



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Seminar Ringvorlesung Home Office 2.0 - Virtual Project Office Karsten Panier

Inhaltsverzeichnis

1	Einleitung	3
2	Gegenstandsbereich	4
2.1	Verteilte Teams	4
2.2	Probleme	5
3	Masterarbeit	7
3.1	Ansatz	7
3.2	Inspiration	8
3.3	Vorarbeiten	8
3.4	Architektur	9
3.5	Vorgehensweise	11
3.6	Risiken	11
3.7	Methodik	12
4	Fazit	13
	Literaturverzeichnis	14

1 Einleitung

Softwareprojekte werden zunehmend in verteilten Teams entwickelt [26]. Dadurch wird die Kommunikation und Zusammenarbeit im Projekt beeinträchtigt. Grund hierfür ist die Distanz zwischen den Teammitgliedern. Dabei kann zwischen folgenden Distanzen unterschieden werden:

- **Räumliche Distanz**
Die Mitarbeiter befinden sich an unterschiedlichen Standorten und können nicht von Angesicht zu Angesicht miteinander kommunizieren.
- **Zeitliche Distanz**
Die Mitarbeiter haben unterschiedliche Arbeitszeiten, die sich wenig oder gar nicht überschneiden. Dadurch muss ein Teil der Kommunikation asynchron erfolgen.
- **Kulturelle Distanz**
Kulturen haben unterschiedliche Regeln und Verhaltenskodexe. Diese sind den Mitgliedern einer Kultur unterbewusst bekannt, aber bei Mitgliedern anderer Kulturen können sie zu Missverständnissen führen. Kulturunterschiede treten auch im gleichen Land auf oder zwischen Unternehmen. [14]
- **Sprachliche Distanz**
In verteilten Teams, in denen verschiedene Muttersprachen gesprochen werden, wird eine Sprache als vorherrschende Projektsprache definiert. Da eine solche Sprache nicht von allen Mitgliedern auf gleichem Niveau beherrscht wird, kann es zu Verständnisproblemen kommen.

Diese Distanzen können in unterschiedlichen Kombinationen auftreten. [5]

Für Agile Projekte stellen diese Distanzen ein besonderes Problem dar. In den zwölf Prinzipien des agilen Manifests wird Wert auf eine direkte Kommunikation von Angesicht zu Angesicht gelegt. [1] Kent Beck empfiehlt ausdrücklich, dass das Team in einem Raum zusammenarbeitet, um ein gemeinsames Bewusstsein für die Arbeit am Projekt zu gewährleisten. [3]

Die Zielsetzung der bevorstehenden Masterarbeit ist die Entwicklung eines virtuellen Projektraums, der den Entwicklern auch über räumliche Distanzen hinweg ein Bewusstsein für die Arbeit der Kollegen am Projekt ermöglicht. Dadurch liegt der Schwerpunkt der Arbeit auf der Überbrückung von räumlichen Distanzen und berücksichtigt weniger die anderen Distanzen.

2 Gegenstandsbereich

Der virtuelle Projektraum ist zur Unterstützung von agilen Softwareentwicklungsteams gedacht, die verteilt arbeiten. Dabei liegt der Schwerpunkt auf der räumlichen Distanz und berücksichtigt im geringen Maße die zeitliche.

2.1 Verteilte Teams

Im Zusammenhang mit der räumlichen Distanz sollte die Organisation des Teams berücksichtigt werden. Diese kann auf zwei Arten erfolgen:

- **Verteilte Teams**
Bei einem verteilten Team entspricht die Organisation des Teams den Standorten. Dies bedeutet an jedem Standort wird ein Subsystem des Gesamtsystems entworfen.
- **Verstreute Teams**
Bei einem verstreuten Team sind die Mitarbeiter eines Teams oder Teilteams über mehrere Standorte verteilt. Teamgrenzen werden dabei nicht nach Standort- oder Teilsystemen entwickelt.

Beide Formen haben Vor- und Nachteile. Ein verteiltes Team kann lokal besser an seinem Teilsystem arbeiten und bei definierten Schnittstellen wird die Integration des Gesamtsystems ermöglicht. Es kann aber zu Spannungen zwischen den Teilteams kommen, wenn auf einer Seite der Eindruck entsteht es werden von der anderen Vorschriften gemacht oder Schuldzuweisungen in den Raum gestellt. Die Verteilung der Subsysteme über die Standorte, befreit nicht von der Aufgabe eine gemeinsame Vision zu erzeugen.

