

# Meeting Management mit Microsoft Surface 2

## Studienarbeit

Abteilung Informatik  
Hochschule für Technik Rapperswil

Frühjahrssemester 2012

Autor(en): Marco Daniele, Michael Meier, Silka Simmen  
Betreuer: Prof. Dr. Luc Bläser  
Projektpartner: Technology Innovation & Management AG, Zürich

<b>1</b>	<b>Abstract</b> .....	<b>5</b>
<b>2</b>	<b>Management Summary</b> .....	<b>6</b>
2.1	Ausgangslage .....	6
2.2	Vorgehen, Technologien .....	6
2.3	Ergebnisse .....	8
2.4	Ausblick .....	8
<b>3</b>	<b>Projektmanagement</b> .....	<b>9</b>
3.1	Projektorganisation .....	9
	Projektbetreuung .....	9
	Verantwortlichkeiten .....	9
	Das Entwicklungsteam .....	10
3.2	Planung .....	11
	Status Meetings und Code Reviews .....	11
	Meilensteine .....	11
	Versionen .....	11
	Zeitplan .....	11
3.3	Infrastruktur .....	13
	Räumlichkeiten .....	13
	Projektinfrastruktur .....	13
	Infrastruktur für Workpad Team .....	13
3.4	Risikomanagement .....	14
	R01 Workpad Projekt wird nicht rechtzeitig fertig .....	14
	R02 Technische Umsetzungsschwierigkeiten .....	14
	R03 Technologie Vielfalt .....	14
	R04 Descoping .....	15
	R05 Hardwaretechnische Limitierungen .....	15
	R06 Einbindung Office Dokumente/PDFs via XPS .....	15
3.5	Projektmonitoring .....	16
	Zeitplan .....	16
	Aufwand nach Aktivität .....	16
	Aufwand nach Entwickler .....	17
	Commits pro Teammitglied .....	18
	Fazit .....	18
3.6	Meeting-Protokolle .....	19
	Kickoff Meeting Protokoll (21.02.2012) .....	19
	Vorstudie Review Meeting Protokoll (05.03.2012) .....	20
	Status-Meeting I Protokoll (19.03.2012) .....	21
	Status-Meeting II Protokoll (02.04.2012) .....	22
	Status-Meeting III Protokoll (16.04.2012) .....	22
	Code Review I Protokoll (24.04.2012) .....	23
	Code Review II Protokoll (16.05.2012) .....	23
<b>4</b>	<b>Vorstudie</b> .....	<b>24</b>
4.1	Ausgangslage .....	24
	Ziele und Aufgabenstellung .....	25
4.2	Was ist Surface .....	25
4.3	Ausformulierung und Analyse der Anforderungen .....	26
	Verbindungsaufbau mit mobilem Endgerät .....	26

	Kommunikation mit mobilem Endgerät .....	28
	Kommunikation mit Dokument Management System.....	28
	Sicherheit auf dem iPad .....	29
	Sicherheit auf dem Surface .....	29
	Session Management und Device Tracking.....	29
	Dokument Handling auf Surface .....	34
4.4	Evaluation des mobilen Endgeräts .....	35
	Workpad.....	35
4.5	Vor- und Nachteile dieses Projektes .....	35
	Vorteile.....	35
	Nachteile .....	35
4.6	Scope dieses Projekts.....	35
	Verbindungsaufbau mit mobilem Endgerät Deliverable #278 .....	35
	Kommunikation mit dem mobilen Endgerät Deliverable #279 .....	36
	Kommunikation mit Dokument Management System Deliverable #280.....	36
	Session Management und Device Tracking Deliverable #281.....	36
	Dokument Handling auf Surface Deliverable #282 .....	36
<b>5</b>	<b>Szenario und Papier Prototyp .....</b>	<b>37</b>
5.1	Szenario .....	37
5.2	Papier-Prototyp .....	37
5.3	Papier-Prototyp Test-Resultate .....	44
	Verbesserungsvorschläge seitens des Testbenutzers .....	44
5.4	Redesign-Papier-Prototyp .....	44
<b>6</b>	<b>Technischer Bericht .....</b>	<b>52</b>
6.1	Einleitung und Übersicht .....	52
6.2	Architektur und Komponenten .....	52
	Geografische Entkopplung von Präsentations- und Applikationslogik.....	52
	Backend-Komponente Authentifizierung.....	53
	Backend-Komponente Autorisierung .....	53
	Backend-Komponente Persistence .....	54
	Backend-Komponente Ticket Store.....	56
	Backend-Komponente Session Manager.....	58
	Backend-Komponente Trusts .....	58
6.3	Funktionen .....	59
	Charmsbar .....	59
	Devicetracking .....	61
	Dokumenthandling auf Surface.....	62
	Dokumentanzeige .....	62
	Dokumentkonvertierung .....	64
	Dokument-Caching.....	65
	Dokument-Übertragung .....	65
	Anmeldeprozess .....	66
	Suspend-Mechanismus .....	67
	Abmeldeprozess .....	67
	Queueing Mechanismus.....	68
	Sessiontracking.....	68
	QR-Code .....	68
6.4	Schlussfolgerungen .....	69
	Resultate .....	69
	Probleme.....	71
	Mögliche Weiterentwicklungen .....	71

<b>7</b>	<b>Persönliche Berichte .....</b>	<b>72</b>
7.1	Marco Daniele .....	72
7.2	Michael Meier .....	73
7.3	Silka Simmen .....	74
<b>8</b>	<b>Benutzerdokumentation.....</b>	<b>75</b>
8.1	Installationsanleitung Entwicklungsumgebung.....	75
	Anforderungen: .....	75
	Backend Solution aufsetzen: .....	75
	Surface Solution: .....	75
8.2	Benutzerhandbuch .....	75
	Charms Bar .....	75
	User Anmelden.....	76
	Dokument Kategorien .....	77
	Dokumentanzeige .....	78
	Abmelden .....	80
<b>9</b>	<b>Anhang .....</b>	<b>81</b>
9.1	Glossar .....	81
	ADO.net .....	81
	.NET Framework .....	81
	Gantt-Diagramm.....	81
	Microsoft.NET.....	81
	Multitouch.....	81
	MSDN .....	81
	SDK .....	81
	WCF .....	81
	WPF .....	82
	XAML .....	82
	XML.....	82
	XNA.....	82
	XPS.....	82
9.2	Literaturverzeichnis.....	83
9.3	Aufgabenstellung Unterzeichnet.....	84
9.4	Copyright-Regelung Unterzeichnet .....	88
9.5	Erklärung über die eigenständige Arbeit.....	89

Durch die zunehmende Verbreitung mobiler Endgeräte besteht ein wachsendes Bedürfnis nach einer nützlichen Integration dieser Geräte in Business Meetings. Das Ziel dieser Arbeit ist es deshalb, Business Meetings effizienter und komfortabler zu unterstützen, indem die Teilnehmer ihre Dokumente von ihren mobilen Geräten direkt einbringen können und dann an einem digitalen Tisch untereinander präsentieren und bearbeiten können.

Dazu wurde im Rahmen dieser Studienarbeit die Lösung STEP zur Unterstützung von Business Meetings entwickelt: Ein Meeting-Teilnehmer bringt sein mobiles Gerät (z.B: iPhone oder iPad) mit den gespeicherten Dokumenten mit in die Sitzung. Mit STEP lassen sich diese Dokumente auf einfache Weise auf den interaktiven Microsoft Surface 2 Tisch übertragen und darstellen.

STEP ist so ausgelegt, dass mehrere sich gegenüberstehende Personen ein Dokument gemeinsam am Surface Table betrachten können. Dazu lassen sich mehrere Sichten auf dasselbe Dokumenten auf dem Surface öffnen, so dass jede Sicht individuell in die Perspektive eines Teilnehmers gedreht und von diesem besser gelesen werden kann.

STEP besteht aus drei Teilen. Dem Surface-Teil, welcher auf dem Surface Tisch direkt ausgeführt wird und die Benutzerinteraktion sicherstellt, dem Backend-Teil welcher die Business Logik und die Kommunikation regelt und dem mobilen Endgerät welches in dieser Lösung integriert wird.

Abschliessend finden wir, dass mit STEP eine praktische und komfortable Lösung für moderne Geschäftssitzungen auf Basis des MS Surface Table und Einbindung mobiler Geräte realisiert wurde. Es freut uns zu wissen, dass unsere Lösung von der Firma ti&m AG weiter genutzt werden wird.

### 2.1 AUSGANGSLAGE

Durch die zunehmende Verbreitung mobiler Endgeräte besteht ein wachsendes Bedürfnis nach einer nützlichen Integration dieser Geräte in Business Meetings. Das Ziel dieser Arbeit ist es deshalb, Business Meetings effizienter und komfortabler zu unterstützen, indem die Teilnehmer ihre Dokumente von ihren mobilen Geräten direkt einbringen können und dann an einem digitalen Tisch untereinander präsentieren und bearbeiten können.

Heutzutage gibt es verschiedene Varianten um an einem Meeting ein Dokument zu präsentieren. Häufig wird ein Beamer eingesetzt und der Meetingteilnehmer bringt sein Notebook, mit den entsprechenden Dokumenten, mit. Alternativ dazu können die Dokumente vorgängig ausgedruckt und auf Papier mitgebracht werden. Diese klassischen Varianten bergen aber einige Nachteile. Im einen Fall trägt der Meetingteilnehmer viele Geräte mit sich herum und muss sein Notebook erst umständlich am Beamer einrichten. Im anderen Fall sind die Dokumente vielleicht nicht mehr aktuell oder der Meetingteilnehmer hat keinen Zugriff auf weitere Dokumente, die er nicht vorgängig ausgedruckt hat.

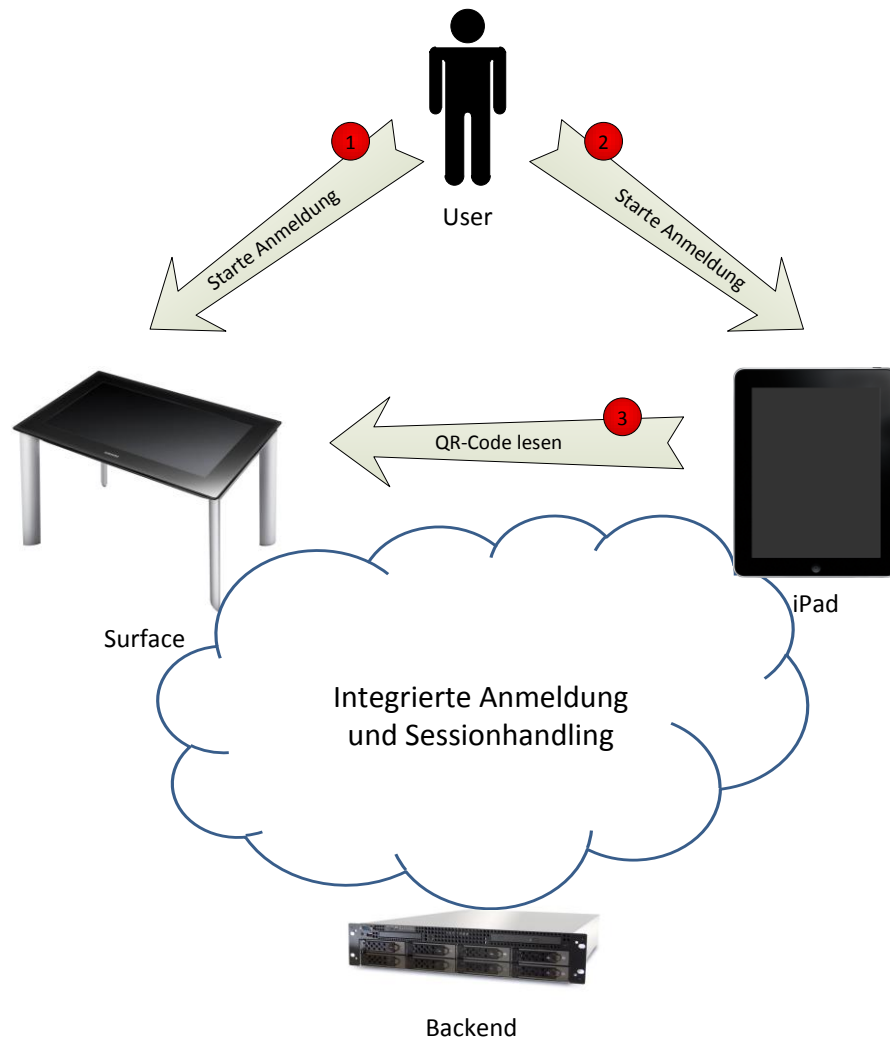
STEP bietet dafür eine Lösung. Dokumente sollen an Sitzungen effizient präsentiert werden können. Ein Meeting-Teilnehmer bringt sein mobiles Gerät (z.B: iPhone oder iPad) mit den gespeicherten Dokumenten mit in die Sitzung. Mit STEP lassen sich diese Dokumente auf einfache Weise auf den interaktiven Microsoft Surface 2 Tisch übertragen und darstellen.

Die Vorteile von STEP liegen somit auf der Hand

- Die Zusammenarbeit wird vereinfacht, nicht mehr nur einer präsentiert, die anderen Teilnehmer können dank Multitouch ebenfalls direkt die Anwendung bedienen oder eigene Dokumente anzeigen
- Reduzierte Papier-Maschine Übergänge
- Aktuelle Darstellung direkt vom Dokument, nicht von möglicherweise veralteten Snapshots
- Der Surface bietet ein grösseres Display als das mobile Endgerät

### 2.2 VORGEHEN, TECHNOLOGIEN

Da mit dem Surface 2.0 eine neue Technologie eingesetzt wurde, konnten die technischen Möglichkeiten oder Einschränkungen noch nicht genau im Voraus abgeklärt werden. Diese Punkte wurden im Rahmen der Vorstudie genau überprüft und dabei einige technische Einschränkungen des Surface festgestellt. Die Anmeldung des mobilen Gerätes zum Beispiel konnte aufgrund dieser Einschränkung nicht, wie ursprünglich angedacht, über eine optische Schnittstelle umgesetzt werden. Jedoch wurde eine gute Alternative gefunden. Der Surface stellt einen QR-Code dar, der vom mobilen Endgerät gelesen wird. Die Grafik stellt die Idee des Anmeldeprozesses schematisch dar.



Weiter haben die Abklärungen ergeben, dass es innerhalb der ti&m AG bereits ein Projekt gibt, dass eine Dokumentverwaltung für das iPad entwickelt. Daher fiel die Wahl des mobilen Endgerätes schnell auf das iPad. Es wurde beschlossen, mit dem Projekt Workpad zusammen zu arbeiten und die Synergien zu nutzen. Weiter hat sich herausgestellt, dass die Implementation einer Editierfunktion den Rahmen dieses Projekts sprengen würde, daher wird auf die Umsetzung einer solchen verzichtet. Nach der umfangreichen Vorstudie konnte die Umsetzung strukturiert und gezielt angegangen werden. Es zeigte sich, dass die Zusammenarbeit mit dem Workpad Team mehr Zeit als erwartet in Anspruch nahm, trotzdem konnte STEP erfolgreich und termingerecht umgesetzt werden.

## 2.3 ERGEBNISSE

STEP ist modular aufgebaut und besteht aus drei Teilen:

- Dem Backend, das die Sessions der mobilen Geräte und der Surface Tische verwaltet
- Dem Surface, welcher die Dokumente auf der Multitouch-Oberfläche darstellen und sich am Backend anmelden muss
- Sowie dem mobilen Endgerät das den QR-Code vom Surface fotografiert, sich am Backend anmeldet und die Dokumente an das Backend sendet

Durch den modularen Aufbau kann STEP auf einfache Art und Weise erweitert und somit auf zukünftige Bedürfnisse skaliert werden. Betreibt eine Firma mehrere Surface Tische, muss das Backend von STEP auf lediglich einem Server installiert werden und die Tische können sich am gleichen Backend anmelden.

STEP ist so ausgelegt, dass mehrere sich gegenüberstehende Personen ein Dokument gemeinsam am Surface Table betrachten können. Dazu lassen sich mehrere Sichten auf dasselbe Dokumenten auf dem Surface öffnen, so dass jede Sicht individuell in die Perspektive eines Teilnehmers gedreht und von diesem besser gelesen werden kann.

## 2.4 AUSBLICK

STEP bietet nicht nur die Umsetzung des exemplarischen Anwendungsfalles sondern bietet als Plattformlösung das Potenzial in verschiedene Richtungen weiterentwickelt oder verwendet zu werden. STEP ist eine solide Basis für Weiterentwicklungen. Dies könnte beispielsweise eine Editierfunktion sein oder die Unterstützung weiterer mobiler Endgeräte-Typen. Ebenfalls bietet STEP das Framework um eine verteilte Lösung für Multi-Homed-Meetings umzusetzen. Auch spannend in diesem Zusammenhang wäre die Integration von Kinect um zum Beispiel eine Chatfunktionen oder Site-to-Site Gestures umzusetzen.



### 3.1 PROJEKTORGANISATION

Das Vorgehen in diesem Projekt orientiert sich am agilen SCRUM-Modell. In der Rolle des Product Owners ist Stefan Pfenninger (ti&m AG). Die Rolle des Scrum Masters ist nicht vorgesehen da im Projekt kein Impediment Backlog gepflegt sondern das klassische Risikomanagement durchgeführt wird.

Ein Sprint dauert zwei Wochen Die Sprints werden zweiwöchentlich geplant. Am Ende des jeweiligen Sprints wird das Ergebnis dem Product Owner oder den Studienarbeits-Betreuern präsentiert.

Im Projektteam sind alle drei Projektmitarbeiter zu gleichen Teilen entscheidungsberechtigt. Das Team organisiert sich über die Projektmanagement Plattform agile platform welche auf Redmine basiert und vom Projektteam selbst gehostet wird. So haben alle jeweils Zugriff auf die aktuellen Dokumente, sehen was die anderen Teammitglieder erarbeitet oder geändert haben, können den Zeitplan einsehen und Arbeitszeiten erfassen.

Bei der Umsetzung wurde ausserdem darauf geachtet, dass das Layer oder Technologie-Spezifische Know-How möglichst gut auf verschiedenen Teammitglieder verteilt ist. Dies wurde durch regelmässige Pairprogramming Sessions und gegenseitigem Review und Refactoring erreicht.

#### Projektbetreuung

Das Projekt STEP wird von seitens der HSR von Herrn Prof. Dr. Luc Bläser, Herrn Silvan Gehrig und Herrn Michael Gfeller betreut.

#### Verantwortlichkeiten

Im folgenden werden die verantwortlichkeits-Bereiche im Projekt STEP genauer erläutert.

##### Surface

Dazu zählt das Erstellen der Multitouch-Benutzeroberfläche für STEP. Des weiteren muss die Darstellung von Office- und PDF-Dokumenten realisiert werden. Damit mehrere Benutzer sich auch gegenüber stehen können, muss eine Funktionalität realisiert werden, dass gezeigte Dokumente gespiegelt dargestellt werden können.

##### Backend

Zu diesem Bereich gehört die Schaffung eines Konzepts, das eine Basis bildet für eine flexible und unkomplizierte Erweiterbarkeit. Die Sessions der angemeldeten Surface Tische und der Mobile Devices sollen im Backend ressourcenschonend verwaltet werden. Die Generierung des QR-Codes gehört ebenso zu diesem Bereich wie die Bereitstellung eines Test Backends für das Workpad Team.

##### Kommunikation und Schnittstellen

Dieser Bereich beinhaltet die Evaluation und Realisierung der Kommunikation mit dem iPad. Dazu muss die Schnittstelle genaustens Dokumentiert werden, damit das Workpad Team die Applikation auf dem iPad realisieren kann. Weil es im Projekt Workpad aufgrund von Ressourcenmangel zu Verzögerungen kam, musste ein Test Mobile Client geschrieben werden, der das iPad simuliert. Weiter gehört zu diesem Bereich noch die Realisierung der Kommunikation mit dem Surface.nbsp;- Lehre als Systemtechnikerin bei Swisscom.

## Das Entwicklungsteam



### **Marco Daniele**

#### *Skills*

- Microsoft .Net C# und Microsoft Office Sharepoint Server
- Custom Solution Development
- MS SQL Entwicklung und Reporting

#### *Background*

- Abschluss der Informatikmittelschule
- Arbeit bei Skybow AG im Custom Development Bereich
- Seit 2008 Arbeit bei ti&m als Senior Software Developer

Ist im Projekt STEP verantwortlich für den Surface.



### **Michael Meier**

#### *Skills*

- Microsoft .Net C# und Microsoft Office Sharepoint Server
- Custom Solution Development
- Heterogene Netzwerke und Serverinfrastruktur

#### *Background*

- Abschluss der Informatikmittelschule
- Arbeit bei Skybow AG im Custom Development Bereich
- Seit 2008 Arbeit bei ti&m als Senior Software Developer

Ist im Projekt STEP verantwortlich für das Backend.



### **Silka Simmen**

#### *Skills*

- Operation, Fullfillment und Maintanace von Carrier Networks
- 3rd-Level Support von Internet, Voice over IP und IPTV Services
- Scripting (Perl, Ruby) im UNIX/Linux Umfeld

#### *Background*

- Lehre als Systemtechnikerin bei Swisscom
- Seit 2005 Arbeit bei Swisscom im Network Operation Center Edge

Ist im Projekt STEP verantwortlich für die Kommunikation und die Schnittstellen.

## 3.2 PLANUNG

### Status Meetings und Code Reviews

Es sollen im regelmässigen Abstand von zwei Wochen jeweils Status-Meetings mit den Betreuern durchgeführt werden. Weiter sollen zwei Code-Reviews mit den Betreuern stattfinden.

