual case study.

3.1 The pilot study

An exploratory pilot study was performed to gain deeper understanding on the radical collocation and its relation to the awareness of the group members, and to validate the appropriateness of the research methods to be used in the actual study. The exploratory study consisted of an unstructured observation (Somekh, Lewin 2005) of the team working in the radically collocated workspace and an unstructured interview (Gillham 2005) of the project manager of the team.

3.1.1 Overview of the pilot group

The pilot group consisted of a radically collocated team of 6 members, who were working full-time for a month developing an existing software product further. The members of the team were students from Helsinki University of Technology with varying software development experience, and they were employed through a course, and they were not employees who had signed a contract. The team consisted of the project manager, the quality assurance manager, the architect and three developers all of which tested their own components. In addition to this, one of them practiced as the main tester, who did most of the integration testing. The team used Scrum (Schwaber 1995) software development process with one week long sprints.

3.1.2 Observation

The observation was performed in two 3 hour long observation sessions. The purpose of the observation was to monitor how the team worked together and to examine the workspace in use. In practice, this was identifying factors in the workspace that created limitations or possibilities for creating or using awareness. In addition to this, communication between team members was examined categorizing roughly by its purpose, content, participants and effect. All findings that were considered relevant by the author were documented.
3.1.3 Interview

The project manager of the team was interviewed to gain deeper understanding of the software process in the form of tools and practices used, how the workspace was used in the project, which elements of it are considered important and why, and to find out what elements in the workspace could be arranged better. The project manager's view on which kinds of awareness components are necessary was consulted as well. The interview was conducted as an unstructured interview (Gillham 2005).

3.1.4 Limitations

The software development was very homogenous, as all pilot study members were male students from Helsinki University of Technology. This can result in missing problems that might be more easily created by a more heterogeneous group, e.g. the lack of privacy. The fact that the team members were not real employees, and they had very little pressure to complete all of the work on schedule might have also resulted in a biased, more relaxed atmosphere than in a commercial software project.

The results of the pilot study are presented in 5.1 the pilot study results.
3.2 The case study

The main part of the research is the case study. This sub-chapter presents an overview of the case project and describes what the goal of the observations and interviews was and how they were executed.

3.2.1 Overview of the case project

The case company is a large software development company in Finland. The case project is a six month long effort to produce a Lotus Forms based web service application for an external client. This project is a successor to a precursor project, which goal was to produce a working environment and single working web-form that was integrated to the databases. The case project product will be a fully functional, easily usable, secure and clustered web-service environment for the client.

The project is part of a project family, which has around 20 people working on it. All the projects in the project family have the same client, and some overlapping functions. Currently, there is a quasi-radically collocated workspace in use for the project family, which means that there is a part of the open office which is formally dedicated for their use, and which doesn’t have anyone else working next to them in a 5 meter radius. The project family is due to be moved in to a radically collocated workspace (a project room) in the following months.

The project family is staffed with a project executive, two project managers, two IT architects, a hardware specialist, two computer mechanics, and ten developers. The project executive’s task is to assist and mentor the project manager, and supervise the whole software development team. The project manager takes care of the management side of the project, such as resourcing and scheduling. The IT architects handle the technical design, specifications and decisions. The hardware specialist assists the computer mechanics in their installation tasks and the developers in their hardware related issues, whereas the developers take care of the actual implementation. The customer relations are taken care mainly by the project executive and the project manager, but other project members occasionally attain customer meetings.
3.2.2 Observation

The purpose of the observations was to gain understanding of the case project family workspace and how the people work in it. The observation was unstructured (Somekh, Lewin 2005), but two questions were given extra focus: how does the team function as a whole in the workspace, and how the workspace itself prohibits or contributes to individual or team work. The observation was conducted by two observers, who observed simultaneously and both did individual notes. These notes were later combined so that no communication events were left unnoticed.

All relevant findings were documented, and a communication log was used to keep track of communication. Its contents were: initiated by, participants, the start time, content, duration and possible effect.

The communication log form can be found in the appendix.

3.2.3 Interviews

The goal of the interviews was to understand the personal needs of the team members and the software processes used in the case projects. The needs and the processes were then reflected to the workspaces in use. The interviews were semi-structured (Gillham 2005). There were in total 8 interviews, six of the interviewees currently worked in the dedicated workspace full time, one was working there around 2 days a week, and an foreign employee who did not speak Finnish and had worked in the dedicated workspace for two months, and later continued the work in remote settings was interviewed. The first seven interviews were conducted face-to-face, and the last one was done via telephone.

The interview consisted of six underlying topics: background information, workspaces and their usage, software processes, software development as collaborative work, awareness and radical collocation.

The background information was gathered while opening the discussion, after which the focus was shifted to the workspaces which were used in the project. The questions included what spaces were used, why, by whom, how often, and if the spaces were shared with other projects. Next, the software process in use in the project was examined, focusing on issues like job partition, work distribution, used practices, perceived uncertainty in the project and communication within the team.
or project. The nature of software development as collaborative work is very relevant to the interview, as it encompasses what is should be expected from the workspace or workspaces. If the software development were perceived as work that should be done in solitary, there would be little need for common workspaces. If software development is perceived as collaborative work, then the topic awareness becomes important. This part of the interview tried to decipher what information the interviewed considered important for successful collaboration and coordination. Finally, the focus is shifted to the radically collocated workspace. The presented questions included questions like: Why do you use radical collocation, does the space meet the reason, how do people function in the radically collocated workspace, what’s good in it, what’s bad in it, how it could be improved, and the personal opinion of the interviewed about the setting.

The complete list of planned interview questions can be found in the appendix.

3.2.4 Limitations
The workspace in use was a dedicated part of an open-office. Even though the distances to people that are not related to the project was more than 5 meters, the company workspace usage protocol was that no work related material was to be left without supervision. This setting can be considered a quasi-radically collocated workspace, and observing it provided mostly information on how the setting influenced communication, as the workspace was not actively used in other kinds of information sharing, such as through project artifacts. This issue was covered in the interviews.