Dies wird eher in einem verstreuten Team erreicht, da von vornherein eine begleitende und intensive Kommunikation zwischen den Standorten notwendig ist. Dies bedingt, dass sich das Team regelmäßig zu persönlichen Treffen an einem Standort zusammenfindet. Dabei bietet es sich an den Standort regelmäßig zu wechseln, so dass alle Teammitglieder den Reiseaufwand auf sich nehmen müssen. Zudem lernen die Mitarbeiter so die lokalen Gegebenheiten der Kollegen kennen und verstehen. [5]

Die Entfernung zwischen den Standorten ist entscheidend für die Häufigkeit der möglichen Treffen, da ab einer gewissen Entfernung der zeitliche und finanzielle Aufwand für häufiges Reisen zu groß wird. [5]

Neben den verteilten Standorten gibt es auch an einem Standort folgende räumliche Distanzen:

- **Mehrere Büros**
Bereits die Verteilung auf mehrere Büros erzeugt eine Distanz. Die Situation des Kollegen im anderen Büro ist nicht wahrnehmbar. Dies führt dazu, dass ein Problem im Projekt nicht jedem im Team klar wird. Die Kommunikation wird ebenfalls erschwert, da es mehr Aufwand bedeutet mit dem Kollegen in Kontakt zu treten. Es ist auch nicht einfach erkennbar, wann ein guter Zeitpunkt ist einen Kollegen anzusprechen.
- **Heimarbeit**
Bei der Heimarbeit entfällt die Möglichkeit der direkten Kommunikation und auch die Flurgespräche sind nicht möglich. In der Regel ist die Heimarbeit zeitlich begrenzt. Eine regelmäßige direkte Kommunikation vor Ort ist somit teilweise gegeben.

Generell lassen sich für die verteilte Zusammenarbeit drei evolutionäre Stufen identifizieren. In der ersten Stufe wird lokal an einem Standort gearbeitet. Es findet keine globale Entwicklung statt. Dies wird in der zweiten Stufe erreicht. Dort wird an mehreren Standorten zusammen an einem System gearbeitet. Das Team wird von einem zentralen Standort aus koordiniert. In der dritten Stufe entfällt diese zentrale Kontrollinstanz und alle Standorte arbeiten gleichberechtigt in Teams. Die Koordination funktioniert im Netzwerk der Beteiligten. Für ein agiles Team, das sich durch Selbststeuerung auszeichnet ist die dritte Stufe notwendig. [4]

2.2 Probleme

Durch diese Verteilung in agile Teams entstehen neue Herausforderungen und bestehende werden verschärft. An dieser Stelle wird eine Auswahl näher beschrieben.

- **Redundante Arbeit**
Bei der Softwareentwicklung werden viele Schwierigkeiten erst während der Entwicklung sichtbar. Oftmals betreffen diese nicht nur einen Teil der Entwicklung oder des zu entwickelnden Systems. Daher ist die Gefahr groß, dass mehrere Entwickler unabhängig von einander an derselben Lösung arbeiten. Aus diesem Grund definieren agile Methoden tägliche Absprachen, wie *Daily Scrum* um auf diese aufmerksam zu machen.

Damit dieser Austausch funktionieren kann, müssen die Teammitglieder lernen welche Informationen relevant sind. Dadurch wird ein aktives Zuhören im Team überhaupt möglich. Ebenso muss eine Kultur des Vertrauens vorhanden sein, damit die einzelnen Mitglieder über ihre Probleme und Schwierigkeiten sprechen. Bei einem lokalen Team ist dies leichter zu lernen, da auch außerhalb der täglichen Absprachen Informationen transportiert werden.

- **Vertrauen**
Die Grundlage für erfolgreiche Teamarbeit ist Vertrauen. Dies zeigt sich nicht nur

darin Probleme anzusprechen, sondern auch in der Akzeptanz der Ergebnisse eines Kollegen. Hat ein Entwickler kein Vertrauen zu seinem Kollegen, so wird er dessen Ergebnisse in Frage stellen und gegebenenfalls daran vorbei arbeiten.

Damit Vertrauen entstehen kann, bedarf es einer offenen, ehrlichen und zuverlässigen Kommunikation. Dabei entsteht es langsam und in kleinen Schritten. Demgegenüber reicht eine Kleinigkeit um das Vertrauen sofort zu brechen. [28]

Vertrauen wird durch persönlichen Kontakt hergestellt und muss gerade bei räumlicher Distanz besonders gepflegt werden. [12]

- Teambildung

Die beste Voraussetzung für die Teambildung ist der persönliche Kontakt. Für verteilte Teams schlägt Scott Ambler daher vor, zu Beginn des Projektes eine Zeit lang gemeinsam an einem Standort zu arbeiten. [2] Diese Maßnahme erlaubt den Mitarbeitern zu einem Team zu werden und später als verteiltes Team zu agieren. Damit es als verteiltes Team funktioniert ist Aufwand nötig, der durch elektronische Kommunikation erreicht werden kann. [20]