Es wurden folgende Meeting und Code-Review Termine festgelegt:

Datum	Meeting
21.02.2012	Kickoff Meeting
05.03.2012	Vorstudie Review Meeting
19.03.2012	Status-Meeting I
02.04.2012	Status-Meeting II
16.04.2012	Status-Meeting III
24.04.2012	Code Review I
16.05.2012	Code Review II

### Meilensteine

Folgende Meilensteine wurden zu Projektbeginn definiert:

Datum	Meilenstein
07.03.2012	MS00 Vorstudie und Planung abgeschlossen
16.05.2012	MS10 Code Complete
30.05.2012	MS11 Projektabschluss

### Versionen

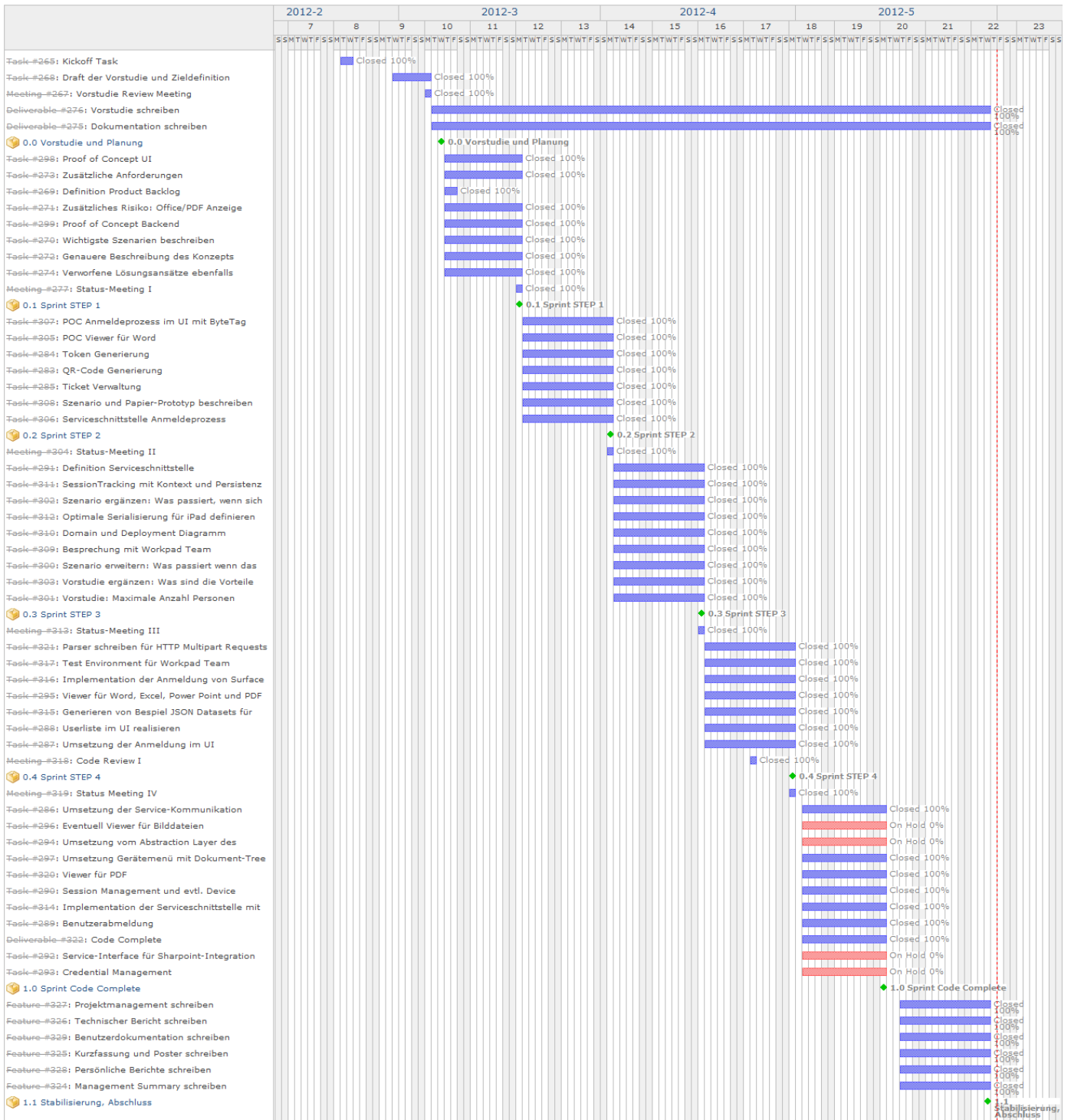
Ein Meilenstein ist erreicht, wenn die ihm zugeordneten Versionen abgeschlossen wurden. Diese Übersicht soll zeigen, welche Versionen definiert wurden und welchem Meilenstein sie zugeordnet wurden.

Datum	Meilenstein	Version(en)
07.03.2012	MS00 Vorstudie und Planung abgeschlossen	0.0 Vorstudie und Planung
16.05.2012	MS10 Code Complete	0.1 Sprint STEP 1 0.2 Sprint STEP 2 0.3 Sprint STEP 3 0.4 Sprint STEP 4 1.0 Sprint Code Complete
30.05.2012	MS11 Projektabschluss	1.1 Stabilisierung, Abschluss

### Zeitplan

Der Zeitplan ist ein dynamisches Gantt-Diagramm und wurde zwecks der Abgabe der Dokumentation an dieser Stelle statisch eingefügt. Der Zeitplan beinhaltet die Deliverables, auf die jeweils die Arbeitszeiten verbucht wurden und die einzelnen Tasks, die jeweils einem Sprint zugeordnet sind. Die Tasks wurden jeweils vor einem Sprint definiert und diesem zugewiesen. Die Deliverables decken sich mit den abzugebenden Dokumenten oder Softwarekomponenten.

- Blau dargestellt sind Tasks, die zu 100% abgeschlossen sind und auf den Status *Closed* gesetzt wurden.
- Rot dargestellt sind Tasks, die für den Sprint geplant waren und auf den Status *On Hold* gesetzt wurden. Aus Zeitgründen musste auf deren Realisierung verzichtet werden.
- Als grüne Diamanten werden Versionen dargestellt.
- Die vertikale, rote, unterbrochene Linie markiert den aktuellen Projektstand.



### 3.3 INFRASTRUKTUR

In diesem Abschnitt wird erläutert, welche Infrastrukturen für STEP benötigt wurden.

#### Räumlichkeiten

Die Arbeitsplätze der Projektmitarbeiter befinden sich in den Räumlichkeiten des Business Partners. Die regelmässigen Status-Meetings und die Code-Reviews finden in den Räumlichkeiten der HSR statt.

#### Projektinfrastruktur

Die Arbeitsgeräte, der Surface sowie das iPad werden vom Business Partner zur Verfügung gestellt. Folgende Software war für die Umsetzung von STEP erforderlich (allfällige Lizenzen wurden ebenfalls vom Business Partner zur Verfügung gestellt):

- Windows 7
- Windows 7 SDK
- Surface SDK
- Visual Studio 2010 Professional
- Diverse Tools wie
  - Sandcastle
  - TortoiseSVN

#### Infrastruktur für Workpad Team

Im Verlaufe des Projektes musste, damit das Workpad-Team möglichst unabhängig von uns und einer lokal virtualisierten STEP-Entwicklungsumgebung operieren konnte, Infrastruktur für eine Fake Surface Umgebung sowie einer Integrations-Testumgebung geschaffen werden. Das STEP Team hat zu diesem Zweck selbst einen vSphere Cluster in Betrieb genommen um die Umgebungen Surface unabhängig fürs Workpad-Team zu betreiben.

### 3.4 RISIKOMANAGEMENT

Dieses Dokument beschreibt die wichtigsten Risiken, die für dieses Projekt identifiziert werden konnten. Die Risiken, deren Auftretenswahrscheinlichkeit als unwahrscheinlich klassifiziert wurden, wurden in diesem Dokument weggelassen, daher ist die Auflistung nicht vollständig. Die Risiken werden im Rahmen der Vorstudie erstmals identifiziert, im Projektverlauf anschliessend aber kontinuierlich neu bewertet oder neue Risiken identifiziert.

#### **R01 Workpad Projekt wird nicht rechtzeitig fertig**

Das Projekt Workpad wird nicht rechtzeitig fertig mit der Implementierung des iPad Clients. Dies würde bedeuten, dass die Projektziele grundsätzlich erreicht werden könnten, jedoch fehlt ein Wichtiger Teil in der Gesamtlösung. Es könnte nie getestet oder auch präsentiert werden, ob die ganze Integration vom mobilen Gerät mit dem Surface funktioniert. Daher ist die Auswirkung dieses Risikos kritisch.

Um dieses Risiko zu verringern und um beurteilen zu können, wie wahrscheinlich der Eintretensfall aktuell ist, finden regelmässig Statusmeetings statt, wo der aktuelle Stand von Workpad diskutiert wird.

*Klassifizierung: Show Stopper*

#### **Updates:**

- 06.03.12: Damit die Backend und Surface Komponenten unabhängig vom Workpad Fortschritt präsentiert werden kann, könnte man als Massnahme die mobile Anwendung simulieren.
- 28.03.12: Das Workpad-Team ist bis am 09.04.12 in den Ferien und wir erhalten bis dahin kein Feedback von ihnen. Das Risiko, dass sie nicht rechtzeitig fertig werden, hat sich damit erhöht. Als Massnahme haben wir die Schnittstellen zu Workpad ausgearbeitet, damit wir sie, wenn sie aus den Ferien zurück sind, unvorzüglich mit Informationen versorgen können. Sie sollten dadurch in der Lage sein, sofort mit der Implementation zu beginnen. Direkt nach Ferienende wurde ein Termin mit ihnen vereinbart.

#### **R02 Technische Umsetzungsschwierigkeiten**

Eine technische Hürde taucht auf, die im Rahmen der Vorstudie nicht als Problem erkannt wurde. Je nach Art der technischen Schwierigkeit, kann diese Hürde das Projekt komplett blockieren oder nur eine kleine Einschränkung darstellen.

*Klassifizierung: Unkritisch bis Show Stopper*

#### **R03 Technologie Vielfalt**

Durch den Einsatz des iPads als Client-Geräts und Workpad als Lösung, ergibt sich ein Technologiestack, der nicht nur Objectiv-C sondern auch Java umfasst. Dies erhöht zusätzlich zur hauptsächlich Eingesetzten Microsoft Technologie mit .NET, Surface mit WPF und Sharepoint die Komplexität.

*Klassifizierung: Moderat*

## **R04 Descoping**

In der ursprünglichen Aufgabenstellung wurde eine Bearbeitungs- und Speicher-Funktionalität gefordert. Unsere Analyse hat ergeben, dass dies nur sehr aufwändig oder für ein Touch-Umfeld unzureichend, zu realisieren wäre und den Rahmen der Studienarbeit sprengen würde. Siehe [Vorstudie](#).

Das Risiko besteht darin, dass dieses Descoping von den Betreuern nicht akzeptiert wird. Um dieses Risiko zu minimieren, werden wir dies an der nächsten Sitzung besprechen.

*Klassifizierung: Show Stopper*

### **Update:**

- 03.04.12: Am Meeting vom 05.03.2012 haben die Betreuer ihr OK zum Descoping gegeben. Damit ist dieses Risiko eliminiert.

## **R05 Hardwaretechnische Limitierungen**

Der Surface ist aufgrund seiner empfindlichen Charakteristik nur unter sehr guten Bedingungen einsetzbar. Er reagiert sehr empfindlich auf die Umgebungsbeleuchtung. Im schlechtesten Fall ist der Surface nicht bedienbar. Das Risiko für STEP ist dabei, dass der Surface nur unter künstlichen Bedingungen bedienbar ist. Vorallem bei der Schlusspräsentation wäre ein solches Verhalten sehr ungünstig.

*Klassifizierung: Kritisch*

## **R06 Einbindung Office Dokumente/PDFs via XPS**

06.03.12

Gemäss Erfahrungen von Silvan Gehrig und Michael Gfeller, kann die Einbindung von Office/Pdf Dokumenten zu Schwierigkeiten in der Präsentation führen. Konkret ist das Look & Feel bei grösseren Dateien nicht mehr so flüssig wie man es von Touch-Anwendungen gewohnt ist.

*Klassifizierung: Kritisch*

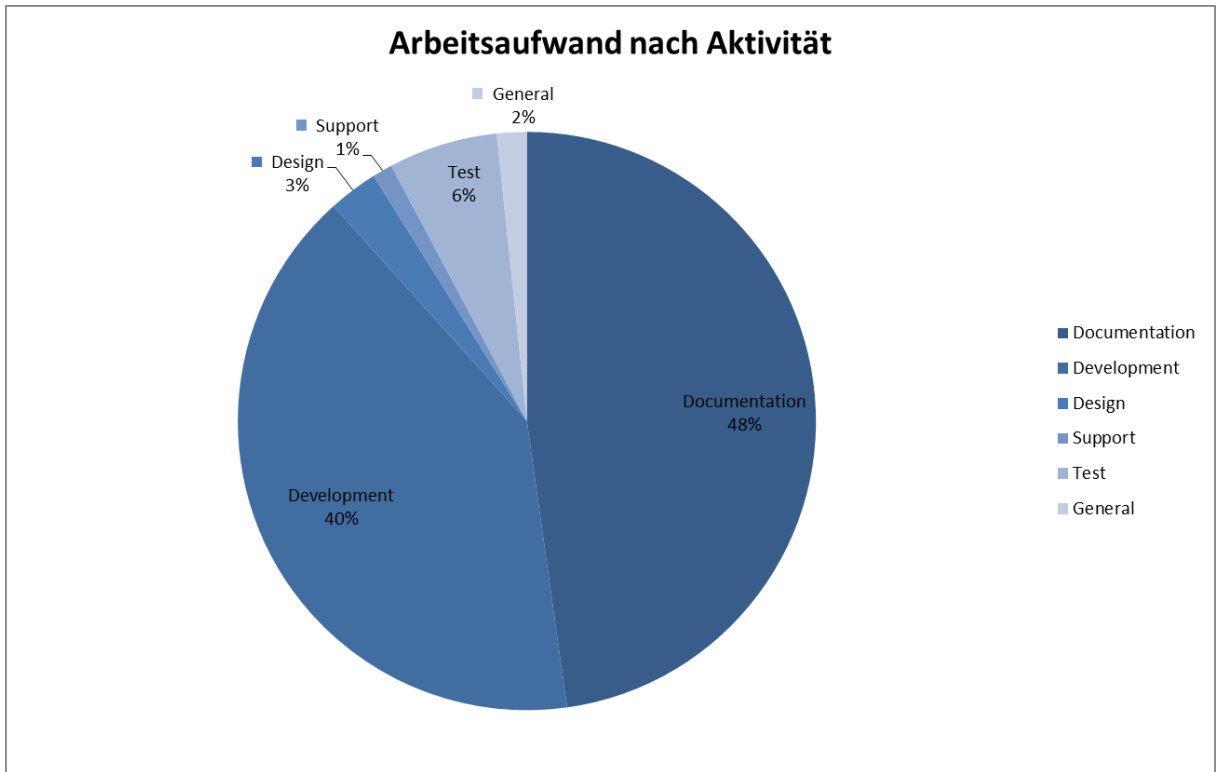
### 3.5 PROJEKTMONITORING

Für das Projekt STEP wurden insgesamt 720 Stunden aufgewendet.

#### Zeitplan

Der Zeitplan ist im Kapitel Planung dokumentiert.

#### Aufwand nach Aktivität

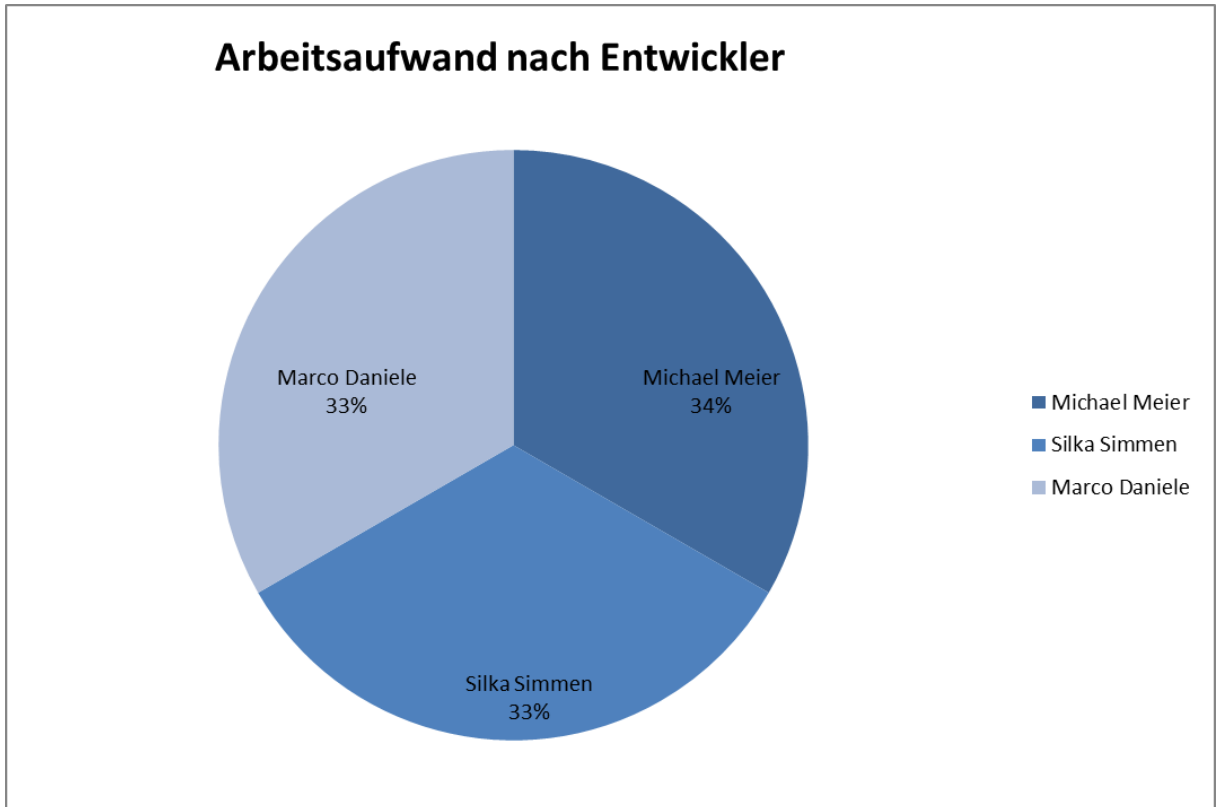


Aktivität	Aufwand in Stunden
<b>Documentation</b>	344
<b>Development</b>	292
<b>Test</b>	44
<b>Design</b>	20
<b>General</b>	12
<b>Support</b>	8
<b>Total</b>	<b>720</b>

Am meisten Zeit, 344 Stunden, wurde für die Aktivität **Dokumentation** aufgewendet, davon wurden 200 Stunden alleine in die Vorstudie investiert. Die restlichen Stunden wurden für die verbleibende Dokumentation verbucht wobei ein weiterer Teil explizit für die Schnittstellen-Dokumentation zum mobilen Gerät aufgewendet wurde. Danach folgt die Aktivität **Development** mit 292 Stunden, diese Aktivität beinhaltet jegliche Arbeit am Code. Design Entscheidungen und Besprechungen mit dem Workpad Team bezüglich der Kommunikation mit dem iPad wurden der Aktivität **Design** zugewiesen. Zur Aktivität **Test** gehören Arbeiten wie das Test Environment für das Workpad Team aufsetzen sowie STEP testen. Die Code Reviews wurden auf die Aktivität **General** verbucht und Hilfestellungen für das Workpad Team auf die Aktivität **Support**.



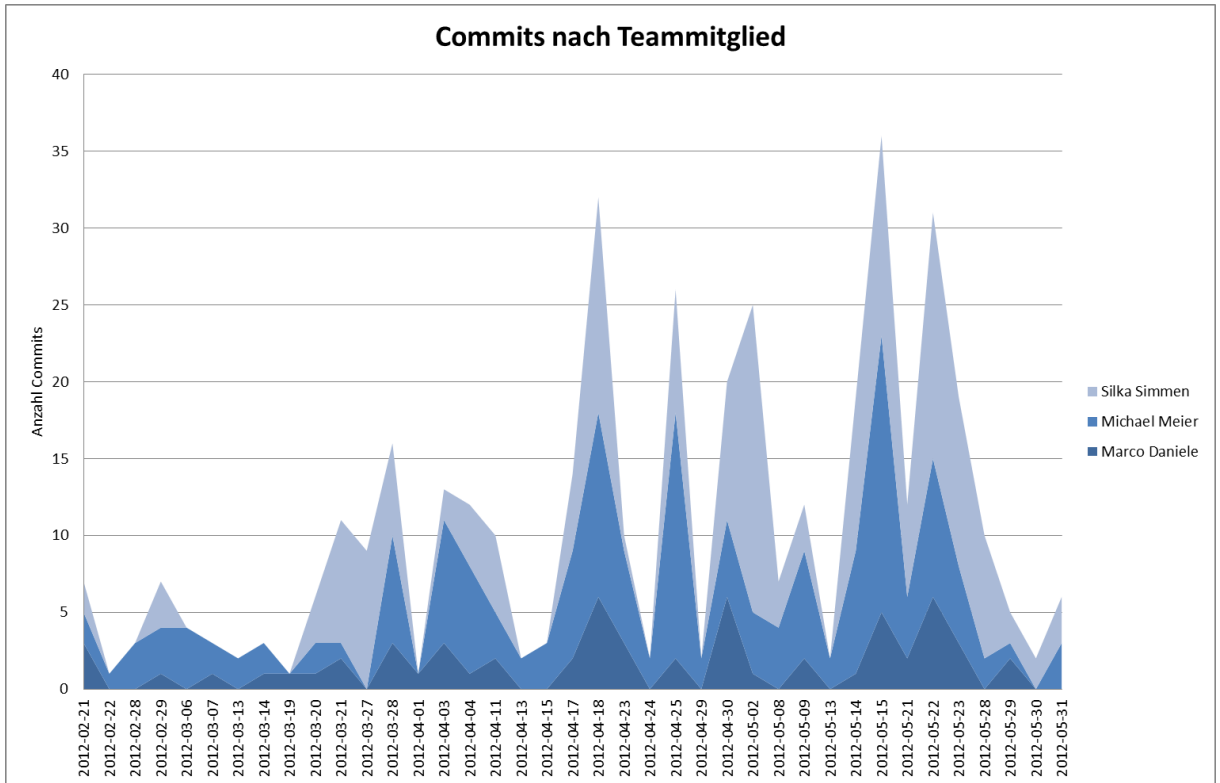
## Aufwand nach Entwickler



Entwickler	Aufwand in Stunden
Michael Meier	240
Silka Simmen	240
Marco Daniele	240
<b>Total</b>	<b>720</b>

Die geleisteten Stunden sind gleichmässig auf die Entwickler verteilt, weil das Team immer gemeinsam an STEP gearbeitet hat. Daher hat jeder Entwickler die gleiche Anzahl Stunden verbucht.