All the interviewees were people that spent considerable time in the dedicated workspace, and none of the people who worked in the project but only visited the dedicated workspace were interviewed. This is a limitation because the reasons why these people choose not to work in the dedicated workspace are not known.

The results of the case study are presented in 5.2 the case study results.

3.3 Analysis of the gathered data
The goal of the data analysis was to identify the pros and cons of the existing workspaces, the desired features of an ideal workspace, and to understand the causality and the impact of the
different factors, such as how possible eye contact or background noise affects communication, and what elements contribute or depress these factors.

After the interviews and the observations were completed, the data analysis began by grouping the statements done by interviewees and the observations under their corresponding topics (awareness, the dedicated workspace, communication, social identification, experience on previous workspaces, tools). If multiple interviewees concurred on an issue, a multiplier was given to that statement. If interviewees disagreed on an issue, it was marked as contradictory, and therefore worth giving more attention in terms of the reasons why it is so.

The observations were analyzed by going through the communication log, indentifying which communication event were work related and which were not, and then adding up the communication events to acquire a total count for both communication types. Furthermore, the amount of communication occurring in close proximity, meaning between people sitting next to each other, was compared to communication occurring between people who are separated by distance. If something happening in the workspace was considered relevant, it was noted and discussed between the observers later on if it really was relevant. In addition to this, other workspace related observations were reflected to the data from the interviews. In practice, this meant finding support or contradiction to the claims made in the interviews.
4 Results

This chapter presents the pilot study results and the case study results, which are derived from the observations and the interviews.

4.1 The pilot study results

This sub-chapter describes the pilot study radically collocated workspace and its usage. In addition to this, the communication patterns and ad-hoc pair and group programming findings are discussed.

4.1.1 The radically collocated workspace

The workspace used in the pilot study project was a roughly 5m * 6m room with two doors and two windows. The equipment in the room were six desktop computers, a whiteboard, a flap board, a projector, a projector screen, a printer, plenty of post-it notes, a microwave oven and a coffee maker. The room layout is presented in Figure 8.

![Figure 8: The pilot study workspace layout](image)

At the main working setup, everyone worked independently on their personal workstation, but the workspace had multiple aspects that were used when needed. The whiteboard provided space for design and notes that were persistent, meaning that they were accessed every once a while, and not
wiped out until they became unnecessary. The flap board was used for throw away design, such as finding a solution for a problem at hand, or discussing code format with the whole team. Post-it notes were used as a bug tracking system in one of the windows, enabling the view of known bugs and their state, as well as assigning them to individuals. Post-its were also used as reminders throughout the workspace. The projector was used to demo the progress of the work to the customers and an occasional code review, but it was not used for anything else.

4.1.2 The communication patterns

There were no formal communication protocols in the project; the only request made by the project manager was to take phone calls outside the workspace. Nevertheless, communication in the workspace was nearly completely limited to the work, and very little non-work related discussions took place. Team members used headphones if they wanted to concentrate.

The observation revealed that even in a radically collocated workspace, people communicate more with people next to them. This was most likely amplified by the fact that the computer monitors somewhat obscured the view to the other side of the table.

Most of the communication in the workspace was fast paced, usually a question directed to an individual or everyone in general, followed by a brief answer. When a question was presented, which could not be answered unanimously, a discussion followed. Any team member could join these discussions and give their opinion. When a consensus was found, the work continued. These discussions could take place when everyone was at their workstation, or by gathering around a single workstation, whiteboard or flap board. Every once a while multiple discussions did occur at the same time, and this did not cause any problems.

Sometimes communication was not initiated with a question, but with an advice, a piece of information that is considered relevant to someone, or an answer to a question that was made earlier. This created an assisting atmosphere, but according to the project manager, some requests for assistance were not done as soon as they could’ve been done.

The daily scrum meetings (Schwaber 2007) were required by the customer and therefore held, but very briefly, since the team considered them useless, because the content of them (what did
everyone do yesterday and what will they do today) was already known to all. The meetings were necessary only when someone had been absent, and an update for him was in place.

4.1.3 Ad-hoc pair and group programming

The method of pair programming which was described in 2.2.3 Agile software development was used in the project as well, but there was no formal pair programming in the pilot project, instead the team members paired up when they saw it fit. Often this was done after a "show me how it's done." request from a teammate, or because the situation was easier to clarify to others by showing it on the screen than by discussing about it. Sometimes this created situations where three or more people were gathered on the same computer, discussing the implementation when one of them was implementing it.

The pair or group programming took a variable amount of time, and the pair broke up when it was no longer seen necessary. The need for pair programming was perceived higher in difficult tasks and lower or none in routine tasks.
4.2 The case study results

This sub-chapter describes the case study dedicated workspace and the data gathered by observing it. In addition to this, previous experiences on radically collocated workspace, opinions of the current dedicated workspace, the usage of Sametime and expectations from the upcoming radically collocated workspace are presented.

4.2.1 The dedicated workspace

The workspace used in the case study project was a roughly 10m * 10m part of an open office plan, which was dedicated to the use of the project family, and all the nearby work posts around the dedicated area were empty in a roughly 5m radius. This section was separated from the rest of the open office with 175 cm high screens from three sides, and with 140 cm high cupboards from one side. The screens between the work posts inside the dedicated area were 125 cm high, just below the eye level of a person sitting down, so that people could see each other. The cupboards were next to an aisle which lead to a few meeting rooms. This aisle was used every fifteen minutes or so by people not related to the project family, which caused some external noise, but otherwise the dedicated workspace was shielded from noise that was not caused by the project family members.

Each work post was equipped with a table, an extension cord power supply, an LAN connection and a fixed-line telephone, which were not used during the observation. A few work posts had extra equipment such as additional table lights. The company policy stated that no equipment was to be left in the work post overnight, but a few extra keyboards, mice and laptop support racks were left here and there. These were considered personal equipment by the project family members.