- Kommunikation

Die Kommunikation stellt die Basis des Vertrauens dar. Zudem ist sie notwendig für die Erzeugung und Aufrechterhaltung des Teams. Über mehrere Standorte wird die Kommunikation erschwert vgl. 2.1. Diese kann lediglich über Telefon oder Nachrichten erfolgen. Auf diesem Wege werden nonverbale Ausdrücke nicht übertragen. Dies führt leicht zu Missverständnissen. [27]

Eine weitere Schwierigkeit ergibt sich aus der Kommunikationsanbahnung. Da man den potentiellen Gesprächspartner nicht vor Augen hat, weiß man nicht ob man ihn gerade jetzt mit einer vermeintlich weniger wichtigen Sache stören kann. Dabei muss zwischen synchroner und asynchroner Kommunikation gewählt werden. Die synchrone Variante unterbricht zwar den Kollegen bei seiner Arbeit, jedoch bekommt man eine direkte Reaktion auf die Anfrage. Kommuniziert man asynchron vermeidet man die Störung, riskiert aber Verzögerungen bei der Beantwortung einer eventuell wichtigen Frage.

Die vorgestellten Herausforderungen zeigen die Bedeutung einer effizienten Kommunikation. Neben der Qualität der Kommunikation ist es wichtig ihren Bedarf überhaupt zu erkennen. Dieser ist in lokalen Teams deutlicher sichtbar als in verteilten.

3 Masterarbeit

Im Kapitel 2.2 wurden einige Herausforderungen identifiziert, die aufgrund der räumlichen Distanz entstehen. Im Rahmen der Masterarbeit soll ein System zur Unterstützung von agilen Teams entwickelt werden. Ein Schwerpunkt bildet die Verbesserung der Kommunikation und der Analyse ihres Bedarfs. Erreicht werden soll dies durch Sensibilisierung der Teammitglieder für die Tätigkeiten der räumlich entfernten Kollegen.

3.1 Ansatz

Der Ansatz der Masterarbeit ist es einen virtuellen Raum, das *Virtual Project Office*, für das Team zu schaffen.

Das Leitbild für die Arbeit stellt ein Großraumbüro dar, in dem das Team gemeinsam arbeitet. Charakteristisch dafür ist eine eigene Atmosphäre, die durch die Geschäftigkeit der Kollegen entsteht. Dabei registrieren die Teammitglieder unterschwellig die Aktivitäten ihrer Kollegen. Gerade bei agilen Teams herrscht in diesen Räumen ein lautes Hintergrundrauschen, da viel kommuniziert wird. Es zeigt sich, dass diese Gespräche unterstützend wahrgenommen werden, solange sie das Projekt betreffen. [5] Es entsteht dadurch eine Atmosphäre, die das Gemeinschaftsgefühl fördert. [3]

Dies bedeutet, dass kein virtueller Raum wie SecondLife [17] gebaut werden soll, sondern ein System, dass den Informationsaustausch für verteilte Teams verbessert. Der Schwerpunkt liegt dabei auf Informationen, die bei einem lokalen Team unbewusst durch Haltung und Gestik ausgetauscht werden. [25] Da sie wie ein Hintergrundrauschen in einem Raum mitgeteilt werden, ohne dass explizit jemand angesprochen wird. Durch diesen impliziten Charakter handelt es sich bei dem *Virtual Project Office* um ein Peripheral Awareness System. [23]

Ziel des Systems ist es die Teammitglieder näher miteinander zu verbinden und Möglichkeiten zur Kontaktpflege zu schaffen. Dabei wird der Kontext des Mitarbeiters berücksichtigt. Um ihm Hilfestellung bei seiner aktuellen Aufgabe zu geben, werden ihm mehr Informationen zu den verwendeten Ressourcen angeboten. Darin enthalten ist auch die Angabe der für seine Aufgabe relevanten Experten [13].

Neben dem zu entwickelnden System besteht eine weitere Aufgabe darin, die Methodik des Teams zu erweitern. Die Teammitglieder sollen so lernen, die für das System benötigten Informationen bereitzustellen. Damit ist zum einen die explizite Auswahl der zu bearbeitenden Aufgabe gemeint, aber auch das Signalisieren, dass man sich in einer konzentrierten Phase befindet und nicht unterbrochen werden möchte.

3.2 Inspiration

Verteilte Arbeit und Kommunikation funktioniert in verschiedenen Bereichen. Von diesen kann man lernen und funktionierende Prinzipien in dem *Virtual Project Office* abbilden.