## Commits pro Teammitglied



In der Grafik sind die getätigten Commits über die Projektdauer pro Entwickler ersichtlich.

## Fazit

Die Zusammenarbeit mit dem Workpad Team hat mehr Zeit in Anspruch genommen als ursprünglich geplant. Ausserdem wurde die Implementation der Features auf vom Workpad Team aufgrund von Ressourcen-Problemen eher knapp terminiert, was von unserer Seite die Implementation eines Mobile Device Simulators nötig machte. Trotz diesem Umstand konnten die geplanten Termine gut eingehalten werden. Am Ende des Projekts musste ein Tag mehr als ursprünglich geplant für die Dokumentation aufgewendet werden.

### 3.6 MEETING-PROTOKOLLE

Dieser Abschnitt dokumentiert die Protokolle der Meetings und Code Reviews. Anzumerken ist dabei, dass das auf dem Zeitplan ersichtliche Status Meeting IV nur eine Demonstration der bisherigen Lösung beinhaltet. Deshalb existiert dazu kein Protokoll.

#### Kickoff Meeting Protokoll (21.02.2012)

##### Abgabe

- Termin Vereinbarung mit allen Betreuern
- Demonstration (Präsentation) bei TI&M
- Auf CD (an SE2P orientieren):
  - Wenn möglich ein PDF Dokument
  - Video (Demonstration)
  - Deliverables (an [HSR Dokument](#) orientieren)
  - Projektplan
  - Anforderungen
  - Architekturdokumentation
  - Storyboard
  - Risikoanalyse (fortlaufend ergänzt, angepasst)
  - Automatisierte Tests
  - Es muss ersichtlich sein, wer was gemacht hat
- Kurzfassung
- A0-Poster

##### Termine

- 20.02.12 - Beginn Studienarbeit
- 29.05.12 - Kurzfassung und A0 Poster an die Betreuer schicken (E-Mail)
- 01.06.12 - Kurzfassung (von den Betreuern freigegeben) an cfurrer@hsr.ch schicken
- 01.06.12 - Abgabe des Berichts an die Betreuer bis 17:00

##### Regelmässige Meetings

- Alle zwei Wochen
- Einladungen dafür verschicken (Teilnehmer: Projektteam, Stefan Pfenninger, Michael Gfeller, Silvan Gehrig)
- Meeting Einladung mit Agenda (spätestens am Abend vorher)
- Kann auch via Telefonkonferenz durchgeführt werden

##### Code Reviews mit Betreuern

- Sollte zwei mal stattfinden

##### TODOs bis 28.2.

- Ziele definieren
- Grobe Planung erstellen (Meilensteine setzen, Wann ist die Vorstudie abgeschlossen?)
- Risikoanalyse
- Termine für Code Reviews planen (2.5.12;16.5.12)
- Termine für Meetings planen (5.3.12;19.3.12(Konflikt);2.4.12;16.4.12(Konflikt);2.5.12;16.5.12)
- Termin Vereinbarung für Abschlusspräsentation mit allen Betreuern (Mit Stefan absprechen, wann)

## **Vorstudie Review Meeting Protokoll (05.03.2012)**

### **Vorstudie besprechen**

- High PrioSzenarien abdecken, so früh wie möglich. Issue #270
- Zusätzliches Risiko Einbindung office/pdf xps hat bereits zu Schwierigkeiten geführt. Erfahrung von Silvan und Michael. Issue #271
- Erwähnen das Server Teil auf Surface, beachten, dass der Surface auch vom Netz getrennt sein könnte. Muss abgegrenzt werden oder genauer erklärt. Issue #272
- Evtl. trennen Surface als Präsentation und Webserver-Teil. Server Architektur evtl. genauer abgrenzen. Issue #275
- Rückspeicherung sollte möglich sein. Issue #273
- Sharepoint Workflow optional einbauen. Issue #273
- Bilder Skizzen zusätzliche als pdf/office trio anbieten? Issue #273
- Idee Kamera Extern? Ebenfalls notieren. Issue #274
- Möglichkeiten, welche abgelehnt wurden ebenfalls in Dokumentation aufnehmen(evtl. für den Technischen Bericht). Issue #274

### **Formale Fragen zum Abgatedokument**

- Finales Dokument evtl. in HTML möglich, PDF in Bachelor zwingend. Silvan klärt ab, ob in den Reglementen was steht.
- Antwort von Michael: HTML geht für die SA in Ordnung. Nur ein Dokument wäre aber wünschenswert (=> BA muss dann so sein).

### **Orientierung über Workpad Projekt Einbezug**

Ist in Ordnung, es muss einfach klar erkennbar sein, was von uns realisiert wurde und was nicht.

### **Orientierung über Descoping**

Ist in Ordnung.

### **Termine Fixieren**

Termine Request versenden. Abschlusspresentation kann auch eine Woche nach offizieller Abgabe stattfinden, vorgeschlagener Termin geht für Luc Bläser nicht.

### **TODOs bis zum nächsten Meeting (19.03.2012)**

- Aufgabenstellung vorbereiten zum unterschreiben.
- Termin Requests für regelmässige Meetings und Code Reviews versenden.
- Termin Vereinbarung für Abschlusspräsentation mit allen Betreuern (Mit Stefan absprechen, wann)

## Status-Meeting I Protokoll (19.03.2012)

### Termin für Abschlusspräsentation fixieren (Vorschlag ab 18.06.2012)

- Abgabe am 1.6. 18.06. ist OK für Abschlusspräsentation, sie zählt dann einfach nicht mehr in der Endbewertung.
- Luc hat dann auch Zeit

### Erste Implementationsarbeiten präsentieren

Erledigt

### Änderungen an der Vorstudie zeigen

- Erledigt
- Ergänzung: Vorstudie wird in technischem Bericht abgehandelt (Vorstudie ist technischer Bericht)

### Szenario zeigen

- Passwortschutz? Was passiert wenn das iPad geklaut wird? Hat man dann Zugriff auf den Sharepoint? Issue #300
- Maximal Personen definieren in der Vorstudie (maximal 8?) Issue #301
- Usability? Was passiert, wenn sich zwei Personen am Surface gegenüberstehen? Feature Idee: Spiegelmodus. Issue #302
- Was ist sind die Vorteile gegenüber einem Beamer? Was sind die Nachteile? In der Vorstudie beschreiben: Rechtfertigung der Lösung und evtl. Vorteile des Architekturansatzes, was könne damit später noch gemacht werden? Kinect mit Beamer? Issue #303

### Allgemeines Feedback

- SSD kreieren wenn viel Interaktion passiert. (in Architekturdokumentation)
- Domain Modell: nicht zu viel auf Technik eingehen
- Domain Modell: dokumentieren, was was bedeutet mit Legende (wenn gemischt wurde)
- Tests für den Backend-Bereich schreiben
- Workpad Risiko minimieren: evtl. Softclient schreiben, wenn absehbar ist, dass Workpad Verzögerung hat

### TODOs bis zum nächsten Meeting (02.04.2012)

- Termin für Abschlusspräsentation fixieren (Es fehlt noch die Bestätigung der Raumreservation)
- Raum jeweils reservieren für Meeting. (Dafür haben wir als Studenten keine Berechtigung)
- Erstes Code Review vorverschieben, damit man noch genug Zeit zum Reagieren hat.
- Erwähnen, dass Coding Conventions von SE2Projekt übernommen wurden.
- Zeiterfassung, damit nachvollziehbar ist wer was gemacht hat. (Tageseinheiten)
- Aufgabenstellung mitbringen (haben wir zwar mitgebracht, wurde aber vergessen abzugeben)

## Status-Meeting II Protokoll (02.04.2012)

Papierprototyp demonstriert

Bisherige Implementation demonstriert

### Feedback:

- SSD's wo sinnvoll erstellen
- Architekturdokumentation mit Domainmodell beginnen
- Risikomanagement überarbeiten
- Zeiterfassung erfolgte zu knapp vor dem Meeting
- Allgemeine Zufriedenheit, wir arbeiten weiter an der Implementation

### TODOs bis zum nächsten Meeting (16.04.2012)

- Termin für Abschlusspräsentation fixieren und Einladungen dazu verschicken
- Aufgabenstellung mitbringen zum unterschreiben
- Zeit erfassen
- 
- Architekturdokumentation beginnen zu schreiben

## Status-Meeting III Protokoll (16.04.2012)

Architekturdokumentation gezeigt

Schnittstellendokumentation gezeigt

Applikation demonstriert

### Feedback:

- Gerichtete Pfeile gefallen (zeigen Abhängigkeiten gut)
- Domainmodell OK
- Wenn etwas nicht UML Standard ist -> beschreiben
- Gedanken zum Szenario dürfen in die Architekturdokumentation verschoben werden

### TODOs:

- 30.04. Meeting einplanen ab 15:00 (mit Luc Bläser)
- Design: Beschreibung hinzufügen, was ist es für ein Diagramm
- Später Sequenzdiagramme hinzufügen für schwierig zu verstehende Abläufe

## Code Review I Protokoll (24.04.2012)

### Feedback:

- Beschreibung auf Interface nicht auf Service Klasse (MobileDeviceService)
- Beschreiben warum Zebra Crossing Adapter mit eigener Dokumentation
- Unbedingt beschreiben, dass Xing von Google kommt (Quellenangabe und Abgrenzung)
- Warum kein Entity-Framework eingesetzt? Ebenfalls in der Architekturdokumentation beschreiben. -> Lightweight, Firmenpolitik
  - Entity-Framework wäre die bessere Wahl gewesen
  - Entweder wechseln zu Entity-Framework oder genau Begründen, warum dies nicht eingesetzt wurde
- Coding Guideline nicht eindeutig für Einrückung: Fall für using definieren.
- Charmbar: Zwei identische Buttons -> mit Style nur einmal definieren
- add.png: Items als Ressourcen definieren. (Freezable)
- Sleep in einem Unit-Test: Begründen warum, oder synchron aufrufen
- Ticket Validation: Begründn, warum die Validate Methode nicht auf dem Ticket selber ist.
  - das zweite if Statement in der Validate Methode im Ticket Store muss zum Ticket selber verschoben werden. So kann diese Methode direkt im Unit Test verwendet werden
- Readonly bei locks (Bei Singletons)
- Maximal ein Singleton pro Applikation: warum werden mehrere eingesetzt?
- Absolute Pfade entfernen
- Property Gross- und Kleinschreibung (Transfer)
- FindParent anwenden, anstatt den Visual Tree fix "hinaufzu gehen"
- Bindings verwenden
- Guidelines: \_var wird gefordert in den Guidelines, aber in Code Beispielen wird gleich das Gegenteil gemacht
- Eventhandler gross schreiben
- Events werden nicht abgemeldet
- Besser Dokumentieren
- Nicht zu viele Features auf's mal implementieren (nur das was definiert wurde)
- Dokumentieren, wie das mit dem Java Connector funktioniert

## Code Review II Protokoll (16.05.2012)

- Suspend Logout Behaviour genauer erläutern
- Dokumentation im Nachhinein ist gefährlich, Konzept im Vorhinein besprechen mit Assistenten
- Viele Dinge nicht prüfbar, weil Dokumentation unzureichend
- Visibility mit Converter lösen
- App.xaml ist im falschen Namespace
- Anleitung zum Installieren oder ein Installer fehlt
- View Model sollte nicht im UI sein
- Event mit Underline oder ohne (Coding Guideline anpassen)
- Datakontext ungleich Model

Da eine neue Technologie eingesetzt wird, konnten die technischen Möglichkeiten oder Einschränkungen noch nicht genau abgeklärt werden im Voraus. Aus diesem Grund wurde die Aufgabenstellung zu diesem Projekt offen formuliert. Diese Vorstudie soll die in der ursprünglichen Aufgabenstellung enthaltenen Anforderungen analysieren. Dabei soll abgeklärt werden, ob die Anforderungen technisch grundsätzlich umsetzbar sind oder ob in der Neuausrichtung der Arbeit, zugunsten einer Alternative, darauf verzichtet werden soll. Am Ende der Vorstudie soll der Scope dieses Projekts genau definiert werden.

#### 4.1 AUSGANGSLAGE

Mit Hilfe des interaktiven und digitalen Microsoft Surface 2 Tisches sollen Business Meetings in Bezug auf gemeinsame Bearbeitung von Dokumenten effizient unterstützt werden. Ein Meeting-Teilnehmer soll seine Dokumente auf dem Tisch eines Sitzungsraums darstellen können, indem er zuerst sein Mobiltelefon mit dem entsprechen geöffneten Dokument auf den Tisch legt. Der Tisch hat im Backend Zugriff auf ein Document Management System wie zum Beispiel MS Sharepoint. Die Dokumente können dann gemeinsam am Tisch gelesen, bearbeitet und mit Notizen vermerkt werden. Am Schluss werden die Änderungen an den Dokumenten wieder zurück im Document Management System zurückgespeichert.

Um diesen Ablauf zu realisieren, bedarf es der Erarbeitung von Lösungen in verschiedenen Aspekten:

1. *Anzeigen von Dokumente vom Mobiltelefon am Tisch:* Konzeption und Umsetzung eines geeigneten Vorgehen, um Dokumente vom Mobilgerät am Tisch öffnen zu können, z.B. durch direkte Übertragung mit Bluetooth, durch optische Erkennung des Mobiltelefons und anschließenden Instruktionen zum Upload an ein Document Management System des Tisches bzw. andere Ansätze.
2. *Meeting Session Programm:* Entwicklung einer Anwendung auf dem Surface Table, welche die Durchführung einer Sitzung mit Dokumenten leitet. Dies umfasst das Starten einer Sitzung, das Uploaden/Auswählen von Dokumenten, Darstellung der Dokumente in geeigneten Viewers oder Editors, das Abspeichern der Dokumente und schliesslich das Beenden der Sitzung.
3. *Office und PDF Viewer am Tisch:* Einsatz von StandardAnwendungen am Tisch, um die Dokumente anzeigen und bearbeiten zu können sowie die Integration dieser in das Meetings Session Programm.
4. *Document Management System:* Verwaltung und Zugriff von Dokumenten in einem bestimmten Document Management System im Backend, welcher unter anderem vom Tisch zugegriffen wird.



## Ziele und Aufgabenstellung

Die Aufgabe dieser Arbeit ist es, eine Lösung für das Meeting Management am Microsoft Surface 2 Tisch zu entwickeln.

Folgende spezifische Ziele werden vorgegeben:

- Aufnahme der Anforderungen für das Meeting Management basierend auf dem Surface 2 Tisch.
- Entwurf eines Lösungskonzeptes zur Umsetzung des Meeting Ablaufs mit Hilfe von dem Surface 2 Tisch (Abläufe, technische Analyse, erster Prototyp)
- Entwicklung der Anwendung für den Surface 2 Tisch, welche den Meeting Ablauf umsetzt. Dies umfasst insbesondere das Öffnen der Dokumente am Tisch mit dem Mobiltelefon, das Darstellen und Bearbeiten der Dokumente am Tisch sowie das Abspeichern am Schluss.
- Integration aller nötigen Komponenten (MobileSurface Interaktion, Meeting Session Programm, Document Management System) zur funktionsfähigen Gesamtlösung.

## 4.2 WAS IST SURFACE

Surface ist ein Produkt der Firma Microsoft, welche ohne Maus, Tastatur oder sonstige herkömmliche Eingabegeräte auskommt. Sämtliche Eingaben werden mit der Hand (bzw. mehreren Händen) auf dem Bildschirm vorgenommen. Diese sogenannte Multi-Touch-Technologie erlaubt Interaktionen zwischen Mensch und Computer oder auch zwischen Computer und anderen physischen Gegenständen. Die zweite Version des Surface, SUR40, wurde in Zusammenarbeit mit Samsung entwickelt. Der Surface 2.0 besitzt nun einen 40" 1080p LCD HD Bildschirm, verwendet als Prozessor wird ein AMD Athlon II X2 und als Grafikkarte kommt eine Radeon HD 6700M zum Einsatz. Als Betriebssystem wird Windows 7 eingesetzt. Im Gegensatz zum Surface 1.0 werden keine Infrarotkameras verwendet, sondern die sogenannte Pixelsense-Technologie wird eingesetzt. Pixelsense bedeutet, dass in den Pixels des Bildschirms Sensoren eingebaut sind, die auf infrarot Licht reagieren. Ein Objekt, das auf den Surface gelegt wird, wird von Surface mit infrarot Licht beleuchtet und reflektiert dieses. Die in den Pixeln eingebauten Sensoren registrieren das reflektierte Licht. Die Informationen aller Sensoren erlauben es Surface ein Bild der auf den Tisch gelegten Objekte zu errechnen. Leider ist diese Technologie sehr lichtempfindlich und bezüglich Erkennung im Vergleich zum Surface 1.0 ein Rückschritt.

Das Surface-SDK basiert auf Microsofts aktuellen Technologien WPF 4.0 und XNA 4.0. Diese unterstützen moderne Touch-Anwendungen mit Windows 7 in der 32- oder 64-Bit-Version.

*Quelle: Surface 2.0 SDK - Architecture (<http://msdn.microsoft.com/en-us/library/ff727809.aspx>)*

In den Dokumentationen zum Projekt wird die Terminologie 'der Surface' verwendet als Bezeichnung für das Produkt Microsoft Surface 2. Das beinhaltet die Hardware und Software, so wie das Produkt als Ganzes vom Hersteller geliefert wurde.

### 4.3 AUSFORMULIERUNG UND ANALYSE DER ANFORDERUNGEN

Für diese Analyse werden die einzelnen Schritte für den in der Aufgabenstellung definierten Anwendungsfall auf die technologische Umsetzung hin abstrahiert.

#### **Verbindungsaufbau mit mobilem Endgerät**

Damit STEP verwendet werden kann, muss mindestens ein Benutzer sein mobiles Gerät mit dem Surface verbinden. Idealerweise wird der Verbindungsaufbau so realisiert, dass dieser eine möglichst geringe Benutzerinteraktion erfordert.

Damit der Anmeldeprozess mit einem Minimum an Benutzerinteraktion sowie vorgängigen Administrationsaufgaben auskommt, wird die Authentifizierung mittels eines Schlüsselaustausch-Verfahrens durchgeführt.

Der Client, in diesem Fall das mobile Endgerät, erzeugt bei der Installation der Clientapplikation oder dem ersten Verbindungsaufbau mit einem Surface, einen eindeutigen Schlüssel. Dieser wird für die Authentifizierung an den Surface übermittelt. Der Surface speichert und verwaltet alle Client-Schlüssel. So können noch unbekanntem Geräten Gast-Rollen zugeordnet werden. Einzelne Geräte können explizit zusätzliche Berechtigungen erhalten oder komplett blockiert werden.

Damit der Surface auf die Daten zugreifen kann, die der Benutzer im Dokument Management System gespeichert hat, müssen diese Credentials, die auf dem mobilen Gerät hinterlegt werden, ebenfalls an den Surface übermittelt werden. Das Wichtigste dabei ist, dass diese Credentials sicher übermittelt werden und auf keinen Fall Drittpersonen zugänglich gemacht werden. Die Übermittlung dieser Credentials geschieht in einem späteren Schritt, nach der Anmeldung.

Es existieren mehrere Varianten, wie der eindeutige Schlüssel des Clients und die Credentials an den Surface übermittelt werden könnte:

#### **Mobiles Gerät liest QR-Code auf Surface**

Der Surface bietet dem Client initial eine Webservice-URL, sowie eine One-Time-Ticketnummer an, welche es dem Client erlaubt, innerhalb einer bestimmten Zeit, einmalig eine Anmeldung vorzunehmen. Diese URL, sowie die One-Time-Ticketnummer, werden als QR-Code auf dem Surface dargestellt, sobald ein Benutzer ein mobiles Gerät anmelden möchte. Das mobile Gerät muss also nur den QR-Code lesen und sich auf die Webservice-URL verschlüsselt verbinden. Über diesen Webservice kann das mobile Gerät dann später die Credentials, die zur Anmeldung am Dokument Management System erforderlich sind, übermitteln. Dieses Verfahren gewährleistet, dass der Benutzer physisch anwesend sein muss und verhindert gleichzeitig Replay Attacks.

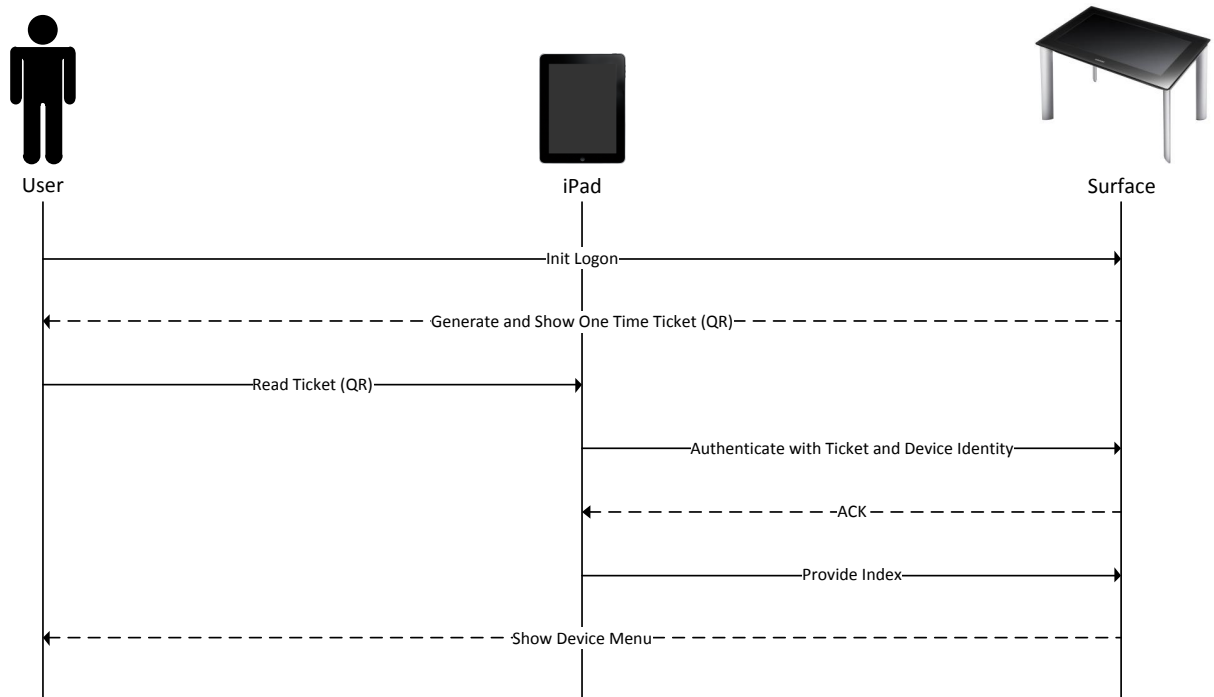
*Quelle und weitere Informationen zu QR-Codes: <http://www.denso-wave.com/qrcode/index-e.html>*

Vorteile:

- Die Credentials können verschlüsselt via Webservice übermittelt werden.
- Es ist sichergestellt, dass der Benutzer physisch anwesend ist.