There were in total 20 work posts in the observed workspace, and 12 people sitting in there for most of the observation. In addition to this, 4 people visited the workspace for less than 20 minutes each. They communicated with one to three people on their visit. Two work posts were manned by the observers. The room layout and the location of the observed and the observers are presented in Figure 9.
The dedicated workspace was observed for 207 minutes, and in that time 65 conversations took place. The duration of the discussions is presented in Table 3, and the amount of discussions per person is presented in Figure 10. 57 of the 65 conversations were related to work issues, and those 8 which weren’t, all lasted less than 3 minutes. Out of the 65 conversations, 61 were engaged with people sitting next to each other (meaning that there was no-one sitting in between their direct line of sight, for example Observed 5 and Observed 2 were considered neighbors. See Figure 9 for details).
Table 3: Communication by duration

<table>
<thead>
<tr>
<th>Duration</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discussions that lasted under 1 minutes</td>
<td>31</td>
</tr>
<tr>
<td>Discussions that lasted 1-3 minutes</td>
<td>14</td>
</tr>
<tr>
<td>Discussions that lasted 3-5 minutes</td>
<td>9</td>
</tr>
<tr>
<td>Discussions that lasted 5-10 minutes</td>
<td>7</td>
</tr>
<tr>
<td>Discussions that lasted 10-20 minutes</td>
<td>4</td>
</tr>
</tbody>
</table>

In the observation period, screen sharing (a person moved to someone else's work post to see his/her display) took place 20 times, on three occasions Observed 7 lifted her laptop over the screen for Observed 6 to see. Out of the observed 207 minutes, 50 minutes were silent and for the rest of the observation period at least two people were discussing. In eight occasions, a two tier discussion turned in to a three tier discussion by someone joining in through overhearing. The amount of communication per minute is presented in Figure 11.
When one of two people that didn't share constant eye contact needed to communicate with the other one, this was easier and done more often when the one who was initiating communication would turn around in his/her chair to gain eye contact (for example Observed 8 initiating communication to Observed 7), than when the initiator could only gain eye contact by shouting or walking to the other person (for example Observed 5 initiating communication to Observed 2).

Cell phones were used 10 times in the observation period, and in most occasions (7 out of 10) this caused no disturbance, because in these cases the ringtone was set to low volume and the people managed to keep their voice down while talking to it. One person who was somewhat older than the rest of the group was as an exception, as he kept his cell phone ringtone on a high volume, and he talked with a high volume to it. Despair could be seen in the faces of the other group members when he spoke to his phone, and for the duration of his phone calls, there were no discussions taking place in the workspace.

4.2.2 Previous experiences on a radically collocated workspace

Two interviewees had previous experience on a radically collocated workspace, one of them had worked in a radically collocated workspace for three years, and the other was in the same workspace for a year or so. The radically collocated workspace was manned by around 20 people, and despite the crowdedness and the inefficient air conditioning that was in use in the workspace then, their overall view of a radically collocated setting was positive. Positive factors identified
were social identification and team spirit, enhancing development through the ease of communication and the ease of use of the workspace, and for possibly easing customer interaction. The only negative aspect identified about the radically collocated workspace was the fact that it isolates the people in it from the people who are outside it.

The social identification and team spirit generated in a radically collocated workspace was perceived very important by both of the interviewees, and when asked with whom outside the current project family they communicate most, they indentified the people from their previous project. One of the interviewees described this as follows:

"We always had lunch together; I didn’t know the people sitting in fifth floor or elsewhere at all. I think that was a good thing, because otherwise there won’t be any team spirit in work like this."

The effect of radical collocation on the work efficiency was positive according to both interviewees, the first perceived the ease of use of the workspace as the main aggregate of efficiency:

"Working there was incredibly easy, the workstation was always ready, everything worked and was ready, and everyone related to project were sitting in the same room. It worked incredibly well."

The other interviewee saw that the ease of communication was the most important factor in enhancing efficiency, this applied to the previous radically collocated workspace, as well as the current dedicated workspace:

"It hastens the development, when you can just turn around in your chair and ask "What do you think about this?" The issues can be handled immediately, and the communication works considerably better when everyone is in the same workspace."

One of the interviewees, who had worked in a radically collocated workspace for three years, had seen customers visit the project room many times. He saw that this seemed to work well according to the customer reactions, but could not specify exactly why:
"I think that in the viewpoint of the customer, it’s nice to have a space where you can go to view a demonstration and things like that."

The sole negative factor about a radically collocated workspace identified was the isolation from people outside the project group that occurred in it, this was not perceived as a very serious issue by one of the interviewees (as he had worked longer in the project room, and since the project had grown all that time, he had met a lot of new people), but the other one had serious concerns about it:

"The bad thing is that you spend time mostly with the project group, and don’t communicate with outsiders. That might not be a bad thing considering the project, but if you think about your own career and future projects, that might be a bad thing."

4.2.3 Opinions on the current dedicated workspace

All of the interviewees worked now or had previously worked in the dedicated workspace, and they all had their opinion about it. All the interviewees except one perceived the dedicated workspace as a better working environment than a private office or the non-dedicated open office that was used for most projects in the case company. The one interviewee that perceived small three people offices better than the dedicated workspace was the foreign employee, who saw that one can concentrate better in a small office. The reasoning behind the support for the dedicated workspace was that in software engineering projects you need to communicate, and communication is easier when everyone is nearby:

"If the other one is sitting there two meters away, it’s very easy to turn to him and say "Hey, can you take a look at this", or have you done this or that. If the person is not next to you, those questions are very easily left unasked. And then people make their own decisions and a week or a month later it turns out that by the way we should have talked about that then."

The abundance of communication was seen as a prerequisite for mutual understanding:

"That would become easier too if people were in the same workspace, it would be quicker to achieve mutual understanding if happens something like that, meaning that someone is doing something that doesn’t fit to the overall work at all."
Abundance of communication means abundance of noise. When asked about the noise and how distracting it is, all but one considered the noise level in the dedicated workspace was not distracting. The one who saw it somewhat distracting was the one who had three years of experience in the radically collocated workspace, and he was the only one to block the noise by listening to music via headphones every once a while. The others said that they have gotten used to the noise, and that it's actually a good thing sometimes:

“There are a lot of projects around this current project, all of them are not directly related to each other, but in any case the customer is the same and we discuss a lot about overlapping and similar issues, in a way you hear someone talk about a solution, and then it’s really easy to say “Wait a minute, we are doing something like this too, is this the same thing?” In a way the amount of overlapping work might decrease quite a lot.”