Ein Beispiel für eine verteilte Form der Softwareentwicklung ist Opensource. [15] In vielen Projekten arbeiten dabei Menschen zusammen, die sich nicht persönlich kennengelernt haben, aber dennoch vertrauensvoll als Team arbeiten. Die Community-Prozesse in solchen Projekten basieren auf persönlicher Anerkennung, die man sich erarbeiten kann. Von Sun wurde mit dem Projekt Sun Spaces [24] eine Inhouse Variante aufgebaut, um diese Strukturen auf ein geschlossenes Team zu übertragen.

Damit nutzt Sun Spaces Web 2.0-Technologien [21] im Unternehmen und stellt somit eine Enterprise 2.0 Lösung dar. Kennzeichnend für Enterprise 2.0 ist die Übertragung von Kommunikationsstrukturen und Formen des Wissensaustausches, wie sie in Sozialen Netzwerken des Internets praktiziert werden, auf den Kontext eines Unternehmens. [19] In diesem Ansatz verschiebt sich der Fokus vom Dokument hin zum Autor, wodurch das Bewusstsein für die Kollegen geschärft wird. Eine Möglichkeit diesen Ansatz in das *Virtual Project Office* zu übertragen, wäre ein Micro-Blogging System wie Twitter [29] einzurichten. Mitarbeiter, die zentrale Rollen wie den Buildmanager innehaben, könnten den Status des Builds und der Testsysteme mitteilen oder woran sie gerade arbeiten. Die Kollegen könnten dadurch den aktuellen Status erfahren und wüßten bei einem Problem, ob bereits daran gearbeitet wird. Diese Transparenz des Buildmanagers kann das Vertrauen seiner Kollegen in ihn stärken.

In Kapitel 2.2 wurden die Vor- und Nachteile von synchroner und asynchroner Kommunikation diskutiert. Typische Medien dafür sind bei verteilten Teams das Telefon für synchrone und E-Mail für asynchrone Kommunikation.

Eine Mischung aus beiden bietet das Instant Messaging [31]. Dabei werden kürzere Textnachrichten übertragen. Der Empfänger kann entscheiden wann er darauf antworten möchte. Entscheiden sich beide zu einem Dialog, entsteht durch die kurze Form der Nachrichten ein Gespräch. Ist dann bereits eine Kommunikation entstanden, kann jederzeit auf das Telefon gewechselt werden, um einem direkten Gespräch näher zu kommen.

Eine wichtige Funktion von Chat-Programmen in diesem Zusammenhang ist, dass sie den Status der Teilnehmer anzeigen. Dadurch wird signalisiert ob er verfügbar oder abwesend ist. Dies zeigt, ob man einen Kollegen stören darf oder er nur im Notfall erreichbar sein will. Individuelle Statusnachrichten erlauben es dem Nutzer seinen Kontakten eine generelle Kurzmitteilung zu übermitteln. Außerdem wird bei einem Chat signalisiert, dass der Gesprächspartner an einer Antwort tippt. Dadurch weiß der Empfänger, dass in kurzer Zeit eine Reaktion erfolgen wird.

3.3 Vorarbeiten

Für die Realisierung des *Virtual Project Office* sind bereits Vorarbeiten gemacht worden.

Zunächst wurden verschiedene Arbeiten untersucht und geprüft welche Infrastrukturen und Erfahrungen sich in die Realisierung einbeziehen lassen [22]. Bei der Arbeit in einem Großraumbüro können sich die Teammitglieder der aktuellen Aufgabe ihrer Kollegen bewusst werden. An welcher Aufgabe arbeitet der Kollege und mit welchen Strategien löst er diese. Diesen Kontext zu einer Aufgabe zu erfassen und detailliert zu modellieren, wurde von Mik Kersten durchgeführt [16]. Das daraus entstandene Werkzeug Mylyn [7] lässt sich zur Extraktion der Tätigkeiten der einzelnen Teammitarbeiter nutzen.

Diese Aufgabenkontexte können als Basis genutzt werden, um dem Team zu zeigen wer an welchem Thema arbeitet. Durch einen zentralen Abgleich der Kontexte kann ein System den an ähnlichen Themen arbeitenden Entwicklern den Hinweis geben, dass es sinnvoll wäre sich untereinander auszutauschen. Solche Verfahren zum Abgleich von Kontexten unter Berücksichtigung von Benutzer- und Teamumgebungen werden im Active Project betrachtet [11].