Nachteile:

- Es sind drei Aktionen nötig: Der Benutzer muss die Client-Applikation starten und die Anmeldung initiieren. Dann muss das mobile Endgerät über den QR-Code auf dem Surface gehalten werden um diesen zu lesen.
- Ein Webservice muss laufen, um die Surface Interaktionen durchzuführen.
- Das mobile Endgerät und die Backend-Applikation, wo der Webservice zur Verfügung gestellt wird, müssen sich via IP-Protokoll gegenseitig erreichen können. Dies ist eine Voraussetzung dieser Variante.



### Surface liest mittels Webcam QR-Code auf mobilem Gerät

Eine weitere Variante ist der Einsatz einer Webcam am Surface, um den Client-Schlüssel vom Display des mobilen Geräts zu lesen. Der Schlüssel würde in dieser Variante als QR-Code auf dem Display des mobilen Geräts dargestellt werden. Die Credentials werden ebenfalls via QR-Code dargestellt und so dem Surface in einem späteren Schritt übermittelt. Dies ist sogleich der grösste Nachteil dieser Variante: Die Übertragung der Credentials für das Dokument Management System via QR-Code ist heikel, auch wenn sie verschlüsselt geschieht. Ein weiterer Nachteil ist, der Einsatz einer externen Webcam, die mit dem Surface verbunden und installiert werden muss. Dabei erscheint vorallem die Benutzerfreundlichkeit unzureichend: entweder ist die Kamera fix am Surface installiert und der Benutzer muss das mobile Gerät zur Kamera bewegen um sich anzumelden oder er muss die lose angeschlossene Kamera zum mobilen Gerät bewegen.

Quelle und weitere Informationen zu QR-Codes: <http://www.denso-wave.com/qrcode/index-e.html>

Vorteile:

- Es ist sichergestellt, dass der Benutzer physisch anwesend ist.

Nachteile:

- Es sind drei Aktionen nötig: Der Benutzer muss die Client-Applikation starten und die Anmeldung initiieren. Die Webcam des Surface so positioniert werden, dass sie den QR-Code auf dem Display des mobilen Geräts lesen kann.
- Die Credentials via QR-Code übermitteln ist auch wenn sie verschlüsselt sind, eher unsicher. Es kann einfach von Dritten mitgehört werden.
- Zusätzliche Hardware muss beim Surface installiert werden.
- Es ist keine "offline" Übertragung von Dokumenten möglich, da keine direkte Verbindung zwischen dem mobilem Endgerät und dem Surface aufgebaut wird.

### Automatische Anmeldung

Bei dieser Variante sucht das mobile Endgerät automatisch Webservices, über welche mit einem Surface kommuniziert werden kann. Dabei verschickt die Client-Applikation IP-Broadcasts um

entsprechende Webservices zu finden. Ist ein Webservice gefunden, werden diesem der Schlüssel des Clients, sowie die Credentials übermittelt.

Vorteile:

- Der Benutzer muss nur die Client-Applikation auf dem mobilen Gerät starten, die Anmeldung geschieht dann automatisch.
- Die Credentials können verschlüsselt via Webservice übermittelt werden.

Nachteile:

- Es kann nicht sichergestellt werden, dass der Benutzer physisch anwesend ist.
- Dem Benutzer ist nicht ersichtlich und er hat keinen Einfluss darauf, wohin seine Credentials verschickt werden.
- Ein Webservice muss laufen, um die Surface Interaktionen durchzuführen.
- Das mobile Endgerät und die Backend-Applikation (wo der Webservice läuft) müssen sich via IP-Protokoll gegenseitig erreichen können. Dies ist eine Voraussetzung dieser Variante.

Die Variante, wo der Surface einen QR-Code mit der Webservice-URL erzeugt und das mobile Gerät diesen einliest, erscheint unter den Aspekten der Benutzerfreundlichkeit und der Sicherheit am Besten geeignet.

## **Kommunikation mit mobilem Endgerät**

Die Kommunikation mit dem mobilen Endgerät muss folgende Kriterien erfüllen:

### **Sicherheit**

Die Kommunikation muss verschlüsselt erfolgen, so dass unabhängig vom verwendeten Transportkanal auch sensitive Daten, wie die Credentials für das Dokument Management System, übermittelt werden können.

### **Bandbreite**

Die Verbindung sollte in der Lage sein ein Dokument bis 2MB innerhalb von einer Sekunde zu Übertragen. Daraus resultiert eine Idealbandbreite von mindestens 16Mbit/s

### **Verbindung und Session**

Die Kommunikation sollte auf bereits Existierenden Mechanismen für Verbindungsorientierte Kommunikation verwenden. Also im Idealfall auf TCP over IP basieren.

Durch den Einsatz von WLAN und Webservices könne die obigen Anforderungen erfüllt werden.

## **Kommunikation mit Dokument Management System**

Während eines Meetings sollen Dokumente von einem Dokument Management System verwendet werden können. Unser Business Partner ti&m AG zieht hier Share Point 2010 als System vor.

Grundsätzlich gibt es zwei Verschiedene Implementationsmöglichkeiten für den Zugriff von Surface auf das Dokument Management System.

### **Clientbasiert**

In diesem Fall wird das Dokument Management System vollumfänglich auf dem mobilen Client eingebunden. Das Clientgerät alleine hat das Dokument Management System im Zugriff und verwaltet auch selbständig die nötigen Credentials sowie das Loginprozedere.

Ein Dokument wird dann für ein Meeting via Client-Applikation geladen. Dabei obliegt es der Client-Applikation das Dokument allenfalls aus einem Offline Cache zu laden.

### **Serverbasiert**

Wird das Dokument Management System serverbasiert (auch der Surface selbst könnte als Server fungieren) eingebunden, muss ein mobiler Client lediglich die gewünschten Dokumente signalisieren.

Dies hat der Client vorher vielleicht über eine Bookmark-Funktion entsprechend eingerichtet. Der Surface macht nun den Dateizugriff direkt auf das Dokument Management System. Dies setzt voraus, dass das Dokument Management System erreichbar und verfügbar ist und der Surface sich dem Dokument Management System gegenüber auch authentifizieren kann. Diese Authentifizierung geschieht entweder mit einem generischen Login, was eventuell Sicherheitstechnisch bedenklich ist, oder sie geschieht mittels Credentials welche vom Client übermittelt werden (Identity Impersonation).

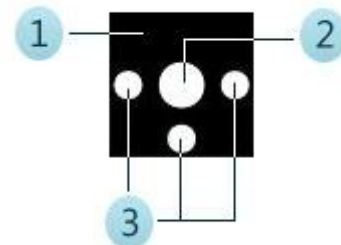
### Sicherheit auf dem iPad

Die Workpad Applikation auf dem iPad muss sicherstellen, dass der Zugang zum eingerichteten Sharepoint Konto gesichert erfolgt. Weiter müssen die Credentials des Benutzers sicher abgespeichert werden. Es muss weiter verhindert werden, dass Unbefugte Zugriff auf die Dateien auf dem Sharepoint oder die gespeicherten Credentials erhalten, falls eine unbefugte Person physischen Zugang zum iPad erlangt. Diese Sicherheitsanforderungen muss die Applikation Workpad erfüllen, die nicht Bestandteil des Projekts STEP ist.

### Sicherheit auf dem Surface

Folgende Sicherheitsanforderungen müssen durch STEP erfüllt sein:

- Das mobile Endgerät muss physisch anwesend sein um angemeldet zu werden.
- Eine zur Anmeldung initiierte Session hat nur eine kurze Gültigkeitsdauer. Sie wird ungültig, wenn nach einer kurzen Zeit keine Anmeldung erfolgt ist.
- Die Kommunikation muss verschlüsselt erfolgen, damit beim mitschneiden des Netzwerktraffics keine sensiblen Daten an Drittpersonen gelangen.
- Die eindeutige ID, mit der ein mobiles Endgerät identifiziert wird, muss sicher in der Workpad Applikation gespeichert werden, damit sie nicht ausgelesen werden kann.
- Ist ein mobiles Gerät am Surface angemeldet, muss die Session nach einer gewissen Zeit ablaufen, wenn der Benutzer untätig geblieben ist.
- Die Dokumente werden vom iPad an den Surface übertragen. Das iPad kommuniziert mit dem Dokumentmanagementsystem.
- Es werden keine Credentials des Dokumentmanagementsystems vom iPad an den Surface übertragen.



### Session Management und Device Tracking

Grundsätzlich soll es möglich sein, dass mehrere Benutzer gleichzeitig am Surface angemeldet sein können. Ebenfalls sollte es möglich sein, dass ein Benutzer mit mehreren mobilen Endgeräten gleichzeitig am Surface angemeldet ist. Das die Surface Applikation dabei erkennt, dass es sich um den gleichen Benutzer handelt, ist eine Option. Eine Session mit dem Surface beginnt mit der Anmeldung des mobilen Geräts am Surface. Nach einer definierbaren Zeit oder durch manuelle Interaktion soll diese Session beendet werden.

Konzeptionell unterstützt der Surface die Interaktion mit physischen Objekten, um diese mithilfe eines Tags eindeutig identifizieren zu können. Dies ermöglicht einen getaggten Gegenstand eindeutig einem Benutzer zuzuordnen zu können.

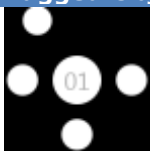
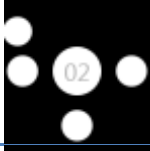
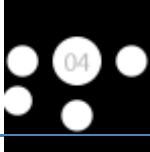
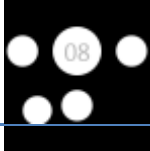
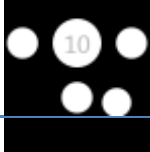
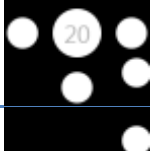
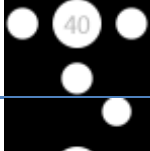
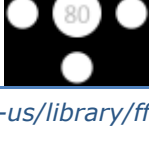
Diese sogenannten Tagged Objects entsprechen einer Fläche von ca. 19mmx19mm und können einen Wert zwischen 0 und 255 enthalten, sprich einen 8 Bit Wert.

Diese Tags bestehen aus folgenden Komponenten:

1. einem infrarotabsorbierenden Hintergrund (schwarze Fläche)

2. einem infrarotreflektierendem Kreis(Durchmesser 6.35 mm) im Zentrum
3. drei infrarotreflektierenden Kreisen (Durchmesser 4.064mm) um den zentralen Kreis, um die Positionen links rechts und unten zu definieren
4. bis zu acht Kreisen (Durchmesser 3.81mm), über welche die Werte ermittelt werden

Dabei entsprechen die vier optionalen Kreisen in der linken Hälfte den ersten vier Bits, die vier Kreisen in der rechten Hälfte den letzten vier Bits, wie in den folgenden Beispieltags auffällt:

Wert	Bit	Tagged Object
1	0000 0001	
2	0000 0010	
4	0000 0100	
8	0000 1000	
16	0001 0000	
32	0010 0000	
64	0100 0000	
128	1000 0000	

Quelle: <http://msdn.microsoft.com/en-us/library/ff727854.aspx>

Dieses Tracking bietet zwei Vorteile:

### Die Platzierung des persönlichen Benutzermenüs

Sie ist mittels eines getaggtten Gegenstandes einfacher, schneller und präziser als die klassische Variante via Touch-Interface.

Denn ohne dieses Tracking Verfahren, würde bei der Anmeldung eines Benutzers am Surface sein Menü wohl irgendwo auf dem Tisch, platziert werden, da der Surface nicht weiss, wo am Tischrand sich der Benutzer aufhält. Eine gute Wahl könnte hierbei die Mitte des Surface sein, da sie von allen Positionen, an der sich der Benutzer aufhalten könnte gleich gut erreicht werden kann. Nach dem Erscheinen seines persönlichen Menüs, wird der Benutzer es umplazieren wollen, damit es sich in

seiner Nähe befindet. Um dies zu erreichen müsste er das Menü mit einem Finger via Touch-Interface zu sich ziehen.

Mit einem getaggten Gegenstand auf dem Surface der dem Benutzer zugeordnet werden kann, ist dies viel komfortabler, da das Menü gleich zu Beginn an der gewünschten Position dargestellt werden kann. Ein Verschieben oder Drehen des Menüs ist intuitiv erreichbar, indem der Gegenstand umplaziert oder gedreht wird.

### **Die Abmeldung vom Surface**

Sobald der dem Benutzer zugeordnete Gegenstand vom Surface entfernt wird, wird der Benutzer abgemeldet und seine Session damit beendet. Die Alternative dazu ist, dass der Benutzer sich manuell über sein persönliches Menü abmelden muss.

### **Ursprünglicher Lösungsansatz**

Ursprünglich war als Device-Tracking Lösung die direkte Interaktion mit dem mobilen Endgerät angedacht. Damit ist gemeint, dass das mobile Gerät seinen eindeutigen Schlüssel in einem Tag darstellt und das Gerät dann mit der Bildschirmseite nach unten auf den Surface gelegt wird. Abklärungen und erste Versuche haben aber gezeigt, dass dies mit Surface in der aktuellen Version nicht möglich ist. Wird der Tag auf dem Display eines mobilen Geräts dargestellt und dieses auf den Tisch gelegt, konnten in Tests keine brauchbaren Resultate erzielt werden. Für diesen Test wurde eine Surface Applikation aus den SDK Samples ([RawImageVisualizer](#)) verwendet, die den Input den der Tisch erhält, also sozusagen seine *Sicht*, gespiegelt anzeigt. So ist es möglich, zu visualisieren, was der Surface erkennen kann.



*Für die Tests wurde ein Tag auf einem mobilen Gerät angezeigt.*



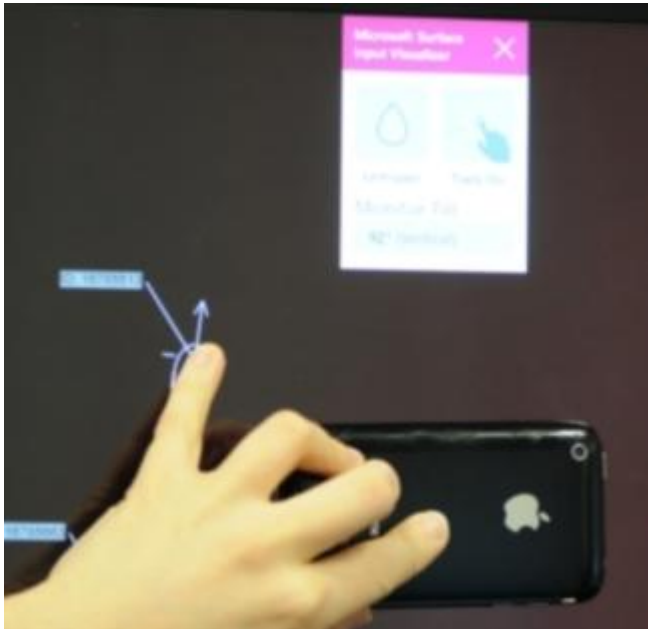
*Das Display des mobilen Geräts wird vom Surface nur als einfarbig-graue Fläche erkannt.*

Eine kleine Verbesserung konnte erreicht werden, indem die Helligkeit des Bildschirms auf dem mobilen Gerät auf 100% erhöht wurde. Für das menschliche Auge waren so die Konturen des Tags schwach erkennbar.



*Die Display-Helligkeit des mobilen Geräts wurde auf 100% erhöht.*

Die Qualität war aber eindeutig unzureichend, um eine Erkennung mit Surface zu ermöglichen. Um dies zu testen, wurde die auf Surface bereits vorinstallierte Software *Input Visualizer* verwendet. Der *Input Visualizer* zeigt an, was der Surface erkennt, bzw. es werden die Tags die gelesen werden konnten, dargestellt. Wird anderer Input erkannt als ein Tag, wird eine generierte ID angezeigt.



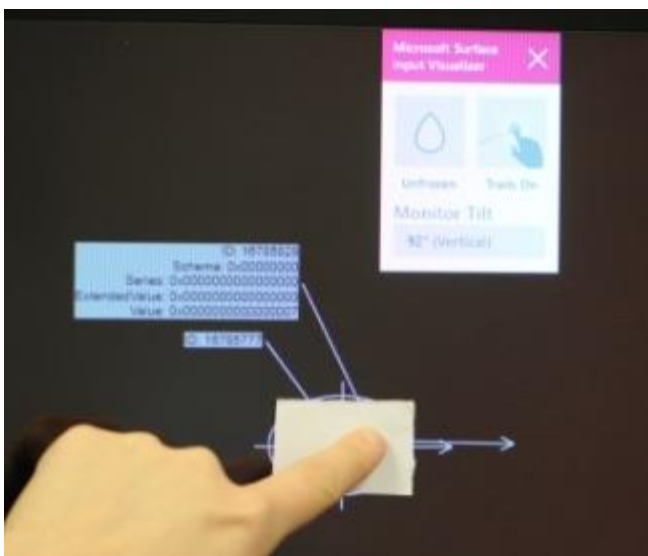
*Der Finger wird als Input erkannt und mit einer ID versehen. Das Display des mobilen Geräts wird nicht als Input erkannt.*

Sind die Tags auf Papier ausgedruckt, werden sie von Surface problemlos erkannt.





Hier zu sehen das Bild, wenn die Tags auf Papier auf den Surface gelegt werden. Schön zu sehen ist ebenfalls, dass nur an den Stellen wo mit der Hand das Papier satt auf den Tisch gedrückt wird das Bild scharf ist. An den übrigen Stellen ist es unscharf.



Wird ein einzelner Tag auf Papier auf den Surface gelgt, kann man im Input Visualizer gut zeigen, dass der Tag problemlos erkannt wird. Auf diesem Bild wurde der Tag mit dem Wert 7 aufgelegt.

### Alternative

Als Alternative bietet sich die Möglichkeit an, anstatt eines mobilen Endgerätes ein Gegenstand zu verwenden, der mit einem Papier-Tag versehen wurde. Damit könnte das Tracking realisiert werden. Unsere Tests mit Papier-Tags haben gezeigt, dass diese vom Surface problemlos erkannt werden. Einem Surface-Benutzer kann also ein Tag zugewiesen werden, der dann an einem Gegenstand (z.B. Spielfigur) auf der Unterseite angebracht wird. Die Einschränkung ist dabei, dass der Surface maximal 256 Tags eindeutig unterscheiden kann und somit maximal 256 verschiedene Personen mit einem Tracking-Gegenstand arbeiten können.

Quelle: Surface 2.0 SDK - Tagged Objects and Tag Visualizations  
<http://msdn.microsoft.com/en-us/library/ff727801.aspx>

Zusammenfassend kann gesagt werden, dass der Benutzer seine Session direkt mit UI-Elementen auf dem Surface steuern können soll. Um dies zu Vereinfachen besteht die Möglichkeit, dass zusätzlich ein Device-Tracking realisiert wird.

## **Dokument Handling auf Surface**

Primär in dieser Anwendung ist die Darstellung einer möglichst grossen Anzahl verschiedener Dateiformate auf dem Surface. Typischerweise sind das Rastergrafiken, Microsoft Office und PDF Dokumente.

Da die Surface 2.0 Shell auf Windows 7 basiert, können auf einfache Weise die, für ein bestimmtes Dateiformat geeignetsten, Anwendungen installiert werden. Das eigentliche Programm kann entweder direkt gestartet oder via Active X/COM und WindowsFormsHost in WPF integriert werden.

Diese Lösung hat den Nachteil, dass die Surface Anwendung von kommerziellen Drittanwendungen und deren Lizenzmodell abhängig ist. Ein weiterer Nachteil ist auch, dass die User-Interfaces dieser Programme nicht für die Touch-Bedienung geeignet sind. Es müssten also für alle unterstützten Dateiformate ein geeigneter Editor geschrieben werden, welcher die wichtigsten Funktionen auf Fingerbedienung optimiert implementiert.

Solche Implementationen würden den Rahmen dieses Projektes sprengen, weshalb die Interaktion mit Dokumenten auf *Read-Only* eingeschränkt wird.

Da eine Editierfunktion aber sehr vorteilhaft wäre, soll es in Zukunft möglichst einfach dahin gehend erweitert werden können. Es soll deshalb so realisiert werden, dass eine Rückspeicherung grundsätzlich möglich sein muss.

Anstatt einer Editierfunktion, könnte innerhalb dieses Projekts eine Integration des Sharepoint Workflows realisiert werden. Diese Integration würde eine Rückspeicherung ebenfalls benötigen.

Die maximale Anzahl Personen, die gleichzeitig ein Dokument auf dem Surface betrachten können liegt bei acht Personen. Diese Einschränkung wird durch die Grösse des Surface vorgegeben. Möchten mehr als acht Personen ein Dokument auf dem Surface betrachten, ist das Sichtfeld auf den Bildschirm von einer oder mehreren Personen eingeschränkt.

## **Spiegelmodus**

Der Spiegelmodus spricht das Problem an, dass sobald mehrer Personen um den Surface Tisch verteilt stehen, nicht alle optimal die dargestellten Dokumente betrachten können. Daher wäre ein Lösungsansatz, dass der Surface Vertikal oder Horizontal gespiegelt werden könnte, so können die Teilnehmer auf beiden Seiten das Dokument sehen. Sobald man jedoch von allen vier Seiten den idealen Einblick auf das Dokument haben will, kann man dies nicht mehr über spiegeln lösen.