When inquired further to the issue on what kind of noise is really distracting and what is not, the interviewees saw that hearing work related issues is beneficial, but hearing non-work related is issues distracting, but the amount of non-work related chatter was perceived so little that it wasn’t a major problem. In addition to this, all noise which was not related to one’s own work was perceived distracting as well, but in the dedicated workspace there really weren’t much of that kind, except for people passing by. When asked about which one is worse, external or internal noise, the two of the interviewees answered:

“Maybe of course the external noise, usually project members discuss about something related to the project, and that doesn’t really disturb in a way when you hear what the other one is planning there. If someone talks about a completely different thing, a completely different project, then you really don’t get it.”

“I think that, especially now when the team is so large, it’s a good thing that we’re there [in the dedicated workspace]. In any case when two people are talking, the rest of the people around hear it too. At least it doesn’t bother me to hear what’s happening in the other projects for the same customer. But when there’s the situation that there are people from different projects, it might distract more, when people are talking about a project that doesn’t interest anyone else, and it only distracts.”

Social identification was perceived as one of the most important things about the dedicated workspace by two interviewees, both of which were female. Knowing someone at a personal level was seen as something that was very important. According to both of these interviewees it lowered the required incentive to ask something:

"The better you get along with someone, the more you will discuss with him about work issues. If you don't know somebody well, you will only ask questions from then when you have to, or it is really necessary."

The dedicated workspace was seen as a catalyst for getting to know a person:

Of course you can’t avoid getting to know everyone in a more personal level, which in my opinion is solely a good thing, because that makes people stick in your mind, and afterwards you can contact them more easily in work related issues, or just to get lunch or coffee. Of course it increases there [in the dedicated workspace]."

When inquired about how long it takes to get to know people in the dedicated workspace, they estimated that one month is the minimum, but the longer the better you know them. The project manager also considered one month as the minimum time it takes for a team to start to collaborate efficiently. There were a few employees that were working for a project that was in the project family, but did not spend much of their time in the dedicated workspace, since they were not full time employees of the project family. In the experience of the project manager, contacts to these people have to be forced:

"I think that they are formed by half-forcing them, I invite them to team meetings and they do show up and we can go over things."

The team members who worked in the dedicated workspace did not socially identify themselves strongly with the people who stopped by every once a while:

"The people who just stop by every once a while, you don't get lunch with them as much, and at least I don't know those people that well."
At least in the viewpoint of the project manager, this is a problem, since the project manager needs to know the people she is working with, in terms of skills, knowledge, and trust:

"And especially when I don’t know them as a person, I have trouble knowing if they are people who will contact me when there’s an issue that needs to be addressed, or do I have to fear they might hide a big problem for months... When you know a person, you know on what level his work needs to be interfered with."

All interviewees agreed that if a work task requires high level of concentration, it’s better to work home and most of the interviewees did that every once a while. Occasionally the team members did work that required concentration in the dedicated workspace, but even though there were quiet rooms next to the dedicated workspace, no-one ever used them for concentration-critical work. Instead they were used for making telephone calls or ad hoc two person meetings. Still, some concentration critical work was done occasionally in the dedicated workspace, and it wasn’t considered impossible:

"It [the dedicated workspace] does have its downsides, when you have to do work that requires concentration its very distracting when someone is asking something constantly, but if you put your Sametime status on “do not disturb” and reply to people who are asking, that I’ll get back to you later, you won’t be bothered."

The use of Sametime will be covered later under 5.2.4 the usage of Sametime.

Awareness was a topic that was inquired from the interviewees by questions such as “By what extent you need to know what others in the project are doing?” and “To what extent others in the project need to know what you are doing?” All interviewees agreed that it is beneficial to know what people are doing on a general level:

"Some kind of conception sticks in the back of your head, which you can someday make use of. “Hey, I remember John Doe thought of a solution to that same thing” comes back to you from somewhere."

The project manager saw that it is very important for her to be in the dedicated workspace, and didn’t know how she would keep up with everything if she wasn’t there:
“I feel that when I’m there, I’m more aware of what’s happening and people contact me more easily than when sending email.”

The physical settings of the workspace were considered adequate by most of the interviewees, while one considered the aesthetics of the dedicated workspace too ugly, and two saw that it could be more ergonomic in terms of chairs, bigger displays etc. The screens between different work posts were contradictory, some saw them as completely unnecessary, some saw that they should be higher and some saw that they are fine the way they are.

The one project member that took aesthetics seriously said he personally enjoyed coming to work in his previous workplace because of the modern workspaces, and if the workspaces at the current workplace would get any worse, he would leave the company. He stated that:

“There has to be quite fat bonuses so that one can endure working in these workspaces.”

4.2.4 Instant messaging

The case company protocol is that all work can be done from anywhere, and because of that, Lotus Sametime is used by all of the employees. Sametime is an instant messaging tool for corporate use, and the interviewees used it mainly for communicating with people who are not nearby. The experiences were very positive, and the interviewees preferred it above email, and some team members even used it to communicate with people next to them, when they were talking about personal issues which they did not want everyone else to hear. Even still Sametime was considered very useful, face-to-face communication was perceived better:

“Of course when you have all of your friends sit next to you, communication is much easier. Of course we have these collaboration tools, like chat, which you use all the time so it doesn’t really matter where you sit, but it is way faster when you can ask directly.”
The problem in the usage of Sametime was identified by all of the interviewees: screen sharing was practically impossible. When asked “What kind of things in this project wouldn’t you even try to discuss over Sametime?” an interviewee answered:

"Most likely it would be related to something visual, like “Look, here is this element.” or something like “Click here.”"

In the occasions, the interviewees agreed that it’s best to visit the person physically, if possible. The interviewees had each other as contacts in Sametime, but in addition to this, they had contacts from previous projects. They used Sametime most with people with whom they are currently working with, but the people with whom they communicated most after that were the people from the previous projects with whom they socially identified themselves most with, and in most cases those were the people with whom they had previous shared a workspace.