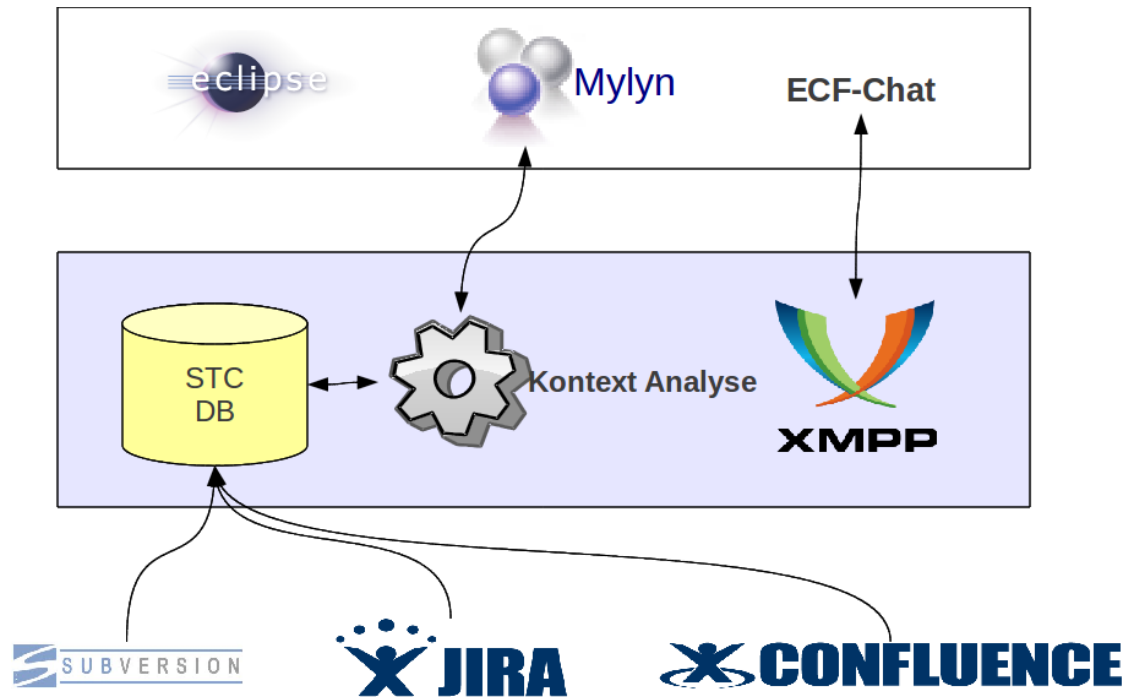
Neben den aktuellen Tätigkeitsinformationen des Teams, können auch zurückliegende Informationen von Nutzen sein. So könnte ein gerade bestehendes Problem durch bereits früher erworbenes Wissen eines anderen Entwicklers gelöst werden. So werden auch die Gründe für die Designentscheidungen nachvollziehbar. Diese Informationen lassen sich aus den Entwicklungswerkzeugen unter Berücksichtigung der sozialen Beziehungen gewinnen [30].

Im Rahmen der Veranstaltungen Projekt 1 und Projekt 2 wurde hierfür die Infrastruktur *LuPanKu* entwickelt. Diese extrahiert Informationen aus Werkzeugen der Softwareentwicklung und setzt sie in Beziehung zueinander [18].

3.4 Architektur

Aus diesen Vorarbeiten wurde für das *Virtual Project Office* die Architektur in Abb. 3.1 entwickelt. In der Architektur sind drei Hauptkomponenten definiert:

- Client
 - Da es sich bei der Zielgruppe des *Virtual Project Office* um Entwickler handelt, wird der Client durch Plug-Ins für Eclipse basierte Entwicklungsumgebungen realisiert [10]. Es sind zwei Plug-Ins vorgesehen:
 - Chat Client
 - Für die Darstellung des *Virtual Project Office* wird ein Chat-Client auf Basis des Eclipse Communication Frameworks [6] verwendet. Dabei wird das Team durch eine Kontaktliste dargestellt. Neben einer Chat Funktionalität liefert die Kontaktliste Informationen über die aktuellen Tätigkeiten der Teamkollegen. Weiterhin ist eine Visualisierung von Kollegen geplant, die einen Bezug zu der aktuellen Aufgabe haben.
 - Mylyn Context Share
 - Wie in 3.3 beschrieben, soll der Aufgabenkontext der Mitarbeiter zentral ana-


Abbildung 3.1: *Virtual Project Office* Architektur

lysiert werden. Dafür ist ein Plug-In notwendig, das den Mylyn-Kontext extrahiert und an einen zentralen Server übermittelt.