Die Idee vom Spiegelmodus wurde verworfen, da sie nur optimal funktionieren würde, wenn genau zwei Personen anwesend sind, bei mehrere würde der Spiegelmodus wenig Benefit generieren, wenn nicht gar hinderlich wäre.

## **Dokumente kopieren**

Anstatt die Dokumente zu spiegeln, könnte man die Dokumente kopieren, so dass mehrer Kopien von ein und demselben Dokument dargestellt werden. Diese Kopien könnte man auch an ein Parent Dokument binden, sobald auf dem Parent gescrollt wird, wird der Scroll Befehl an seine Children weitergegeben.

Der DocumentViewer wurde so implementiert, dass der erste Viewer, der das Dokument darstellt, der Host des Dokuments ist. Vom Host kann man Kopien erstellen, welches von den jeweiligen Meetingteilnehmer nach belieben positioniert werden können.

Der Dokument Host kann seine Kopien an sich linken, so dass die Kopien jeweils synchron zum Host durch das Dokument bewegen z.B. scrollen.

#### 4.4 EVALUATION DES MOBILEN ENDGERÄTS

Der Business Partner ti&m AG wünscht nach Möglichkeit eine Erstimplementation des Clients auf dem iPad. STEP soll zukünftig aber auch weitere Plattformen wie Android oder Windows 8 unterstützen können.

##### Workpad

In einem unabhängigen Projekt der ti&m AG, wurde mit Hilfe des internen iPad Application Frameworks die Workpad-Lösung entwickelt. Sie gibt dem Benutzer eine backend-gestützte Lösung für die Dokumentverwaltung auf dem mobilen Gerät. Diese Lösung bietet für STEP eine ideale Plattform für die Clientsoftware. Die Lösung wird daher im Rahmen dieses Projektes um eine Backend-Anbindung an SharePoint Server 2010 sowie die benötigten Surface Interaktionen erweitert.



#### 4.5 VOR- UND NACHTEILE DIESES PROJEKTES

##### Vorteile

- Collaboration
- Reduziert Papier-Maschine Übergänge
- Aktuelle Darstellung direkt vom Dokument, nicht von möglicherweise veralteten Snapshots
- Grösseres Display als mobiles Endgerät

##### Nachteile

- Kostspielige Anschaffung eines Surface Tisches
- Betrachtung ab höherer Teilnehmerzahl suboptimal
- Eingeschränkt auf diverse Dokumenttypen

#### 4.6 SCOPE DIESES PROJEKTS

Wie in der Analyse begründet umfasst der Scope dieses Projektes folgende Funktionen, Work-Items, etc.:

##### Verbindungsaufbau mit mobilem Endgerät Deliverable #278

- Null, ein oder mehrere mobile Endgeräte können am Surface angemeldet werden.

- Auf dem Surface wird ein Menü pro angemeldetem Gerät angezeigt.
- Ein Gerät kann über den in [Verbindungsaufbau mit mobilem Endgerät](#) beschriebenen Vorgang angemeldet werden.

### **Kommunikation mit dem mobilen Endgerät Deliverable #279**

- Die Kommunikation muss verschlüsselt stattfinden.
- Die Verbindung muss in der Lage sein, ein Dokument bis 2MB innerhalb von einer Sekunde zu übertragen.
- Das mobile Endgerät soll via IP-Protokoll mit dem Surface kommunizieren.
- Es wird vorausgesetzt, dass sich die Surface-Applikation und das mobile Endgerät via IP erreichen können.

### **Kommunikation mit Dokument Management System Deliverable #280**

- Es ist eine Voraussetzung, dass das mobile Gerät die Credentials für das Dokument Management System hinterlegt hat, oder diese Möglichkeit anbietet.
- Die Credentials werden über eine sichere Verbindung an den Surface übertragen.
- Die Surface Applikation kann sich zum Dokument Management System mit den gelieferten Credentials verbinden.

### **Session Management und Device Tracking Deliverable #281**

- Ein angemeldetes Gerät muss nach einer definierten Zeit automatisch abgemeldet werden. (Session-Timeout)
- Es muss ein Mechanismus realisiert werden, der den vorzeitigen Sessionabbruch verhindert.
- Die Session für ein Gerät kann vom Benutzer manuell terminiert werden.
- Tracking über einen getaggten Gegenstand ist optional und nicht im Scope.

### **Dokument Handling auf Surface Deliverable #282**

- Im Menü pro Gerät werden Dokumente, die zur Verwendung mit dem Surface markiert wurden, aufgelistet.
- Ein Benutzer kann ein aufgelistetes Dokument öffnen und Surface zeigt dieses an.
- Der Benutzer kann auf dem mobilen Endgerät ein Dokument direkt auf den Surface übertragen.
- Dokumenttypen die unterstützt werden sollen: Word (docx), Excel (xlsx), Power-Point (pptx) oder PDF (pdf)
- Eine Unterstützung von Bild-Formaten wie zum Beispiel JPEG, Bitmap oder PNG ist optional.
- Die Dokumente werden nur als Read-Only geöffnet. Eine Editier-Funktion wird nicht angeboten.
- Die maximale Anzahl Personen, die STEP gleichzeitig verwenden wird durch die Grösse des Surface auf acht Personen limitiert.

## 5 SZENARIO UND PAPIER PROTOTYP

In diesem Dokument wird das Soll-Szenario, sowie der Papier-Prototyp dokumentiert, wie STEP später verwendet werden kann.

### 5.1 SZENARIO

Greg ist Mitarbeiter der ti&m AG und ist zuständig für die Betreuung von Kunden. Greg hat einen Arbeitsrechner, auf welchem er Office- oder PDF- Dokumente erstellt, bearbeitet oder betrachtet. Die Dokumente werden bei ti&m AG mit einem Dokument Management System verwaltet. Als Lösung wird dafür Microsoft Sharepoint eingesetzt. Greg hat also einen Sharepoint Account, über den er Zugriff auf die von ihm benötigten Dokumente erhält. Um Dokumente unterwegs oder bei Kunden vor Ort zu betrachten setzt er ein Apple iPad ein. Auf dem iPad ist die Applikation Workpad installiert, welche ihm den Zugriff auf seine Sharepoint Dokumente ermöglicht. Da Greg häufig mit der Workpad Applikation arbeitet, hat er die Credentials seines Sharepoint Account in Workpad hinterlegt.

Der Kunde Jeff hat bei Greg eine Offerte für eine individuelle Softwarelösung angefordert. Greg hat nach umfangreichen Abklärungen die finale Version der Offerte erstellt. Greg lädt Jeff zur ti&m AG ein, um die Offerte zu präsentieren. Die Offerte ist auf dem Sharepoint gespeichert.

Am Tag des Meetings trifft Jeff bei ti&m AG ein und wird von Greg in einen Sitzungsraum geführt, wo ein Microsoft Surface installiert ist. Der Surface ist ans interne Firmennetzwerk von ti&m AG angeschlossen und die Applikation STEP wurde bereits gestartet. Nach der Begrüssung tippt Greg auf dem Surface das Symbol an, um ein Gerät anzumelden. Der Surface zeigt nun einen von STEP generierten QR-Code an. Auf seinem iPad startet Greg jetzt die Workpad Applikation. Dort wählt er im Menü aus, dass er sich an einen Surface anmelden möchte. Die Kamera des iPad wird aktiviert und das Kamerabild auf dem Display dargestellt. Greg bewegt das iPad über den Surface, damit der QR-Code für die Kamera des iPad sichtbar wird. Konnte der Code gelesen werden, wird auf dem iPad wieder die Workpad Oberfläche dargestellt. Das iPad ist nun am Surface angemeldet. Der Surface zeigt ein Menü an, wo die in der Workpad-Applikation für Greg verfügbaren Dokumente aufgelistet werden. Greg "zieht" dieses Menü zu sich, um sich einen Überblick zu verschaffen. Er wählt das Offerten-Dokument aus, das er Jeff präsentieren möchte.

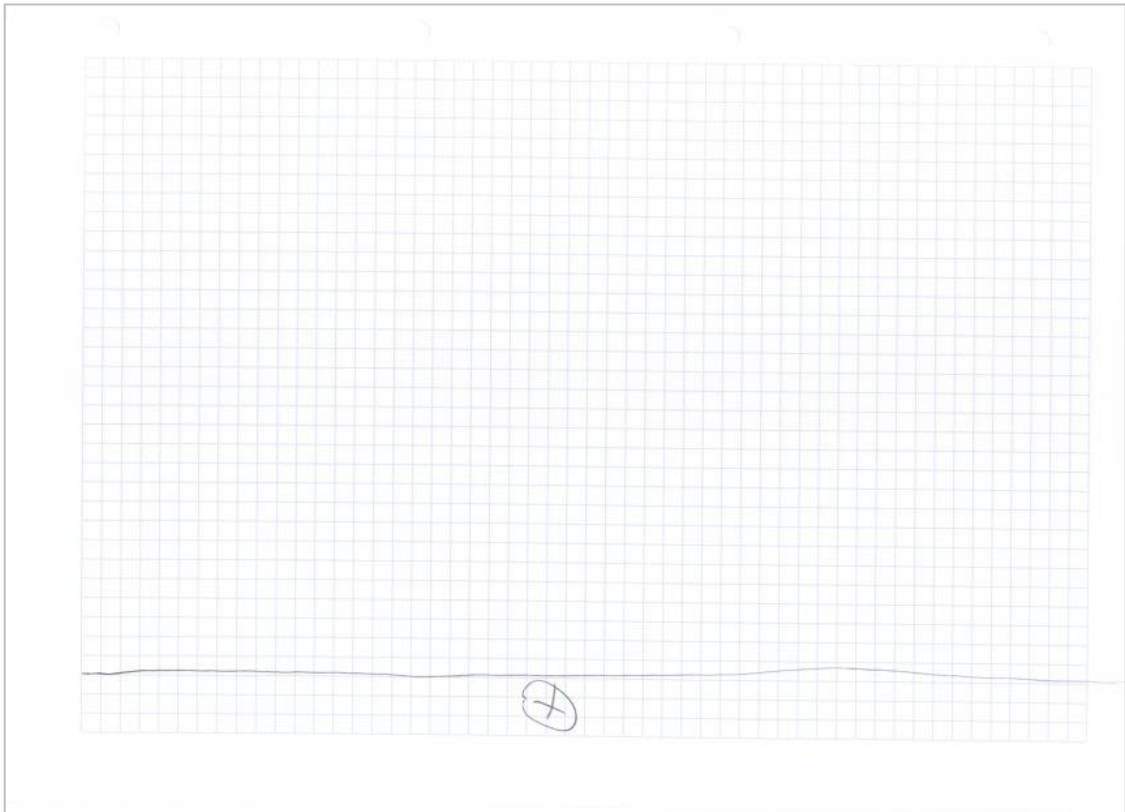
Das Dokument wird geöffnet und auf dem Surface dargestellt. Greg verschiebt, dreht und zoomt das Dokument, bis es für beide gut einsehbar positioniert ist. Sie diskutieren den Inhalt der Offerte und Greg scrollt bei Bedarf durch das Dokument. (Der Kunde ist nicht ganz einverstanden und scrollt wieder nach oben um einen Punkt nochmals zu lesen)

Haben sie die Offerte vollständig besprochen, verabschieden sie sich und Jeff verlässt den Raum. Greg meldet sein iPad vom Surface ab, indem er im Menü wo die Offerte aufgelistet ist, den Menüpunkt "Abmelden" auswählt. Er verlässt jetzt ebenfalls den Raum.

### 5.2 PAPIER-PROTOTYP

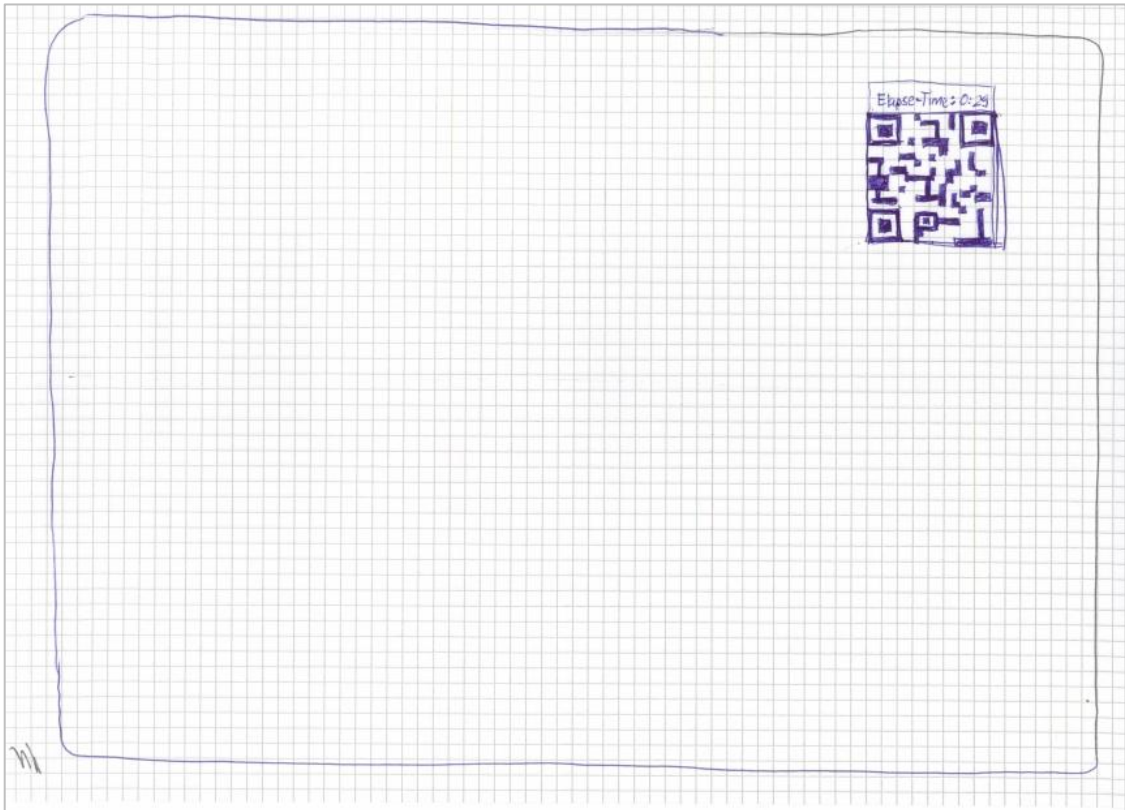
Der hier gezeigte Papier-Prototyp wurde für STEP erarbeitet und wurde ebenfalls im Modul [User Interfaces 2](#) verwendet. Um das erarbeitete Design auf gute Bedienbarkeit zu überprüfen, wurde zum Szenario ein Papier-Prototyp erstellt. Dieser Prototyp wurde anschliessend von einem zukünftigen Benutzer, mit dem Szenario als Ausgangslage, durchgespielt.

## Surface - Charmbar



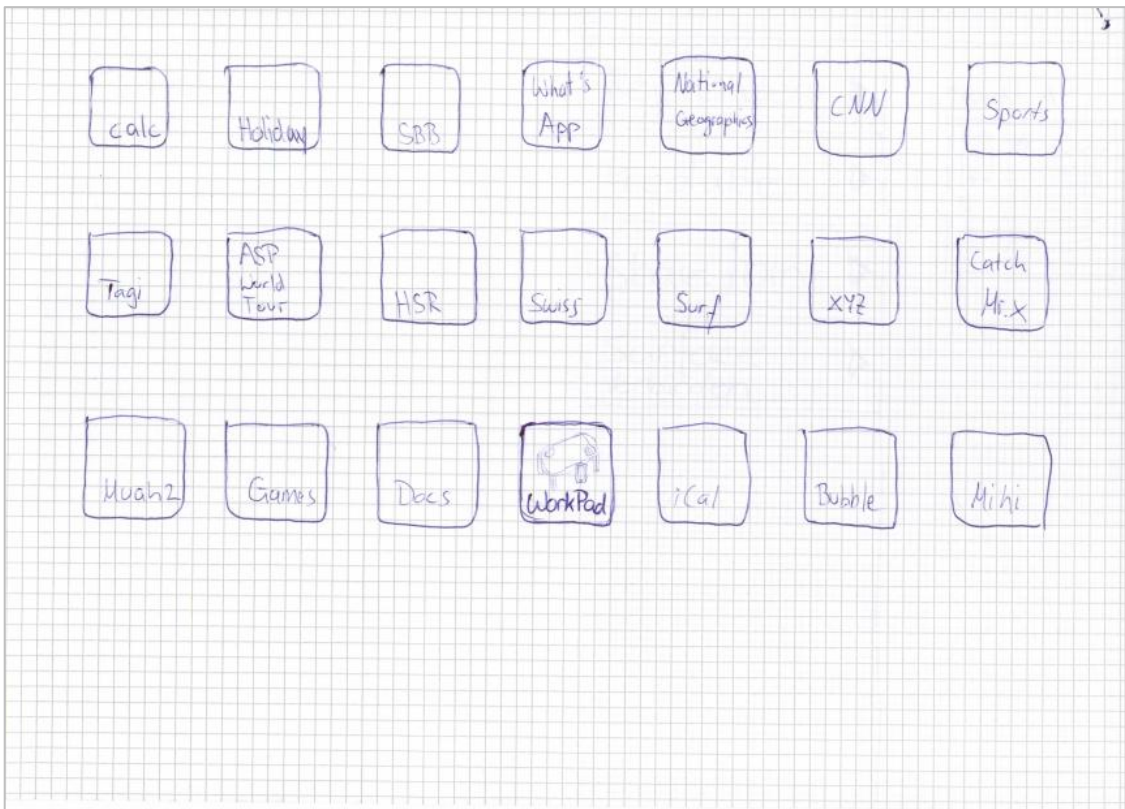
Auf der Charmbar, die auf dem Surface angezeigt wird, soll der Benutzer den Button (+) betätigen, um den Anmeldeprozess zu initiieren. Hierbei spielt es keine Rolle auf welcher Seite des Surface der Benutzer steht. Denn von jeder der vier Seiten kann eine solche Charmbar eingeblendet werden.

## Surface - QR-Code



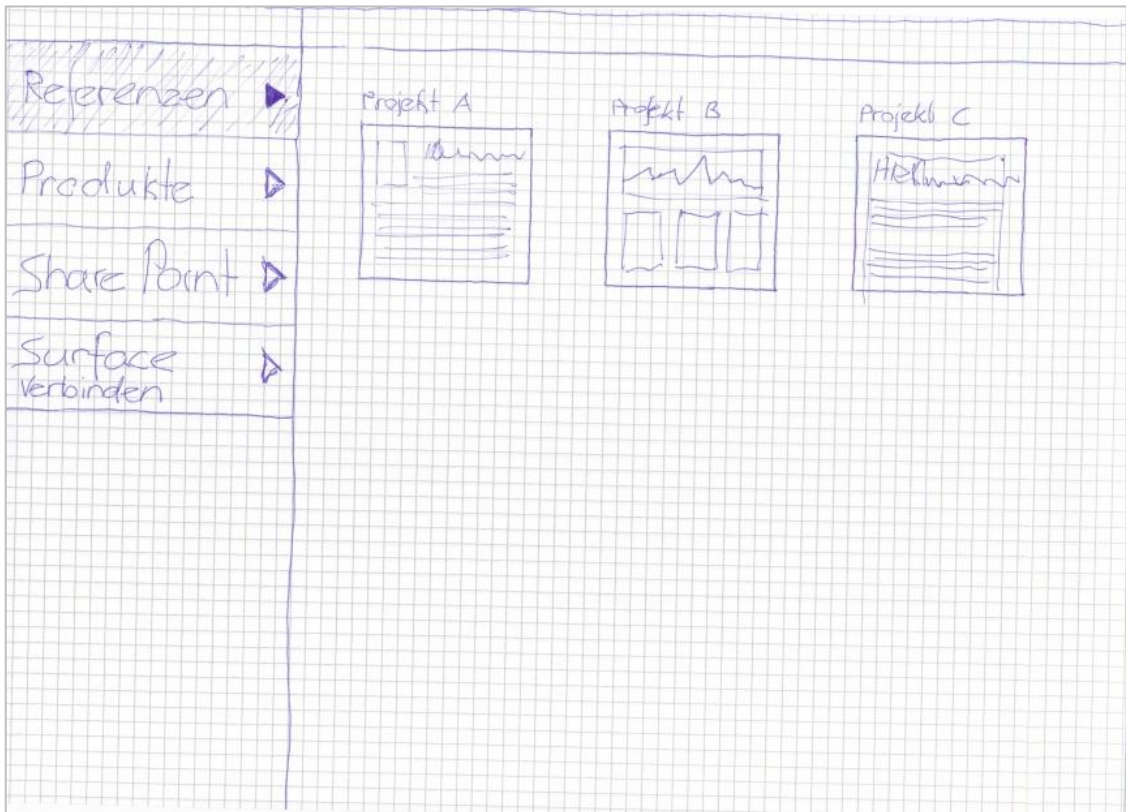
Der QR-Code (mit Gültigkeits-timer) wird auf dem Surface angezeigt, damit er dann vom iPad gelesen werden kann.