4.2.5 Expectations about the upcoming radically collocated workspace

As mentioned before, this project family was due to be moved in to radically collocated workspace in the next few months. Some of the interviewees were aware of this and others were not, but when asked what they would want or expect from the new workspace, following topics came up: The concept of centre of excellence, the usage of flap boards etc. for information sharing, dedicated meeting rooms, dedicated hardware, aesthetics and improved job performance.

A general expectation of the new radically collocated workspace was that it would be a centre of excellence. Centre of excellence was the concept in the company, and the method of radical collocation was part of that concept. The radically collocated workspace would be a physical location, which would be known to most people working in the building. This makes it possible to just to go there and ask about things related to the project, even if the person asking does not know exactly who has the information he or she needs:

"It would be an actual physical place where you can go like “Hey, there’s a place where people know about things.” That might be easier for people, because now all the people circulate around and no-one knows who knows what or on what they are working."
"I feel that the expertise kind of concentrates somewhere. In a way it’s just a space, but it kind of gives you the feeling that it’s a centre of expertise for something."

Another issue that was anticipated by the project manager and the IT architect were the usage of flap boards, whiteboards and such. Unlike to the dedicated workspace now in use, where leaving anything in the workspace overnight was prohibited by the company policy, it would be possible to decorate the radically collocated workspace as one wanted:

"Yes, and of course it would be a good thing if it were a space that is dedicated to the project, and the people who are not related to it would be excluded. And you could put pictures on the wall, flap boards on view and put the things related to the project to use."

Whereas most of the interviewees saw flap boards and whiteboards as equipment that will be handy for sketching in ad-hoc meetings (currently when sketching is needed, it is done on a piece of paper). The project manager had different plans for the whiteboards; she had previously tried to set up an online bulletin board for the project, which she has now given up on. She described the experiences on the online bulletin board and why a physical one would be better as follows:

"It just doesn’t work. People won’t remember to check it out, but if they walk past the table which describes that who’s doing what on the wall every morning when they come to work, or every day when they move around, you would look at it."

The IT architect had a different slightly different approach. He saw that the walls should be filled with high level architecture designs, and rationalized this with the need for understanding the software as a whole:

"In my opinion the big picture about what we are working with should definitely not be understated. If we bring a so called programmer or a developer to the project and he is given a task like “Code this part.”, then in my opinion it is vital to understand where that piece of code is located, what’s around it and what’s the context."

The project manager was the one who had most problems with the meeting rooms, so she desired dedicated meeting rooms inside or in the vicinity of the radically collocated workspace. She was
that one to two small rooms would be adequate, and that a corner with a few sofas and a small table could be used for these purposes as well, and that could be used for relaxing purposes as well. The meeting rooms could be used for all sorts of meetings, such as regular customer meetings or project status meetings.

Better hardware was a common request, dedicated tabletops for everyone was one of things that was perceived very important as the laptops used by the developers had small displays and too little performance. In addition to this, the project family could have dedicated server machines, which would be freely customizable by the project team. At the moment, the project family had none, and installing anything on an external machine was practically impossible. The request for a projector came up with two interviewees, they saw that a projector is something that you very often need, but too seldom is available, as the current company policy is that you have to get them from a specified place and sign them out for the period you need them.

Aesthetics was something that was only perceived important by two of the eight interviewees, and one of them only referred to the décor being 80’s style, and that a better decoration would make spending time at work more comfortable. The other one perceived envisioned the workspace would be a place where you would want to come, and this would be achieved through matching furniture, lighting and colors. He expected the job performance of the team to increase through improving the workspace, because he saw that the workspace is an important motivator to the team. He concluded the issue about an enjoyable workplace as follows:

"Let's say that there are 3 million people that work in Finland. Let's say that one and a half million have to go to work, the other one and a half million get to go to work."
5 Discussion

This chapter answers the three research questions by discussing the further implications of the results. These are presented under the topics **the effect of radical collocation on awareness**, **the communication cycle**, **maximizing the benefits of radical collocation in software projects**, **individual learning versus team learning** and **forcing awareness**.

5.1 The effect of radical collocation on awareness

This sub-chapter answers the first research question *"How does radical collocation increase awareness in software projects?"*

Two main reasons why radical collocation increases awareness can be identified and these are the abundance of informal communication in radically collocated settings and strong social identification to the radically collocated team. It is difficult to say which one comes first, since it seems as social identification is increased through communication, and communication is increased through social identification. Nevertheless, both are an important factor in increasing awareness, since abundance of communication is necessary for a high level of awareness, and it seems that social identification is necessary for abundance of communication. A radically collocated setting in turn seems to be a strong catalyst for social identification. This causality chain is presented in Figure 12.

Figure 12: The causality chain between radical collocation and high level of awareness

The results also suggest that after social identification is achieved, radical collocation is no longer a prerequisite for abundance of communication, but it still is very efficient in increasing awareness, because it enables the overhearing effect to work, which was perceived very important by the interviewees. Radical collocation also blocks the external (not coming from project members) noise, so it further enhances the overhearing. In addition to this, radical collocation enables the use of project artifacts as a method of increasing awareness. The results are not clear on the issue of a how
long period of radical collocation is necessary to achieve social identification and what other factors than collocation influence it, but the minimum length of the period is estimated to be one month, and the longer it is, the stronger the social identification is.

5.2 The communication cycle

This sub-chapter answers the second research question "How does radical collocation enhance coordination and collaboration in software projects?" by presenting the communication cycle, which describes how the communication actually takes place and how information is transferred in the radically collocated workspace. Two distinct communication patterns were identified through observing the workspaces. These are the information pull and the information push patterns. The actual conversations taking place can be described as a communication cycle, where every team member can be the recipient or the transmitter of information. The communication cycle is presented in Figure 13.