- *LuPanKu*-Server

Die zentrale Infrastruktur des *Virtual Project Office* wird durch den *LuPanKu*-Server bereitgestellt. Dieser enthält für die Clients einen Chat-Server. Die Kontexte der Teammitglieder werden zum Kontext-Analyzer übermittelt. Dieser kann dem Chat-Server weitere Statusinformationen über die Teammitglieder zur Verfügung stellen. Des weiteren führt er den in Kapitel 3.3 geforderten Kontextvergleich durch.

Die Kontexte werden zusätzlich mit den Metainformationen aus den Entwicklungswerkzeugen abgeglichen. Diese werden in Form eines Social-Technical-Congruence Networks (STC) in einer Datenbank gespeichert. [11]

- Entwickler Werkzeuge

Der Server erzeugt die STC-Datenbasis aus verschiedenen Werkzeugen wie Versionsverwaltungssysteme, Bug Tracking und Wikis. Diese Informationen werden dann in Beziehung zu den Benutzern und ihrem sozialen Netzwerk gespeichert.

3.5 Vorgehensweise

Das *Virtual Project Office* soll in Zusammenarbeit mit einem realen Projekt entwickelt werden. Dadurch soll zum einen der Nutzen des Systems verifiziert werden und eine bedarfsorientierte Planung erfolgen. Im ersten Schritt wird der *LuPanKu*-Server installiert und mit den vom Team verwendeten Werkzeugen konfiguriert. Danach erfolgen Basisimplementationen der geplanten Komponenten Chat-Client und Mylyn Context Share. Diese werden im weiteren Verlauf entsprechend den Bedürfnissen des Teams weiterentwickelt.

Die zentrale Komponente, um peripheral awareness zu ermöglichen, ist der Kontext Analyzer. Dieser muss dafür Kontexte miteinander vergleichen, um den Entwicklern wertvolle Information zu liefern. Der Abgleich von Kontexten wird in mehreren Schritten implementiert. Die erste Stufe berücksichtigt lediglich die verwendeten Ressourcen im Kontext. In einer zweiten Stufe fließen dann die Degree of Interest Informationen ein [16]. Ob diese Abgleichverfahren den Anforderungen genügen soll geprüft werden oder ob weitere evaluiert werden müssen. Eine mögliche Ausbaustufe wäre es zu berücksichtigen, durch welche Tätigkeit eine Ressource in den Kontext aufgenommen wurde.

Der Kontextabgleich liefert wertvolle Informationen zu den aktuellen Tätigkeiten. Daneben kann es wie in Kapitel 3.3 beschrieben von Nutzen sein die vergangene Entwicklung des Projektes zu berücksichtigen. Dafür muss der aktuelle Kontext mit den Informationen aus der STC-Datenbank abgeglichen werden.

3.6 Risiken

Bei der Realisierung des *Virtual Project Office* sind einige Risiken zu berücksichtigen.

Für die Erzeugung der benötigten Kontexte sind zwei Quellen vorgesehen. Dabei handelt es sich zum einen um die Komponente Mylyn zur Erzeugung des aktuellen Aufgabenkontextes. Dies setzt voraus, dass die Entwickler Mylyn verwenden. Die andere Quelle besteht in den Informationen, die sich aus den Werkzeugen extrahieren lassen. Die Qualität dieser Informationen hängt davon ab, ob die Entwickler die benötigten Metainformationen, wie Tags im Wiki oder Referenzen auf Fehler beim Check-In pflegen.

Die Lösung wird in Zusammenarbeit mit einem Team entwickelt. Dadurch wird der Nutzen für das Team überprüft. Dies erlaubt jedoch keine generelle Aussage, ob die Lösung insgesamt oder in Teilen auch auf andere Teams übertragbar ist. Jedes Projekt ist individuell. Daher müssen Methoden und Praktiken in jedem Projekt neu angepasst oder erarbeitet werden.[9]

Ein Mehrwert entsteht nur, wenn das gesamte Team das System nutzt. Dafür muss es von den Entwicklern akzeptiert werden. Daraus ergeben sich Anforderungen an die Usability des Werkzeugs. Es soll dezent Informationen liefern und darf den Entwickler nicht bei seiner Arbeit behindern. Das System macht die Aufgaben und deren Umsetzung jedes

Entwicklers im Team sichtbar. Dies könnte zu einem Vertrauensproblem führen, wenn die Informationen zu seinem Nachteil verwendet werden. Daher muss der Entwickler die Möglichkeit haben seine Privatsphäre zu schützen.

Die Qualität der bereitgestellten Informationen muß kritisch betrachtet werden. Es können unnütze oder falsche Informationen durch den Kontext-Analyzer ermittelt werden, die dem Entwickler nicht weiterhelfen. Dies kann zu einem Vertrauensverlust in das System führen. Werden auf der anderen Seite die Regeln für die Informationsbereitstellung zu restriktiv gefasst, können wertvolle Informationen verloren gehen.