## iPad - Home Screen



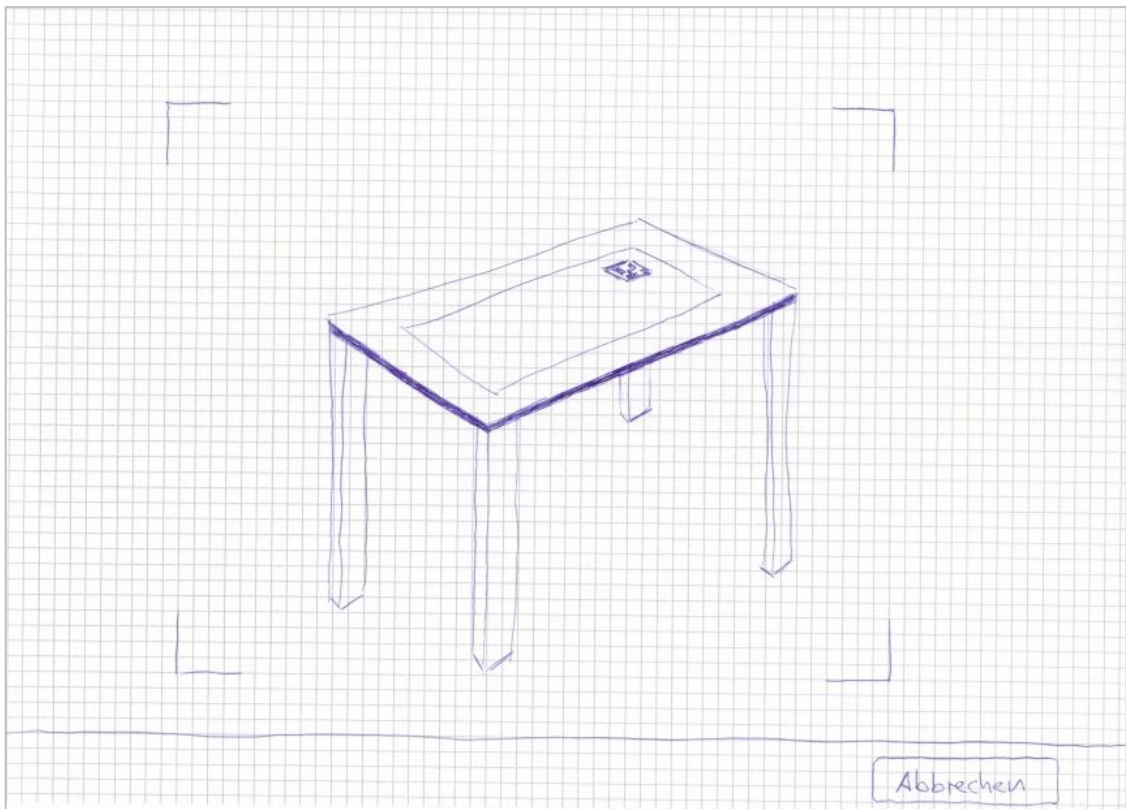
Der Benutzer nimmt nun das iPad zur Hand. Auf dem iPad wird jetzt der Home Screen mit den installierten Apps angezeigt. Der Benutzer startet die Workpad App.

### iPad - Workpad App gestartet



Auf dem iPad wird die zuvor gestartete Workpad App angezeigt. Der Benutzer wählt die Schaltfläche "Surface Verbinden" um den Anmeldeprozess seitens des iPad einzuleiten.

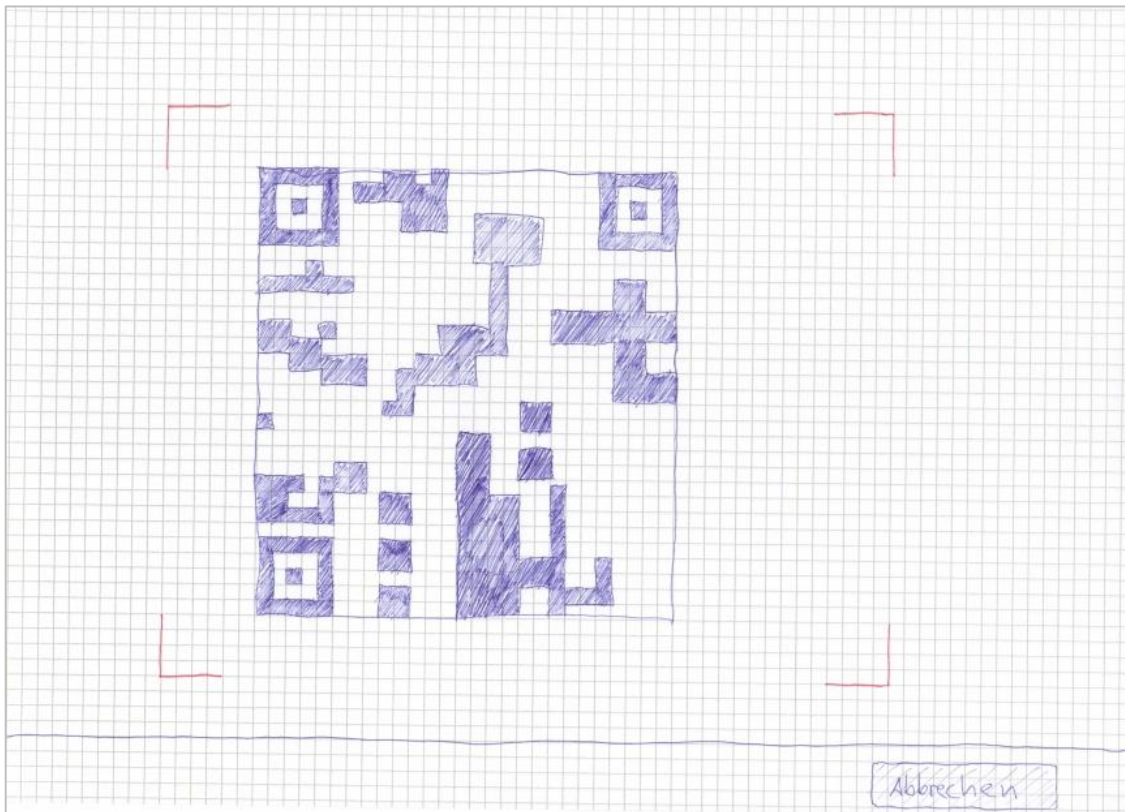
### iPad - Kamera ist aktiviert um den QR-Code zu lesen



Die Kamera des iPad wurde aktiviert und das Kamerabild wird nun anstatt des Workpad Menüs angezeigt. Der Benutzer hält nun das iPad über den auf dem Surface angezeigten QR-Code.

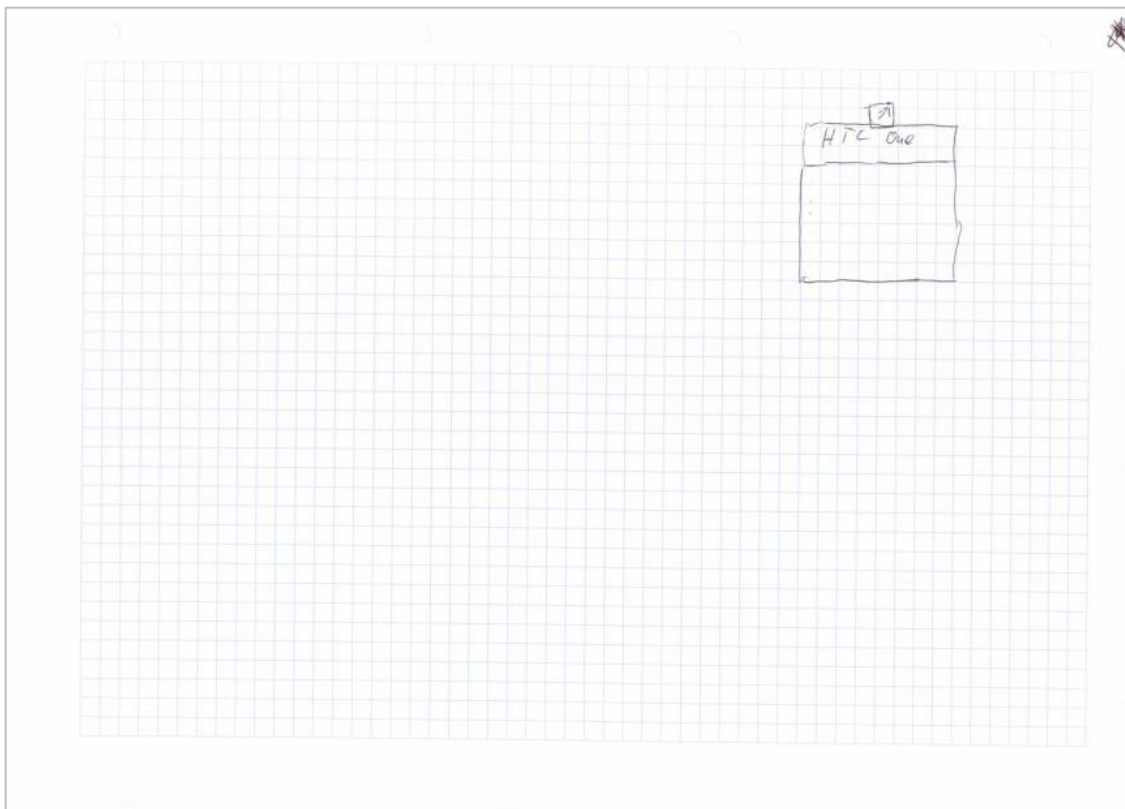


## iPad - QR-Code wurde erkannt



Sobald der QR-Code von der Workpad App erkannt wurde, meldet sich das iPad beim Surface an.

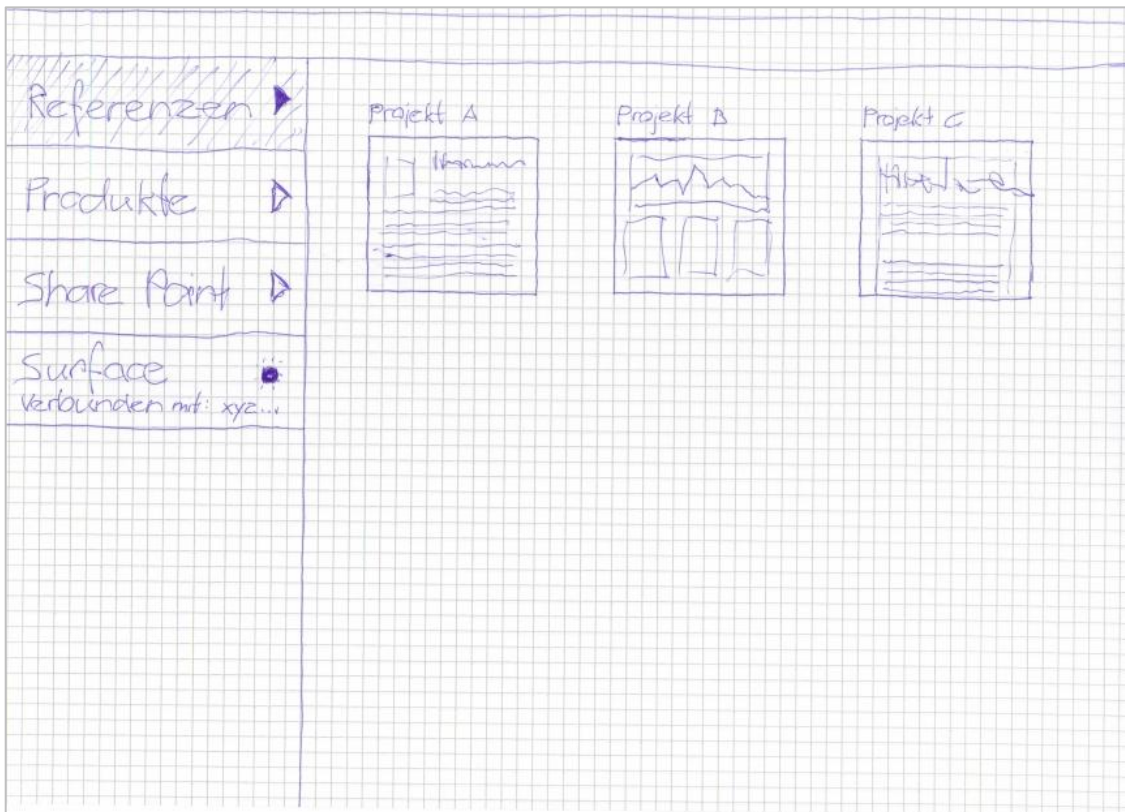
## Surface - Gerätemenü



Da das iPad nun mit dem Surface verbunden ist, wird auf dem Surface nun ein Menü für das verbundene iPad angezeigt. Der Benutzer hat nun die Möglichkeit, das Dokument, das er öffnen

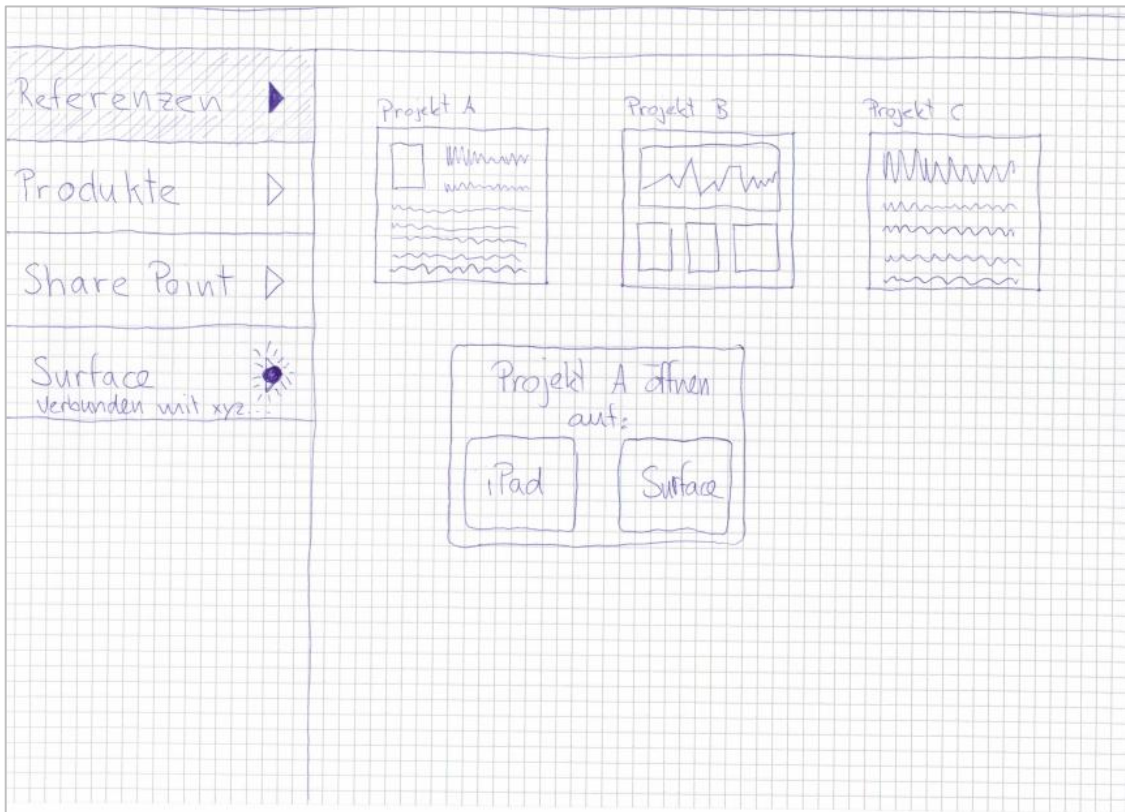
möchte aus diesem Menü auszuwählen oder aber er öffnet das Dokument auf dem iPad. In diesem Papier-Prototyp stellen wir letztere Möglichkeit vor.

### iPad - Am Surface angemeldet



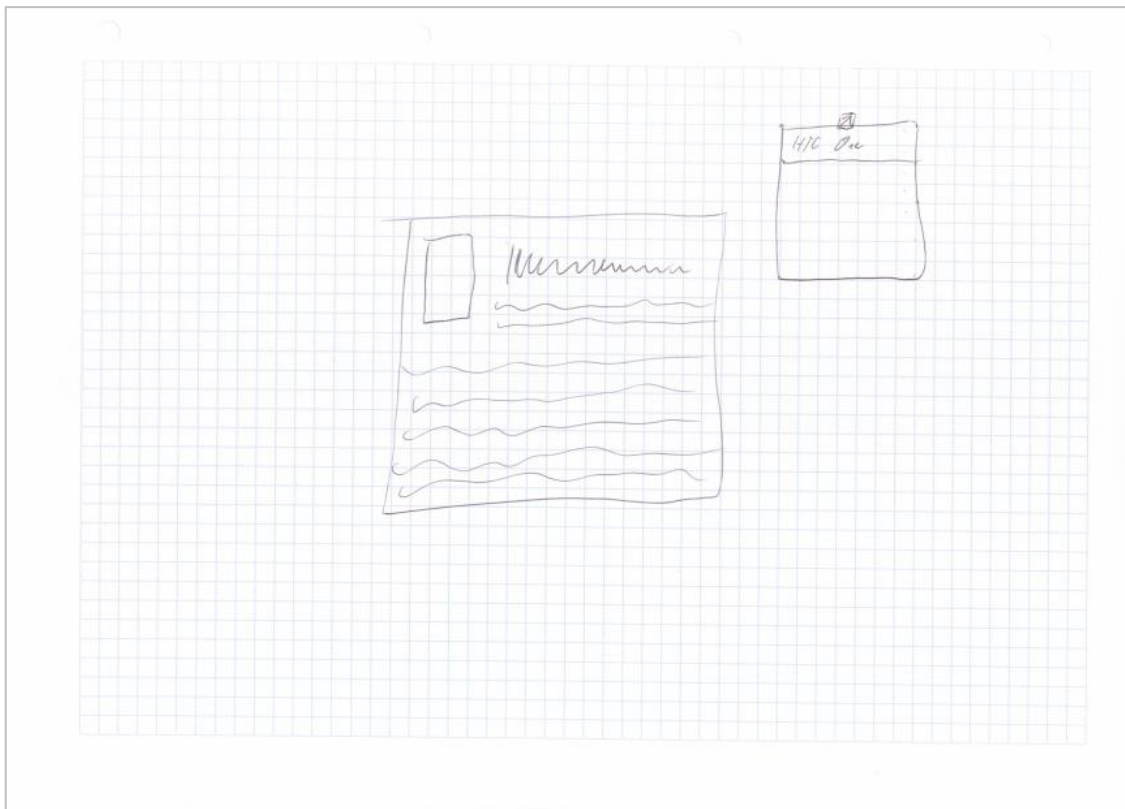
Dass das iPad jetzt mit dem Surface verbunden ist, kann am Menü erkannt werden. Dort heisst der letzte Menüpunkt jetzt "Surface - Verbunden mit xyz". Wobei xyz auf dem Papier ein Platzhalter für den Namen des Surface ist. Der Benutzer wählt jetzt das Dokument aus, das er auf dem Surface öffnen möchte.

## iPad - Wo soll das Dokument geöffnet werden?



Dem Benutzer wird nun ein Dialog angezeigt, wo er entscheiden kann, wo er das Dokument öffnen möchte. Er wählt die Schaltfläche Surface.

## Surface - Dokument wird angezeigt



Das Dokument wird jetzt auf dem Surface dargestellt, wo der Benutzer es seinen Bedürfnissen entsprechend drehen, zoomen oder verschieben kann.

### 5.3 PAPIER-PROTOTYP TEST-RESULTATE

Dem Benutzer schien zunächst nicht ganz klar, dass man sich zuerst am Surface anmelden muss, bevor ein Dokument auf dem Surface dargestellt werden kann. Der Test-Benutzer wollte als erstes auf das anzuzeigende Dokument klicken, anstelle des Menüpunkts "Verbinden mit Surface".

#### Verbesserungsvorschläge seitens des Testbenutzers

Beim Öffnen des Dokumentes sollte immer die Auswahl zwischen lokal oder auf dem Surface anzeigen erscheinen. Wird Surface ausgewählt, sollte automatisch der Verbindungsprozess seitens des iPad gestartet werden, falls man noch nicht verbunden ist. Der gestartete Verbindungsaufbau sollte für den Benutzer weitere Instruktionen enthalten, sodass er weiss, dass er auf dem Surface den Anmeldeprozess ebenfalls initiieren muss.

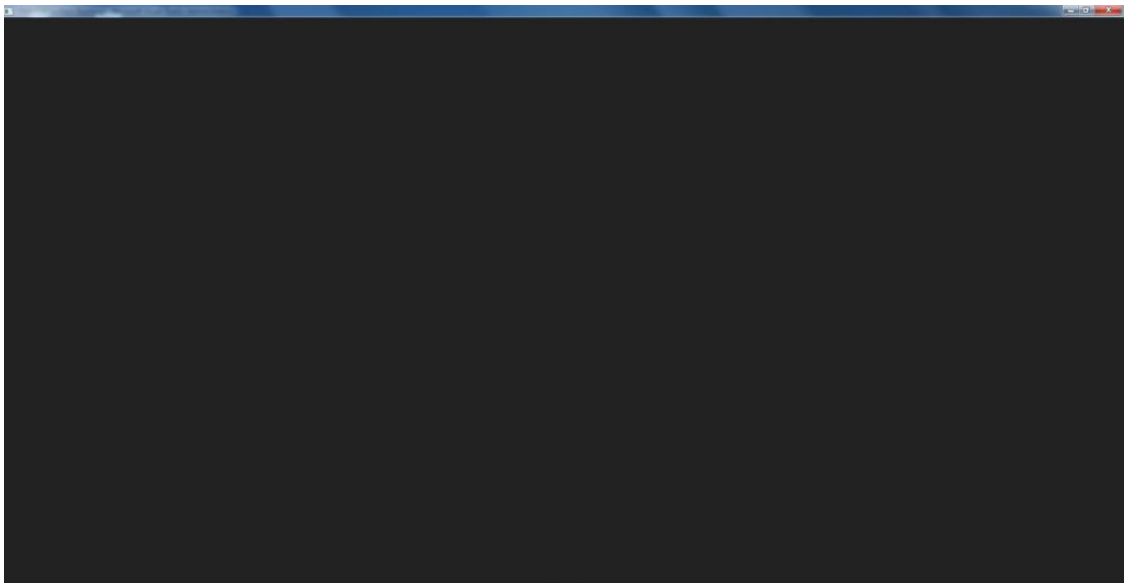
Eine Alternative wäre, die Schaltfläche Surface im Dialog ausgegraut darzustellen, falls man noch nicht verbunden ist. Dies signalisiert dem Benutzer unmissverständlich, dass er noch mit keinem Surface verbunden ist.

### 5.4 REDESIGN-PAPIER-PROTOTYP

Aufgrund des Benutzer-Feedbacks und einigen Limitierungen bei der Umsetzung des iPad Clients Workpad wurde der Prototyp anschliessend beschrieben angepasst.