![Communication Cycle Diagram](image-link)

**Figure 13: The communication cycle**
5.2.1 Information pull

Information pull is information transfer, which is initiated by the recipient of the transferred information (Cybenko, Brewington 1998). A simple question “What is the time?” presented to anyone is an example of information pull. The information pull process in a radically collocated software project is presented in Figure 14.

![Information pull diagram](image)

**Figure 14: Information pull**

**Do I need information?**

This is the question that enables information pull. We somehow become aware that our work cannot proceed until the necessary information is acquired, or we find out that our information in some subject is insufficient or outdated. In a software project, this piece of information could be a function call that we do not remember, if all components are ready for a build, or just simply what to next, when all the tasks assigned to you are completed.
Do I know where can I get the information I need?

To acquire the information we need, we must know where to look. We must take to account that there may be multiple sources of information that can provide the piece of information we need. For example, we can search for a function call in the software development environment, read it from a manual, or ask a team member. When there are multiple sources of information, we can choose the most suited information source for this situation, this being the fastest way to acquire the information, the most reliable source of information, or some other criteria. If we do not know where to get the information we need, we must query someone else for that information, or try to find out that information in any way we can, such as using online search engines. Note that querying a person will disturb him, if he is not available.

Is the person holding the information present and available?

Consulting a team member is often the fastest way to acquire information, but it can also be the only source. If a team member is our selected choice of information, we must take account if the team member is present and if he should be disturbed at the moment or not. If so, we can proceed to ask for the information we need. If he is not present, we must wait or select another source of information (if one exists). Disturbing a colleague, who is not available (meaning that he is focused in his own work) is always a tradeoff between your work and his own work – the option to wait can also be used when the colleague is unavailable.

5.2.2 Information push

Information push is information transfer, which is initiated without a request from the recipient of the transferred information (Cybenko, Brewington 1998). Telling someone that he has dropped his wallet without noticing it is an example of information push. Information push is either initiated when we realize that we have information that someone needs, or when we realize that someone needs information that we have. The information push process in a radically collocated software project is presented in Figure 15.
Do I have information that someone needs?

Information push can be initiated by becoming aware of someone's need for information, which we possess. If we become aware of the need through a request of information (pull), we answer it by providing the information (push). We can also become aware of a person's needs through other methods than requests, such as knowing that someone will start on working on something that we have deeper knowledge of – it is possible to know what people need, even if they don't know it themselves.

Does someone need information I have?

Information push can also be initiated by acquiring information, which we know someone needs. In these cases, the need becomes before the information, meaning that we discover something that someone has previously needed, or the need becomes clear when we acquire the information. In a software project, this could happen e.g. in a code review. When we notice an error, we can either fix it by ourselves or tell the original programmer where the mistake is, what's wrong with it and how
it should be fixed. In this case we acquire the information about the mistake, and the need that becomes clear is that someone can learn and not to redo the mistake in the future.

If the answer to either “Do I have information that someone needs?” or “Does someone need information I have?” is positive, we must consider the availability of the person needing the information the same way as in information pull, before proceeding to communicate.

5.3 Maximizing the benefits of radical collocation in software projects

This chapter answers the third research question “How should the physical workspace be arranged to maximize the benefits of radical collocation in software projects?” by discussing how the workspace should be arranged, and what else should be taken into consideration when trying to maximize the benefits of radical collocation. Three improvement suggestions can be derived from the pilot study and the case study: Display projection, re-planning seating and using communication protocols.

5.3.1 Display projection

Observing what others are doing is important according to the 2.3.3 Workspace awareness framework. In software development, one can’t really see what other people are working on, unless you go behind them and look at their display. On the other hand, in the pilot study radically collocated workspace a projector did exist, and one was expected of the future case study radically collocated workspace. The projector (or possibly a widescreen display) could be used in projecting displays for others to see, this could happen either on purpose or ad hoc.

Purposeful display projection would enable easy screen sharing with others, meaning that if one had for example a question related to the code he is working with, there would be no need for everyone to gather at his computer, but he could project the display to the wall for everyone to see. This would require very light transition (preferably one click) between the different displays, because switching cables is more effort than gathering around a computer.

Ad hoc display projection would mean that when no-one would want to project their own screen, everyone’s screen would be projected in turn for a minute or so. This would create a better understanding of what everyone else is doing currently, and further facilitate the 6.1.2 information
push mechanism. If an ad hoc projection would be used, privacy issues should be considered as well, allowing people to disable their display sharing at will.

One other potential usage for a projector could be projecting the awareness tools which were discussed under the topic 2.3.2 Awareness and software development. In particular awareness tools which try to provide an overview of the software under development could be beneficial, since any team member could see the recent changes or any other information that is considered important with a glance.

### 5.3.2 Seating

The seating order in the pilot study and the case study were static, meaning that people sat in the same seat in practice every day, even if they could have sat wherever they wanted. When this is combined with the observation that people communicate more with people currently next or visible to them, an improvement can be suggested. The seating order must be dynamic; so that you can choose to sit next to people whose work is most is related to yours, who you perceive you have most need to communicate with, who you think you can learn most, or some other criteria. In addition to this, it may be wise for the project manager to encourage this, as it is possible that people tend to stay in the seat they have familiarized themselves with. This situation can change daily, so people need to be able to logon to their personal profile from any computer. Even though this was not an issue in the case study, as people worked on their laptops, it should be taken in to account if laptops are not used.

Another suggestion can be derived from the observation that people communicate more with people visible to them: All team members should be visible to each other in all the time, or the team should be split to groups that need to see each other. This is possible with a round table layout, which is presented in Figure 16. This should be combined with displays which are located lower than the height of the face, and no screens should be used to block the view. One possible low height monitor setting is presented in Figure 17.
5.3.3 Communication protocol

Possibly the most important questions related to radical collocation concern communication. How much of it should exist, what communication should exist, when should it happen, who should communicate and why? This chapter gives clarifies those issues by trying to create a balance between noise and communication, and by taking into account concentration critical work.

The balance between noise and communication

As stated before under the topics 2.2.5 radical collocation and 2.4.1 the office, the noise generated in the common physical workspace can ruin concentration and cause stress. On the other hand, communication must exist, in order for the team to be able to collaborate. This creates a tradeoff situation – how much communication is an optimal amount?