3.7 Methodik

Zur Minimierung der vorgestellten Risiken und der frühzeitigen Identifikation von Problemen ist ein methodisches Vorgehen angelehnt an agilen Verfahren geplant. Das System wird in kleinen Iterationen entwickelt. Bei jeder Iteration soll ein Inkrement an den Kunden, das agile Team, ausgeliefert werden. Dabei steht für jede Entwicklung der Mehrwert für den Kunden im Vordergrund. Jede Iteration wird mit den Erfahrungen und Anregungen des Kunden neu geplant. Die Leitlinie für die Realisierung ist es mit einfachen Maßnahmen einen hohen Mehrwert für das Team zu erlangen. Dies soll zu einem einfachen Design und einer intuitiven Handhabbarkeit des Systems führen.

Die Anwendung soll die sozialen Beziehungen im Team unterstützen. Dies führt zu einer Wechselwirkung der Technik und dem Team. Es muss daher als ein sozio-technisches System betrachtet werden [8]. Dies bedeutet das zu den Methoden der Ingenieurwissenschaften ebenfalls Methoden der empirischen Sozialforschung berücksichtigt werden müssen. So kann nicht allein durch Metriken, die den Teamerfolg messen, Rückschlüsse auf den Einsatz des *Virtual Project Office* gezogen werden, da in einem so komplexen System viele Einflüsse eine Rolle spielen. Es müssen empirische Fallstudien analysiert werden um die Theorie zu validieren. Dabei spielen die kurzen Feedbackzyklen der agilen Entwicklung eine wichtige Rolle, da sie eine Änderung in der sozialen Teamstruktur durch das *Virtual Project Office* leichter identifizierbar machen.

4 Fazit

In Kapitel 2 wurden verteilte agile Teams und deren Herausforderungen beschrieben, die durch verschiedene Distanzformen entstehen. Daraus wurde der Ansatz eines *Virtual Project Office* für räumliche Distanz abgeleitet. Zielsetzung dieses Ansatzes ist das Bewusstsein des Teams für entfernt sitzende Kollegen zu schärfen. Dabei sollen die Tätigkeiten und Situationen sichtbar gemacht werden und die Kommunikation gefördert werden.

Aus dieser Vision und den Vorarbeiten 3.3 wurde eine Architektur abgeleitet, um ein Peripheral Awareness System zu entwickeln. Dieses soll in enger Kooperation mit einem verteilten Team erstellt werden, um dessen Nutzen zu überprüfen. Dabei werden unterschiedliche Arten von räumlichen Distanzen, wie Home Office, mehrere Büros und Standorte berücksichtigt. Dafür wurde in Kapitel 3.5 ein erster Arbeitsplan entworfen. Dieser wird bedingt durch das iterative Vorgehen aus Kapitel 3.7 während der Arbeit angepaßt und verfeinert werden. Das Vorgehen dient auch zum frühzeitigen Entdecken von Risiken wie in Kapitel ?? beschriebenen. Dadurch soll eine Reaktion auf diese im Rahmen der Masterarbeit ermöglicht werden.

Literaturverzeichnis

- [1] AGILE: Manifesto for Agile Software Development. (2010), 02. – URL <http://agilemanifesto.org/>
- [2] AMBLER, Scott: Bridging the Distance. In: *Dr. Dobb's Portal* (2002). – URL <http://www.drdobbs.com/184414899>
- [3] BECK, Kent: *Extreme Programming Das Manifest*. Addison Wesley, 2000
- [4] CARMEL, Erran: *Global software teams: collaborating across borders and time zones*. Upper Saddle River, NJ, USA : Prentice Hall PTR, 1999. – ISBN 0-13-924218-X
- [5] ECKSTEIN, Jutta: *Agile Softwareentwicklung mit verteilten Teams*. dpunkt.verlag, 2009
- [6] ECLIPSE: *Eclipse Communication Framework*. – URL <http://www.eclipse.org/ecf/>. – Zugriffsdatum: 20. Dezember 2010
- [7] ECLIPSE: *Eclipse Mylyn*. – URL <http://www.eclipse.org/mylyn>. – Zugriffsdatum: 14. Juli 2010
- [8] FERSTL, Otto K. ; SINZ, Elmar J.: *Grundlagen der Wirtschaftsinformatik 1*. Oldenbourg, 2006. – URL <http://www.worldcat.org/isbn/9783486579420>. – ISBN 9783486579420
- [9] FLOYD, C.: Software-Engineering - und dann? In: *Informatik Spektrum* 17 (1994), Nr. 1, S. 29–37
- [10] FOUNDATION, Eclipse: *Eclipse*. – URL <http://www.eclipse.org/>
- [11] GOMEZ-PEREZ, Jose M. ; GROBELNIK, Marko ; RUIZ, Carlos ; TILLY, Marcel ; WARREN, Paul: Using task context to achieve effective information delivery. (2009), S. 1–6. ISBN 978-1-60558-528-4
- [12] HANDY, C.: Trust and the virtual organization. In: *Harvard Business Review* (1995)
- [13] HOLSTEN, Matthias: Expertensuche im Kontext von Enterprise 2.0. (2010). – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2010-aw2/holsten/bericht.pdf>
- [14] HVATUM, Lise B.: Agile practices and Distributed Teams. In: *Cutter IT Journal* 20 (2007)