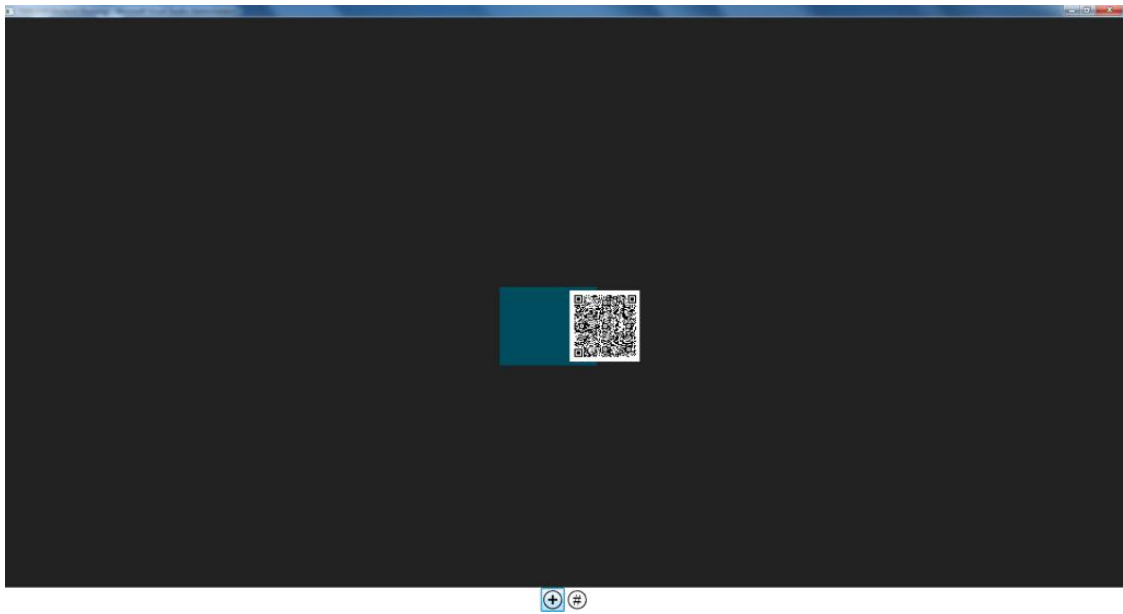
Da in der ersten Version lediglich das iPad als Mobilgerät unterstützt wird, wird an dieser Stelle das iPad stellvertretend für ein beliebiges mobiles Endgerät mit einer passenden Client-Software genannt.

#### Surface - initial



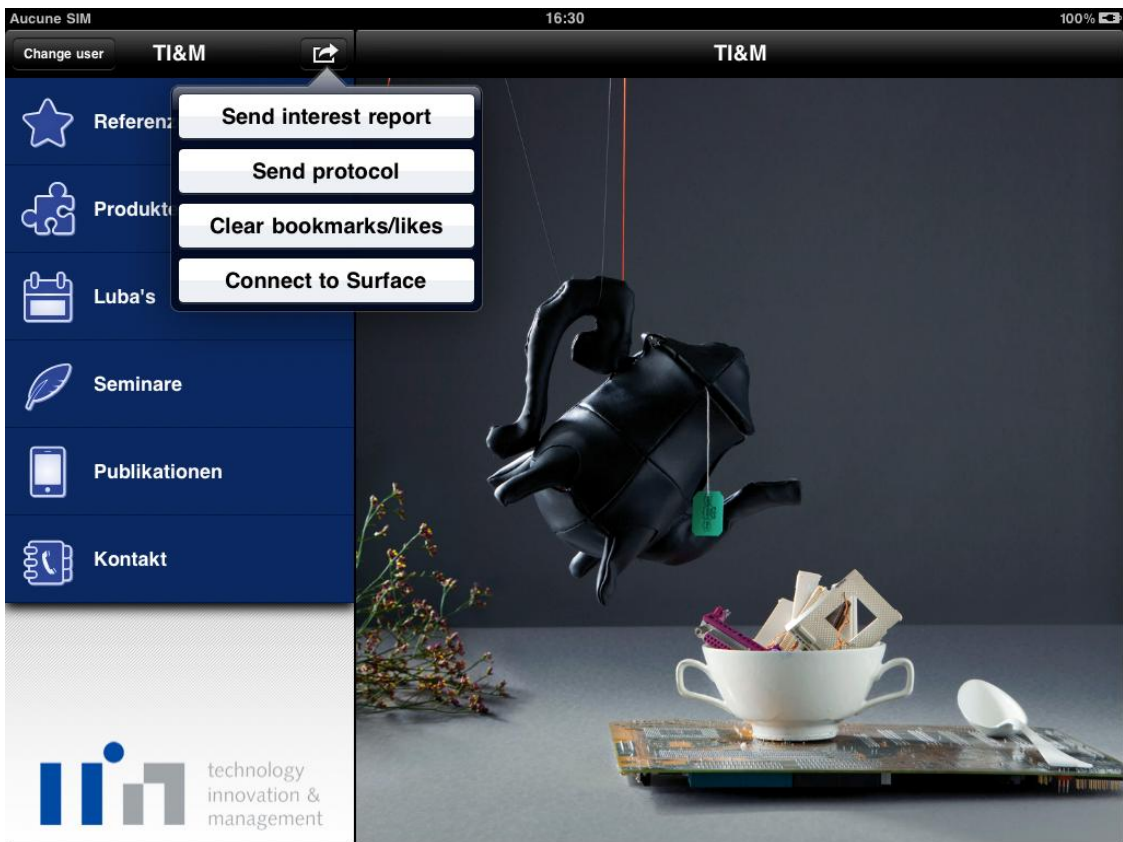
Surface - Um an einem vom Surface unterstützten Meeting teilnehmen zu können muss als erstes ein Benutzer oder Clientgerät hinzugefügt werden. Dieser Prozess wird auf der Charobar mithilfe vom (+)-Button gestartet. Hierbei spielt es keine Rolle auf welcher Seite des Surface man sich befindet, da auf jeder der vier Seiten eine solche Charmar existiert.

## Surface - QR-Code



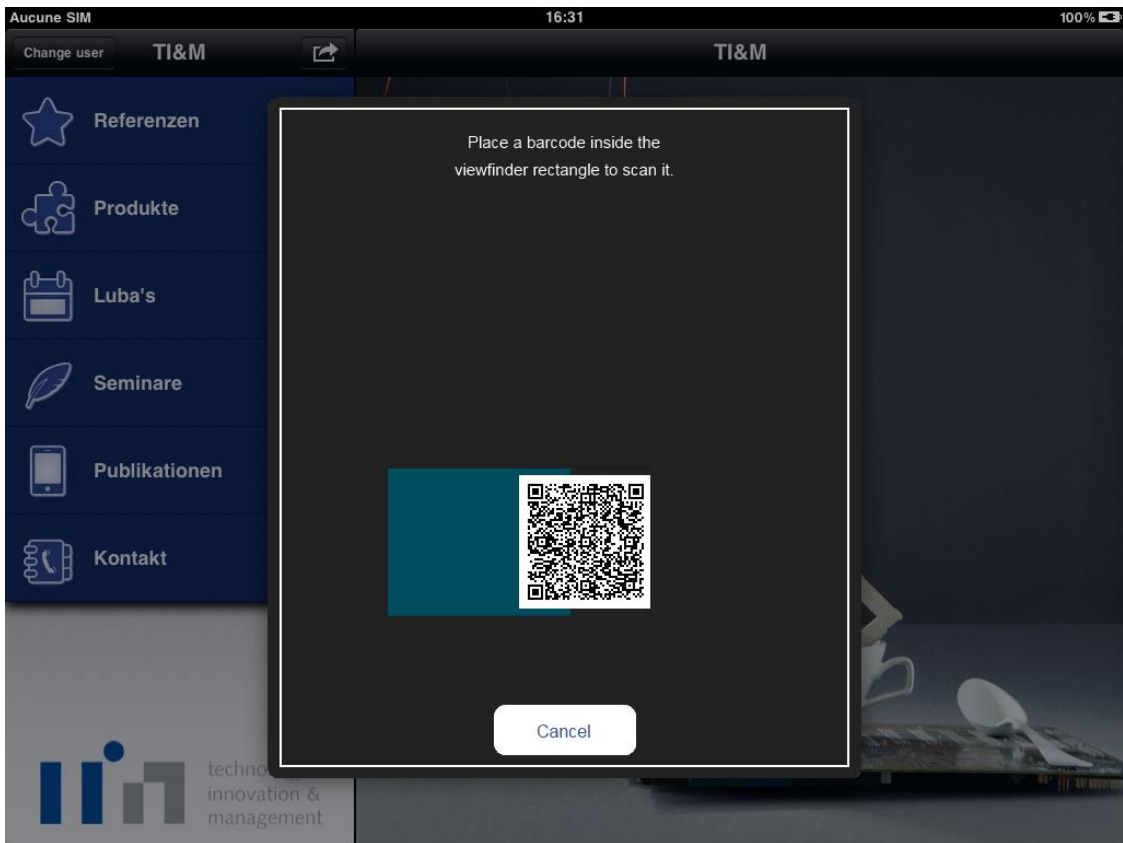
Surface - Der QR-Code (mit Gültigkeitstimer) wird auf dem Surface angezeigt. Dieser enthält die Identität des Surface, ein zeitlich begrenztes One Time Ticket und die Serveradresse auf die der Client verbinden muss.

## iPad Connect to Surface



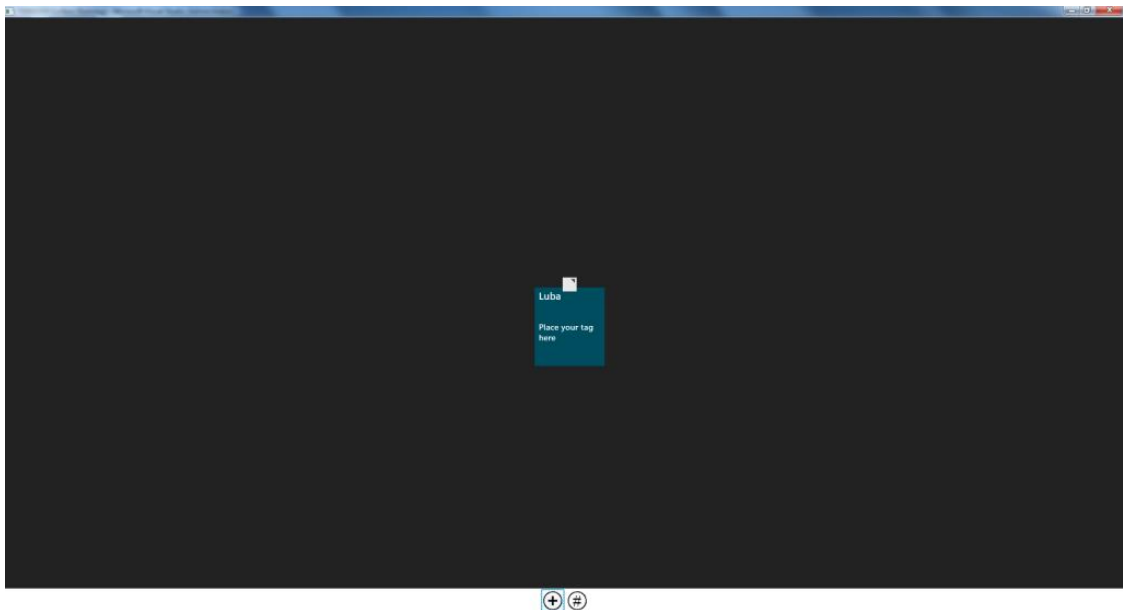
iPad - Auf dem iPad ist die Workpad Applikation gestartet welche als Client-Software dient. Über Workpad hat man nun die Möglichkeit eine Verbindung mit einem Surface aufzubauen.

## iPad Scan QR Code



iPad - Nach der Entscheidung sich mit dem Workpad zu verbinden, muss lediglich der QR Code innerhalb des Kamerasuchers platziert werden der somit automatisch gescannt wird. Das iPad verbindet sich dann selbständig unter Zuhilfenahme der im QR Code encodierten Informationen mit dem Surface.

## Surface User Logged In



Surface - Sobald ein mobiles Endgerät eine Verbindung aufgebaut hat, erscheint sein Device Menu auf dem Surface. Über dieses Menu hat der Nutzer die Möglichkeit alle Funktionen direkt am Surface

anzusteuern. Zusätzlich kann sich aber ein Nutzer auch dazu entscheiden die Bedienung über sein mobiles Endgerät (hier Workpad) vorzunehmen.

### iPad Dokument Handling

The screenshot shows an iPad application interface. At the top, it displays 'Aucune SIM', '16:31', and '100%'. The main header includes 'Change user', 'TI&M', and 'All categories'. A left sidebar menu contains icons and labels for 'Referenzen', 'Produkte', 'Luba's', 'Seminare', 'Publikationen', and 'Kontakt'. The main content area is titled 'Referenzen' and features three app categories: 'Börse Stuttgart', 'BCV Mobile Bank', and 'Coachdata'. Below these, there is a detailed section for the 'Börse Stuttgart mobile app' with a list of features and a 'Like' button.

**Börse Stuttgart mobile app**

- Online Börsenhandel von über 40 Wertpapieren.
- Echtzeit-Kurse für Finanzprodukte und Währungen.
- Tagesverlaufs-, Monats- und Jahresverläufe.
- Watchlisten mit frei konfigurierbaren Schwellwert-Meldungen (push notifications).
- Zugang zu existierenden webbasierten Daten.
- Börse Stuttgart TV.
- Support für 25 Smartphones (inklusive BlackBerry).
- <https://www.boerse-stuttgart.de/n>

**Börse Stuttgart wird mobil mit BSgroup Technology Innovation AG.**

iPad - Das iPad ist jetzt mit dem Surface Verbunden. Es kann normal mit dem Workpad client weitergearbeitet werden. Wenn der Nutzer ein bestimmtes Dokument auf seinem Workpad nun allen am Surface zur Verfügung stellen will, kann er das mittels dem "Broadcast-Button" am rechten Bildschirmrand bewerkstelligen.

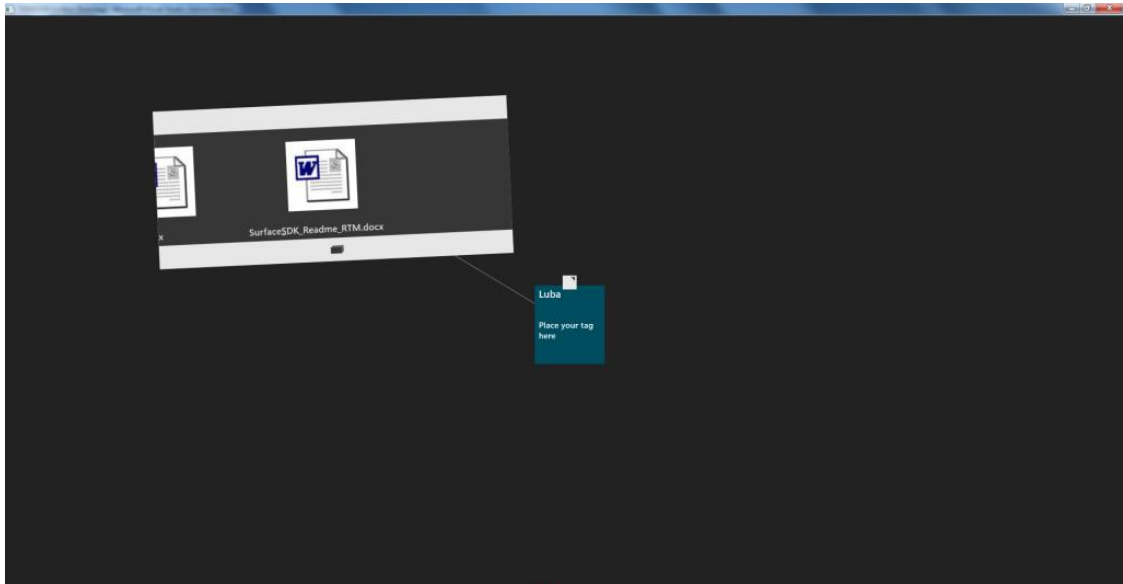
### Surface Dokument Handling I

The screenshot shows a Surface application interface with a grid of document thumbnails. A 'Luba' tag is being placed on one of the thumbnails, with a 'Place your tag here' prompt.

Surface - Wenn der Nutzer die Anzeige der Dokumente über den Surface steuern will, kann er das über sein Device Menu bewerkstelligen in dem er über das Quadrat am ober Rand mittels einer Swipe-Geste eine entsprechende Kategorie auswählt.

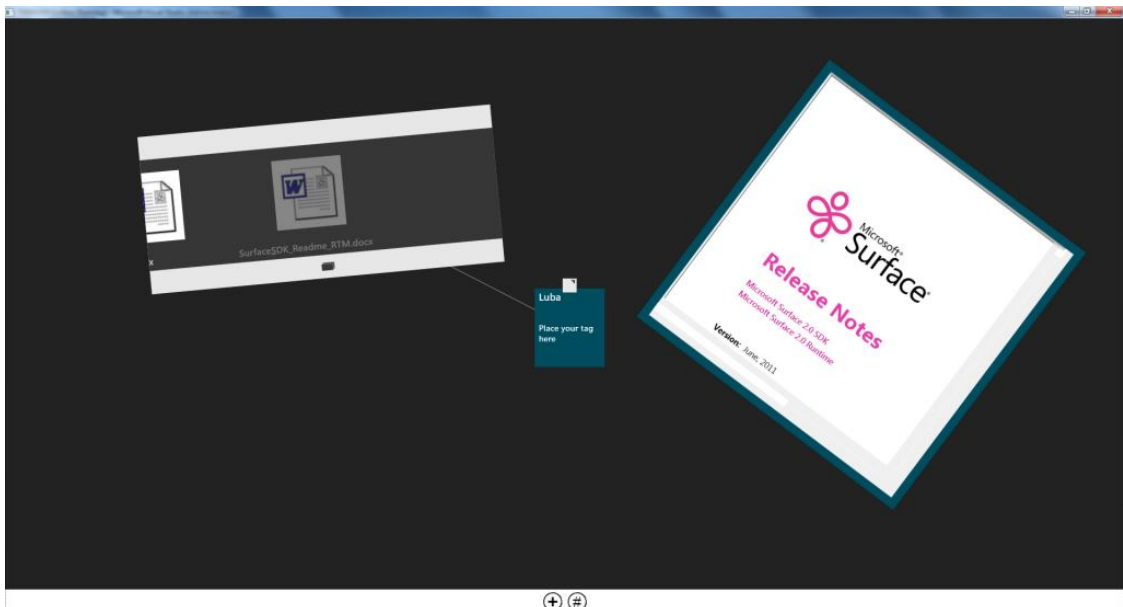
Im obenstehend Bild ist zu erkennen, dass das Device-Menü benutzerkontextspezifisch die selbe Struktur wie das verbundene Workpad reflektiert.

### Surface Dokument Handling II



Surface - Innerhalb einer geöffneten Kategorie werden dann die entsprechenden Dokumente aufgelistet. Der Nutzer öffnet das gewünschte Dokument indem er es per Drag and Drop auf den Surface zieht.

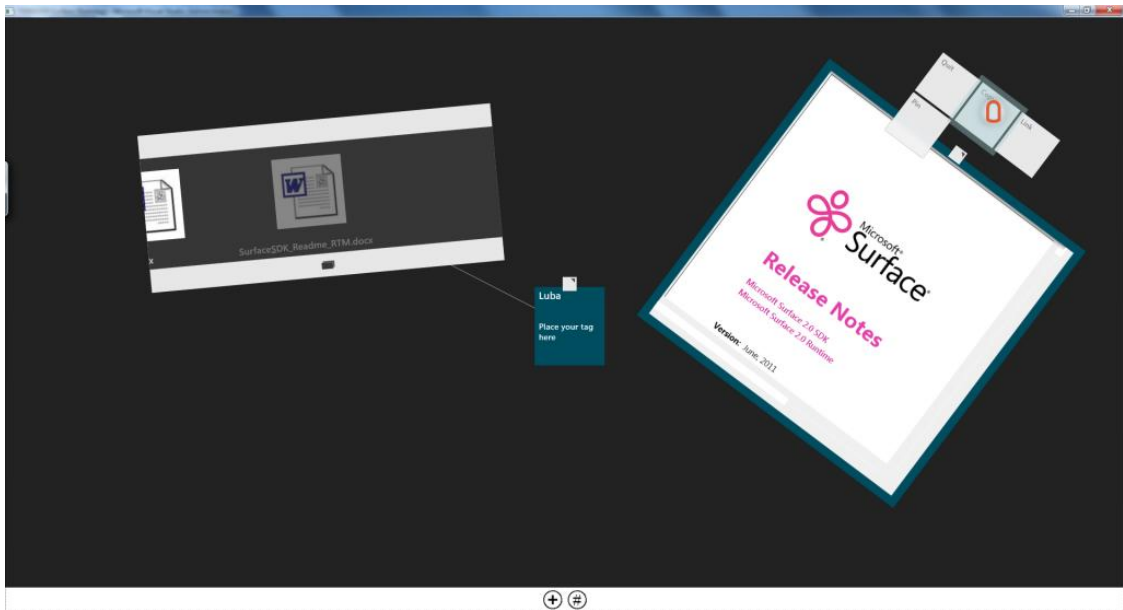
### Surface Document Viewing/Presentation I



Surface - Sobald aus einer Kategorie wie im letzten Schritt beschrieben das Dokument geöffnet ist, kann dieses auf dem Surface entsprechend gedreht und gezoomt werden. Der Dokumentrahmen hat dabei die selbe Farbe wie das zugehörige Device Menu.