The results from the case study show that overhearing discussions that concern your own work was considered to disturb less than non-work related discussions, and actually the work related discussions were not even considered disturbing, but beneficial. Overhearing work related discussions that were not related to your own work were considered disturbing. This means that you should not be able to hear people which are not related to your work, which is achieved by separating the radically collocated workspace from other workspaces with soundproof walls. The effect of noise is presented in figure 18.
Even though non-work related noise is harmful for work performance, non-work related discussions may be necessary for social identification. Because of this, it might be good to allow some level of non-work related communication in the radically collocated workspace, but still try to keep most of the non-work related discussion outside the workspace e.g. in the cafeteria or informal meeting spaces.

In addition to forbidding non-work related communication in the radically collocated workspace, communication protocols can be used to further increase the awareness of the team members. Simple communication rules can be created to inform others what you are doing, such as talking aloud when opening a file, so that people know what file you are opening and why. Furthermore, rules like "If you cannot solve the problem related to your work in 10 minutes, consult a colleague." can be useful in promoting collaboration between team members.
Concentration critical work

As stated in the chapter 2.2.1 **the characteristics of software development**, software development is a thinking-intensive and communication-intensive activity. This creates yet another tradeoff situation, sometimes the team members must concentrate fully in their own work, and sometimes they must be available for communication. The case study results indicated that people usually worked home when they require concentration, but occasionally did concentration critical work at the dedicated workspace too. Multiple methods can be suggested for facilitation concentration in the radically collocated workspace.

First, the team members that require concentration can leave the common physical workspace and work elsewhere in quiet. This enables maximum concentration, but can be interpreted as "escaping" from the group, and makes all information that is available in the common physical workspace (such as whiteboards) unavailable. In addition to this, the need for extra workspace can be excessive. This method was used in the study by Teasley et al. (2002).

Second, dedicated quiet moments can be used, meaning that no communication is allowed in the common physical workspace for a certain timeframe. This would mean that everyone in the team would have e.g. 2 hours a day, when they could work in the radically collocated workspace without disruptions. These hours can take place outside the regular work hours as well, if the working hours are flexible. This method is also used in the study by Teasley (2002).

Third, headphones can be used. The pilot study team members used headphones to block out the noise and indicate the need for personal concentration. This method was used only by one person in the case study, as people did not perceive headphones as necessary, because they considered that they are able to just ignore the noise. This practice could be encouraged more, but the downside is that headphones do not provide maximum concentration.

5.4 **Individual learning versus team learning**

The case study results indicated that it takes at least one month for a team to start to function efficiently. To maximize the performance of the software development team as a whole, it would feel natural to keep the team intact for as long as possible. This conflicts with the fact that an individual person can learn more when they are presented with new challenges (e.g. new projects),
meaning that the performance of a single individual will more likely increase more if teams are not kept intact for very long durations.

Because of this tradeoff, the optimum duration for keeping a team intact cannot be universally defined, as it depends on the organizational goals. In any case, team learning vs. individual learning is an aspect which should be kept in mind when resourcing projects. This means that a single individual might not always be the best resource unit to be used, but it should be possible to shift entire software development teams from one project to another.

5.5 Forcing awareness

The case study project manager had experimented with an electronic bulletin board, and she had came to the conclusion that a physical one would be better, because it would be impossible for the team members to not look at it. A radically collocated workspace is better than distributed settings in terms of awareness, because team members cannot avoid the awareness information while they are in the radically collocated workspace. This is beneficial when team members feel like they are too busy to follow up on the work done by others or if they are just not motivated enough to do it.

The overhearing effect (see 2.4.2 the effect of close proximity on communication) cannot be easily avoided; the only efficient way is by actively refusing not to listen, which most likely means that something is severely wrong in the project. The information that is transferred through the 2.4.3 project artifacts can be received even without intentional attention (e.g. by glancing at the whiteboard).

Forcing awareness is an aspect that should be taken in to account when planning on how to transfer information. In practice, this means leaving information in sight, instead of placing information to places which require a person to be active in order to access it. This is relatively easy in collocated setting, as physical environment can be used in displaying the information, but more difficult in distributed settings, where information is practically always behind at least one mouse click.
6 Summary and conclusions

This chapter summarizes this thesis by discussing how well were the goals of this thesis met and its limitations. Finally, implications for further research and conclusions are presented.

6.1 How well were the goals of this thesis met

The first goal was find out how radical collocation increase awareness in software projects. The research provided two answers to the question, the most important one being the causality chain of radical collocation, social identification, abundance of communication and high level of awareness, and the second was that radical collocation forces awareness. These two answers fulfill the goal well.

The second goal was to understand how radical collocation enhances coordination and collaboration in software projects. The 5.2 communication cycle presents the information transfer taking place in the radically collocated workspace, and information transfer can be seen as a prerequisite for coordination and collaboration. Understanding that radical collocation enhances coordination and collaboration through information transfer, and how that information transfer takes place, meets the goal adequately.

The third goal was to find out how the benefits of radical collocation could be maximized. This goal was met through giving improvement suggestions, such as re-arranging the seating and the use of communication protocols. The improvement suggestions meet the third goal well.

6.2 Limitations

This sub-chapter assesses the limitations of the literature study, research methods used and the case study, which are discussed below.

The literature study was carried out with the topics awareness, software development and workspace in mind. These topics were derived from the research questions, but it is possible that other topics could have been derived from the research questions as well. To ensure that relevant articles were not left out due to searching with keywords related to the selected topics, the
references in the articles that were found were given extra attention. The literature study was the basis for planning the pilot study and the case study. For these purposes, the literature study is considered adequate by the author.

Due to the qualitative nature of the research questions, unstructured observations and semi-structured interviews were selected as the research methods in the case study, as these methods were considered best by the author to provide data for answering the research questions. In addition to this, they were very applicable to the amount of interviewees and observable workspaces used in this study.