- [15] INITIATIVE open source: *Mission Open Source Initiative*. – URL <http://www.opensource.org/>
- [16] KERSTEN, Mik: *Focusing knowledge work with task context*. Vancouver, BC, Canada, Canada, Dissertation, 2007
- [17] LINDEN RESEARCH, Inc.: *SecondLife*. – URL <http://secondlife.com/>
- [18] MATTHIAS HOLSTEN, Karsten Panier Daniel W.: *Projektbericht Home-Office 2.0*. 2010. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2010-proj1/panier.pdf>
- [19] MCAFEE, Andrew P.: Enterprise 2.0: The Dawn of Emergent Collaboration. In: *MIT Sloan* 47 (2006), Spring, Nr. 3, S. 21–28
- [20] MCKINNEY, Vicki R. ; WHITESIDE, Mary M.: Maintaining distributed relationships. In: *Commun. ACM* 49 (2006), Nr. 3, S. 82–86. – ISSN 0001-0782
- [21] O'REILLY, Tim: *Web 2.0 Compact Definition: Trying Again*. Oreilly. 02 2010. – URL <http://radar.oreilly.com/archives/2006/12/web-20-compact.html>
- [22] PANIER, Karsten: *Home Office 2.0 - Collaborative Working - Related Work*. 2010. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2010-aw2/Panier/bericht.pdf>. – Zugriffsdatum: 20. Dezember 2010
- [23] PEDERSEN, Elin R.: People presence or room activity supporting peripheral awareness over distance. In: *CHI 98 conference summary on Human factors in computing systems*. New York, NY, USA : ACM, 1998 (CHI '98), S. 283–284. – URL <http://doi.acm.org/10.1145/286498.286763>. – ISBN 1-58113-028-7
- [24] REISER, Peter: *Community Equity Overview*. – URL <http://www.slideshare.net/peterreiser/community-equity-overview-2419832>
- [25] RICKHEIT, Gert (Hrsg.) ; DEUTSCH, Werner (Hrsg.) ; HERMANN, Theo (Hrsg.): *Psycholinguistik / Psycholinguistics*. Berlin : De Gruyter Mouton, 2003 (Handbücher zur Sprach-und Kommunikationswissenschaft / Handbooks of Linguistics and Communication Science 24). – URL get-book.cfm?BookID=7654
- [26] SHAO, Benjamin ; DAVID, Julie S.: The impact of offshore outsourcing on IT workers in developed countries. In: *Commun. ACM* 50 (2007), February, S. 89–94. – URL <http://doi.acm.org/10.1145/1216016.1216026>. – ISSN 0001-0782
- [27] THUN, Friedemann S. von: *Miteinander reden 1. Störungen und Klärungen. Allgemeine Psychologie der Kommunikation*. Reinbek bei Hamburg : Rowohlt, 1981 (Rororo. 7489=Rororo-Sachbuch). – ISBN 3-499-17489-8
- [28] TUCKMAN, B.: Developmental sequence in small groups. In: *Psychological Bulletin* 63 (1965), S. S. 384–389

- [29] TWITTER: *Twitter*. – URL <http://www.twitter.com>. – Zugriffsdatum: 20. August 2010
- [30] VALETTO, Giuseppe ; HELANDER, Mary ; EHRLICH, Kate ; CHULANI, Sunita ; WEGMAN, Mark ; WILLIAMS, Clay: Using Software Repositories to Investigate Socio-technical Congruence in Development Projects. (2007), S. 25. ISBN 0-7695-2950-X
- [31] WIKIPEDIA: *Instant Messaging* — *Wikipedia, Die freie Enzyklopädie*. 2011. – URL http://de.wikipedia.org/w/index.php?title=Instant_Messaging&oldid=83733164. – [Online; Stand 6. Februar 2011]