Nevertheless, the amount of research data in this study is somewhat limited, as the pilot study only consisted of one interview and six hours of observation, and the actual case study of 8 interviews and a total of 3.5 hours of observation of the dedicated workspace. This amount of data is enough for a qualitative study, but it would not have been enough for a quantitative study, which could have been used for measuring the amount of awareness through e.g. a survey.

It is also possible that the some or all of the case study interviewees are biased in their opinion about radical collocation, because they wish to move to a better workspace. This might cause understating of problems related to radical collocation and overstating the expected benefits of radical collocation.

Overall, it is uncertain how well the results of this study can be generalized, as the results may differ greatly depending on the interviewees and the workspaces in use.

6.3 Implications for further research

This thesis has created many implications for further research, first of which concerns software processes and their relation to the used workspaces. What kind of workspaces suit different software processes best is a subject worth researching further, since this thesis did not take in to account the effect of different software processes, and therefore was only a narrow view to the abundance of different workspaces and processes in use.

Awareness is something that could be measured by quantitative methods as well, for example by asking team members about how much they know about the work of the others in the project, and
see if they were right or not. The implication for further research here is that one could measure the awareness of people working in different workspace arrangements, and find out which arrangements are best for collaborative work.

One other issue regarding awareness might be worth looking into, and this is the question if the difficulties in distributed software development could be tackled by increasing the awareness of the team members, or are current day tools too inefficient in transferring adequate amount of awareness information. If so, more efficient tools could be developed.

Social identification in software development is strongly dependant on a collocated setting, but outside software development it may be possible to achieve social identification in distributed settings as well. Virtual realms like World of Warcraft could be an interesting research topic on how social identification works on distributed settings, and how communication or software development tools could be designed so that they would improve social identification.

More research on how much the workspace layout affects the overall success (or other factors, such as employee satisfaction) of software projects would be beneficial as well, since the subject has not been researched widely.

6.4 Conclusions

The purpose of this work was to examine the method of radical collocation and its relation to awareness. It seems clear that radical collocation is very effective in increasing the awareness of the team members, because it facilitates social identification, which in turn increases the amount of informal communication. This informal communication increases the information transfer between team members, and radical collocation forces other people to overhear it, which is not considered distracting, but rather beneficial, as long as the discussion is work related. This suggests that the proximity and social identification of the software development team members are both factors, which should be taken into account when managing software projects, and workplace layout can be used to influence them both.
References


Olson, G.M. & Olson, J.S. 2000, "Distance Matters", *Human-Computer Interaction*, vol. 15, no. 2/3, pp. 139-178.


Appendixes

Appendix A: The communication log sheet

Date: __________ Case project: ______________ Workspace: __________

<table>
<thead>
<tr>
<th>Start time</th>
<th>Initiated by</th>
<th>Participants</th>
<th>Content</th>
<th>Duration</th>
<th>Other notes (possible effect etc.)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

69
Appendix B: The interview guide

Personal:
Who are you and what do you do at company X?
What's your background at IBM?
Can you briefly describe your day at work?

Background:
Can you briefly describe the project at hand? (technologies etc)
How many people are working in the project?
What is the expected duration of the project?

Workspaces:
What kind of workspaces do you use? Who are the users of those workspaces?
- Closed offices, non-territorial offices, open-plan offices, bullpens, project rooms etc. Which ones are shared? Simultaneously or not?
- Do you have enough meeting rooms? Who do you meet there? Why not meet in the radically collocated room?
- Are there any informal places besides e.g. the cafeteria (meeting places outside elevators etc.)?
- Can you think of a way how your workplace / workspace affect your work productivity?
- Does the workplace affect your use of time in the workplace (problems in reserving meeting rooms or finding people, easy to find people for knowledge sharing or horizontal coordination)?
- How would you describe the work atmosphere / stress level during different phases of project / iteration?
- Are there lots of interruptions or other problems in your work resulting from noise?
- How much time do you spend daily in different workspaces (including cafeteria)? (käyttöaste)

Processes:
What kind of software process do you use?
- Plan-driven (meetings, solitary work?)
- Agile (Scrum, what, iteration lengths?)
- What communication tools and technologies do you use?
- How would you describe the software process used (negative /positive)?
- Do you see the workspaces in use prohibit or enable the processes in use?
• Rate on 1-5, how well does your workspace meet your software process needs?

Description of different communication modes (impersonal, formal f-to-f, informal f-to-f) here:

How do you distribute work tasks? Does re-distribution occur among team members? (coordination related)

Do you show (teach) other people how things are done, or do they do the same to you? In what situations and how?

Do you discuss work related issues that do not concern your current project with somebody? Who, when, where and why?

Can you think of where you last had a conversation regarding some innovation and in what situation / where?

How would you describe the level of uncertainty (changes, lack of information, confusion) during different phases of project / iteration?

How would you describe the level of work interdependence (inside team / project) during different phases of project / iteration?

Is there any personal (non-work related) communication at work (also in email etc)? Where, when, with whom? Do you think it affects e.g. the work atmosphere? Do you feel it is a source of interruptions?

Description of radical collocation here:

**Radical collocation general + Awareness:**
For what kind of work is radical collocation suitable? Why?

For what kind of work is radical collocation unsuitable? Why?

Do you perceive software development as collaborative work?

In what detail do you see that team members should know about the project? Why?

In what detail do you see that team members should know what everyone else is doing in the project/team at the moment? Why? How much do you know about what everyone else is doing? Do you see that radical collocation assist in knowing what everyone else is doing? How much?

Do team members learn more in radical collocation?

**Radical collocation:**
What is the cause the teams are radically collocated?
How do you see that radical collocation helps in reaching that cause?
What are the expected benefits of radical collocation?

Do you see that the layout is designed to reach the cause of radical collocation?
What is very good in the existing workspace?
What is not good in the existing workspace?
What are the most important benefits in the workspace?
What are the "rules" (communication etc) in the radically collocated workspace?
How would you improve the existing workspace, if the sky was the limit?
What would you try to achieve by that?
How do you see that the size of the project is related to radical collocation?
What your opinion on radical collocation?
Rate radical collocation on 1-5?

Can you think of a way how your workplace / workspace affects your work productivity?