## Surface Document Viewing/Presentation II



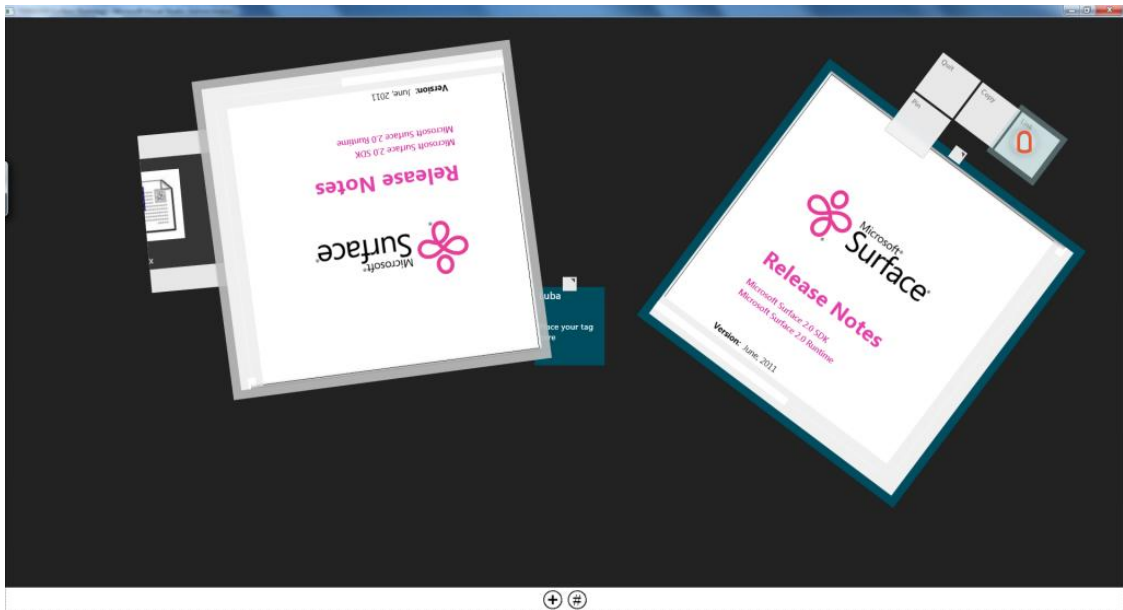
Surface - Jedes Dokument hat sein eigenes kontextbezogenes Menu. Dieses ist vom Look and Feel her an das Device Menu angelehnt. Das Device Menu des Hauptdokumentes erlaubt Kopien zur Ansicht oder virtuellen Verteilung zu erstellen.

## Surface Document Viewing/Presentation Copy



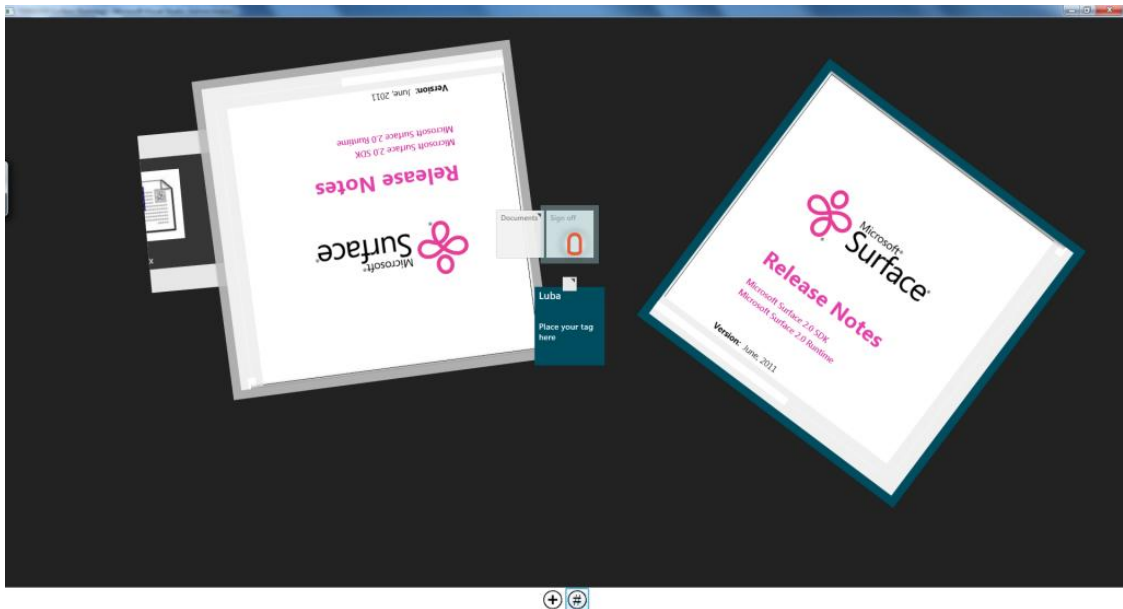
Surface - Im obenstehenden Bild wurde über das Document Menu des Hauptdokumentes eine Kopie erstellt und für die Ansicht für das Gegenüber gedreht. Das "Kind"-Dokument ist zum einen optisch durch den grauen Rand gekennzeichnet, zum anderen unterscheidet sich es auch dadurch, dass es nicht die selben Möglichkeiten hat wie das Hauptdokument. So lässt sich das "Kind"-Dokument zum Beispiel nur Schliessen und Pinnen damit es nicht absichtlich verschoben wird.

## Surface Document Viewing/Presentation Link



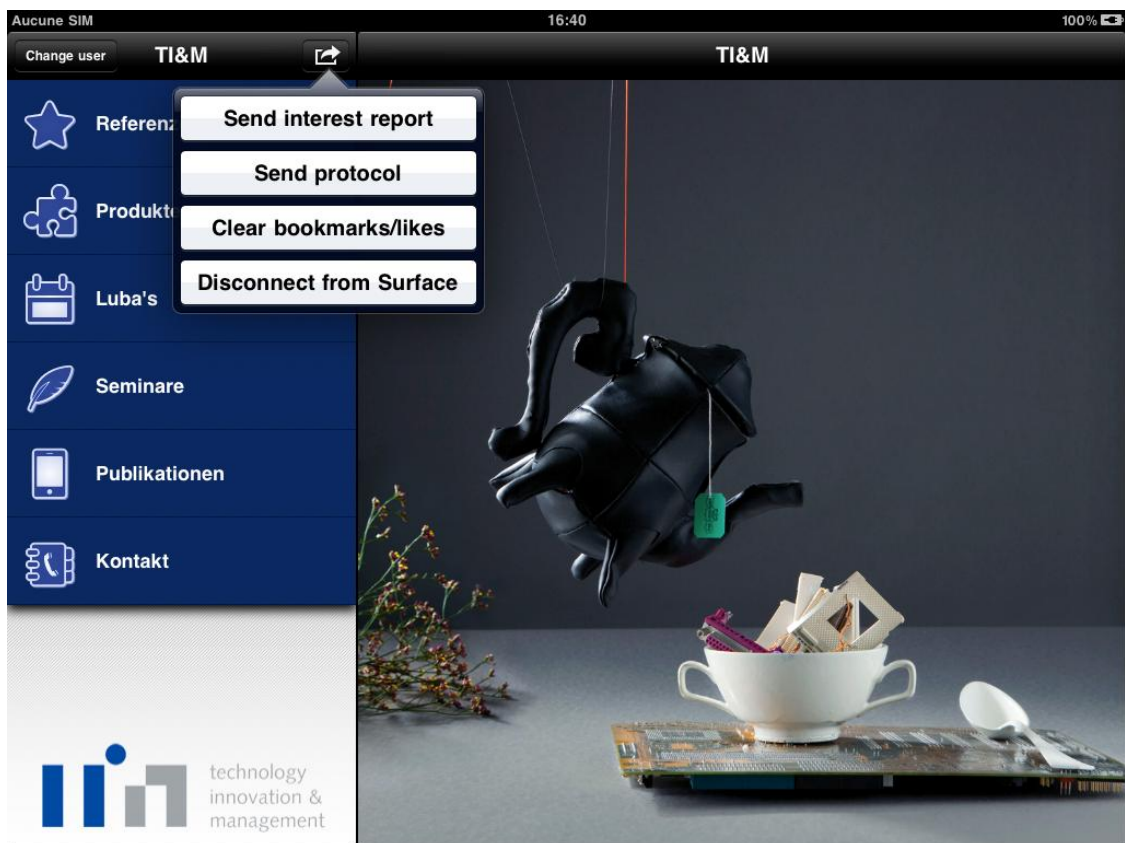
Surface - Werden mehrere Kopien eines Dokuments auf dem Tisch zur idealen Ansicht aller Teilnehmer verteilt, hat derjenige welcher einige Passagen zeigen möchte das Problem, dass alle Teilnehmer an einer unterschiedlichen Stelle oder Seite im Dokument sind. Aus Diesem Grund kann der Owner des Hauptdokumentes alle seine Kopien Linken so, dass die Ansicht und Scrollstatus zwischen allen Dokumenten identisch ist.

## Surface - Sign Off



Surface - Ist nun das Meeting beendet oder ein bestimmter Nutzer möchte die Sitzung verlassen, kann er sich, unabhängig davon ob seine Dokumente noch geöffnet sind, über das Device Menu abmelden. Alle auf der Benutzersession existierenden Elemente werden automatisch terminiert.

## iPad Disconnect from Surface



iPad - Ist nun das Meeting beendet oder ein bestimmter Nutzer möchte die Sitzung verlassen, kann er sich, entweder über sein Device Menu am Surface oder oder wie hier auf dem Screen-Shot angezeigt, vom Surface abmelden.

6.1 EINLEITUNG UND ÜBERSICHT

Ziel dieses Projektes ist die Umsetzung einer Lösung fürs "Meeting Management mit Microsoft Surface 2". Dies beinhaltet die Integration von mobilen Endgeräten sowie die Implementation von einem Verfahren, um nicht nur die Authentifizierung sondern auch den gesamten Datenaustausch zu regeln.

Beim Design der Lösung wurde darauf geachtet, nicht nur diejenigen Anforderungen welche durch die Vorstudie weiter konkretisiert wurden zu erfüllen, sondern das gesamte Projekt in einem konzeptionell ganzheitlichen Kontext zu sehen. Von unserem Product Owner bei der ti&m AG ist auch bekannt, dass die Lösung als POC für ihn sicher interessant ist und intern auch so genutzt wird. Die Lösung muss aber für einen produktiven Einsatz bzw. den Weiterverkauf an Kunden in verschiedene Dimensionen skalierbar sein.

So wurde aus dem „Meeting Management mit Microsoft Surface 2“ STEP was für Surface Table Exchange Plattform steht. Ziel dieser Lösung ist es, möglichst den geforderten Use Case abzudecken und zeitgleich möglichst viele Komponenten und Grundinfrastruktur für das zugrundeliegende System bereitzustellen.

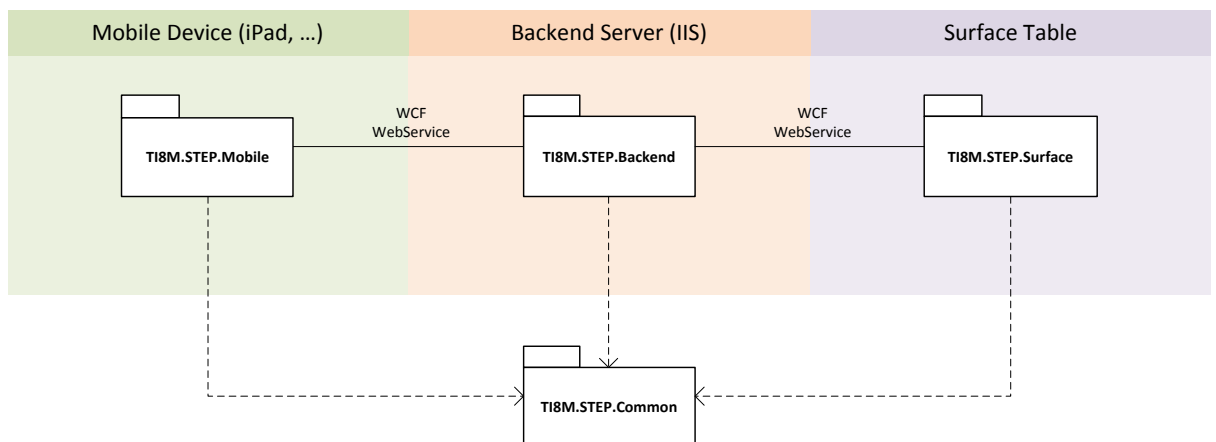
6.2 ARCHITEKTUR UND KOMPONENTEN

**Geografische Entkopplung von Präsentations- und Applikationslogik**

Als erster Designentscheid wurde die Präsentations- und Applikationslogik des Surface geografisch getrennt. Diese Trennung erfordert die Einführung einer einheitlichen Schnittstelle dieser Schichten welche auch die lokalen Systemgrenzen überwindet. Durch diese Entkoppelung ist es jetzt möglich das neu gewonnene Backendsystem dahingehend zu entkoppeln dass es in der Lage ist mit mehreren Surface Instanzen gleichzeitig zu kommunizieren.

Zukünftig können so zum Beispiel standortunabhängige Sitzungen von Surface zu Surface stattfinden bei denen Dokumente vom mobilen Endgerät nicht nur auf einem, sondern gleich noch beim Remote-Peer dargestellt werden können.

Aufgrund dieser Trennung und den einzelenen durch Systemgrenzen auferlegten Einschränkungen ist die Applikation wie folgt gegliedert.



Das obenstehende Diagramm stellt zum Einen die Hauptkomponenten bzw. Packages sowie deren Verteilung und Deployment auf den verschiedenen Zielsystemen dar. Ausserdem ist auch gleich die Art und Weise der Abhängigkeiten und Kommunikationswegen zwischen den Packages ersichtlich.

- **Surface**  
Die Surface Applikation welche auf dem Surface Table ausgeführt wird. Diese Applikation beinhaltet alle Komponenten die für die Darstellung und die Ausführung aller Funktionen am Surface benötigt werden.
- **Backend**  
Service Host für die Kommunikation zwischen Mobile Device und Surface. Diese Applikation beinhaltet alle Komponenten für das SessionManagement und die Authentifizierung.
- **MobileDevice**  
Das Mobile Endgerät welches für die Ansteuerung der Serviceschnittstellen verwendet wird. Im Rahmen dieses Projektes wird auf hier auf eine bereits existierende Lösung namens Workpad gesetzt die von einem externen Entwicklungsteam um die Surfaceintegration erweitert wird. Aus diesem Grund ist dieser Teil der Applikation nur als geschlossene Komponente dokumentiert.  
Einzige Ausnahme, welche zugleich auch Abhängigkeiten zu unserer Common Bibliothek hat, bildet die Mobile Device Implementation die aus zeitlichen Gründen als Workpad Simulator implementiert worden ist.

### **Backend-Komponente Authentifizierung**

Zuerst muss ein mobiles Gerät als ein Bestimmtes wiedererkennbar authentifiziert werden. Für diese Authentifizierung sind folgende Daten nötig:

**DeviceId**

Die eindeutige Identität des Gerätes. Diese wird bei der Installation der Client-Software generiert und ist über den gesamten Geräte-Lebenszyklus dieselbe.

**SurfaceId**

Die Identität des Surface, auf den der Client verbinden möchte.

**Token**

Ein One-Time Ticket welches zeitlich begrenzt ist und nur für einen Anmeldevorgang verwendet werden kann.

Dieser Token ermöglicht auf einfache Weise die Absicherung gegen Replay-Attacks.

### **Backend-Komponente Autorisierung**

Auf dem Backendsystem können dem verbundenen Gerät, durch die eindeutige Identifikation über die DeviceId, Rollen zugewiesen werden. Momentan ist ein rein rollenbasiertes Modell vorgesehen, bei dem in unserem Use Case jedes Gerät automatisch die Gäste-Rolle erhält. Dies kann aber im Bedarfsfall bis zu einem Rollen-Rechte basierten Modell ausgebaut werden.

Die Implementation der Autorisierung im Persistenz-Layer ist im Datenbankdiagramm im Kapitel *Datenbank Scripting in Backend.Database* ersichtlich. Die effektive Implementation im Session und User-Kontext ist im Klassendiagramm im Kapitel *Ticket Store* ersichtlich.

## Backend-Komponente Persistence

Damit einem Gerät, Surface oder mobile Device, gewisse Einstellungen wie zum Beispiel Rollenzuweisungen gemacht werden können, muss das Backend Informationen über die Laufzeit der Applikation hinaus speichern.

Das STEP Backend bietet dafür eine relationale Datenbank auf Basis von MS SQL an. Der Einsatz einer relationalen Datenbank bietet folgende Vorteile:

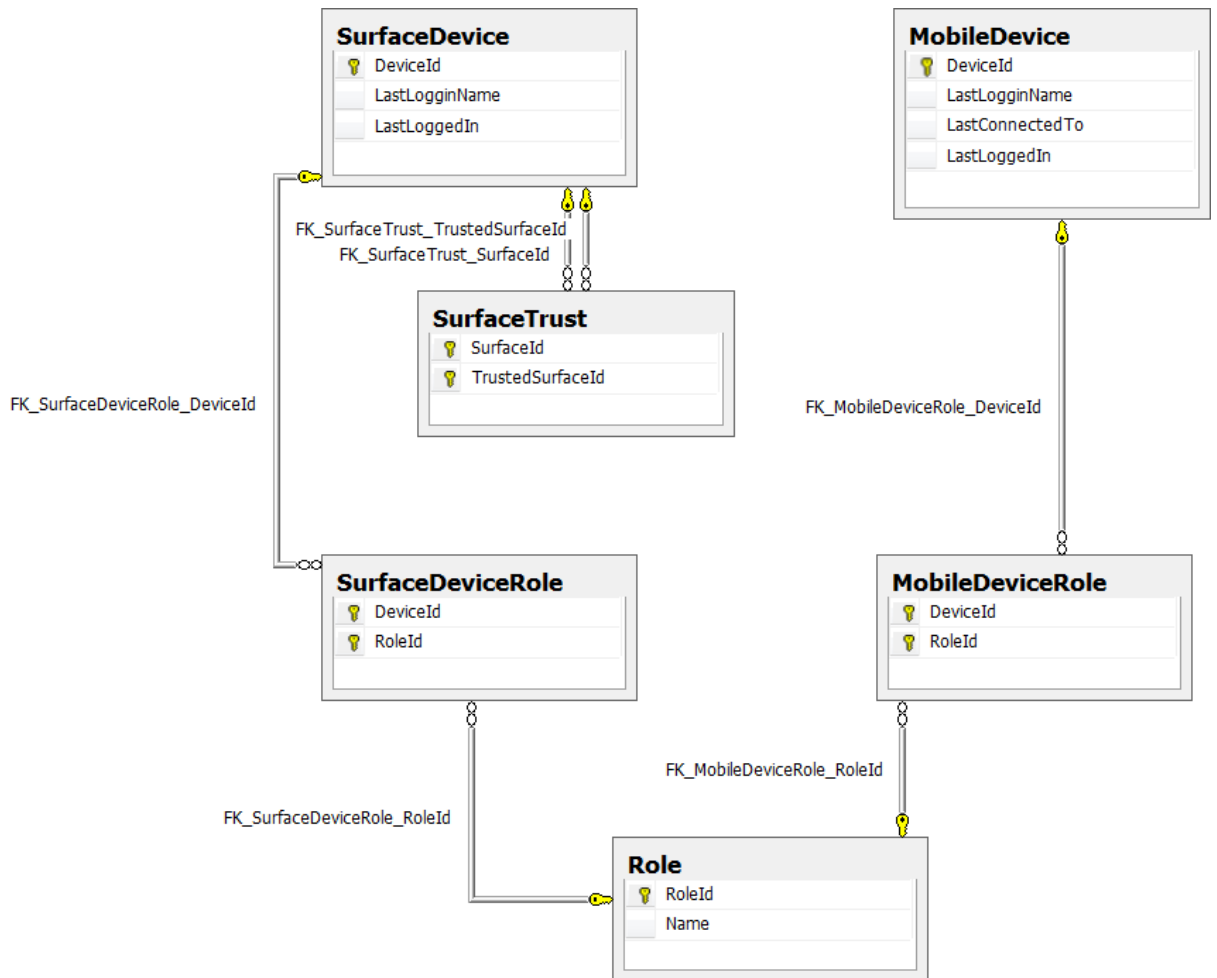
- **Skalierbarkeit**  
Auf Datenbank-Ebene kann auf jegliche Speicheranforderung einfach reagiert werden. Sei es Replikation, DB-Clustering, Parallelisierung von Zugriffen oder Performance Optimierung.
- **Integrierbarkeit**  
Daten können von oder zu Fremdapplikationen über die SQL-Server-eigenen Integration Services integriert werden.
- **Relationalität / Konsistenz**  
Durch die Abbildung der Zuordnungen von Geräten mit Hilfe des Datenbank-Schemas bekommen wir die Sicherstellung der referenziellen Integrität vom DBMS geschenkt. Die Definition von natürlichen Schlüsseln kann zusätzlich einer fehlerhaften Erstellung von Duplikaten vorgebeugt werden. Ausserdem erlaubt eine Datenbank die typisierte Speicherung von Informationen.

### Datenbank Scripting in Backend.Database

STEP bietet für die Schema-Pflege und Erweiterung eine gewisse Grundinfrastruktur. So erleichtern beispielsweise die wiederausführbaren Datenbank-Scripts die Installation und Pflege der Datenbankobjekte.

Weiter stellt STEP folgende Helper-Functions und Stored Procedures zur Verfügung:

- **fn\_CheckFunctionExists, fn\_CheckTableExists, fn\_CheckProcedureExists**  
Vereinfachen die Überprüfung und Fallbehandlung wenn das entsprechende Objekt bereits existiert.
- **pr\_DDLProcedure**  
Vereinfacht die Erstellung bzw. Änderung von Stored Procedures. Falls eine Stored Procedure bereits existiert, empfiehlt es sich diese nicht zu "droppen" und wieder neu zu erstellen, da die in operativen Umgebungen typischen Execute-Grants für low-privilegierte Benutzer dadurch verloren gehen würden.



### Backend.Persistence Implementation

Der Datenbankzugriff im Persistence-Layer von STEP ist aus untenstehenden Gründen, mittels dem klassischen ADO.net umgesetzt.

- Reduktion von Round-Trips  
Durch die Kapselung von den anwendungszentrierten Datenhaltungsoperationen in Stored Procedures können die Datenbank Roundtrips auf ein Minimum reduziert werden.
- Tiefer Technologoie-Stack  
Der Einsatz von zusätzlichen Frameworks, welche auf einem höheren Abstraktions-Level ansetzen, muss der zusätzliche Nutzen gegenüber den zusätzlichen Abhängigkeiten und Limitierungen abgeschätzt werden. In STEP verfolgen wir den Ansatz, den Technologie-Stack und somit Abhängigkeiten zu weiteren Bibliotheken nach Möglichkeit tief zu halten bzw. zu vermeiden.
- Reglementierungen und Anforderungen des Product Owners  
Aufgrund den oben aufgeführten Gründen und dem eher geringen Nutzen eines OR-Mappers, da wir einen sehr selektiven Datenzugriff und keine reine CRUD Anwendung umsetzen, wünscht der Product Owner den Einsatz eines OR-Mappers explizit nicht.

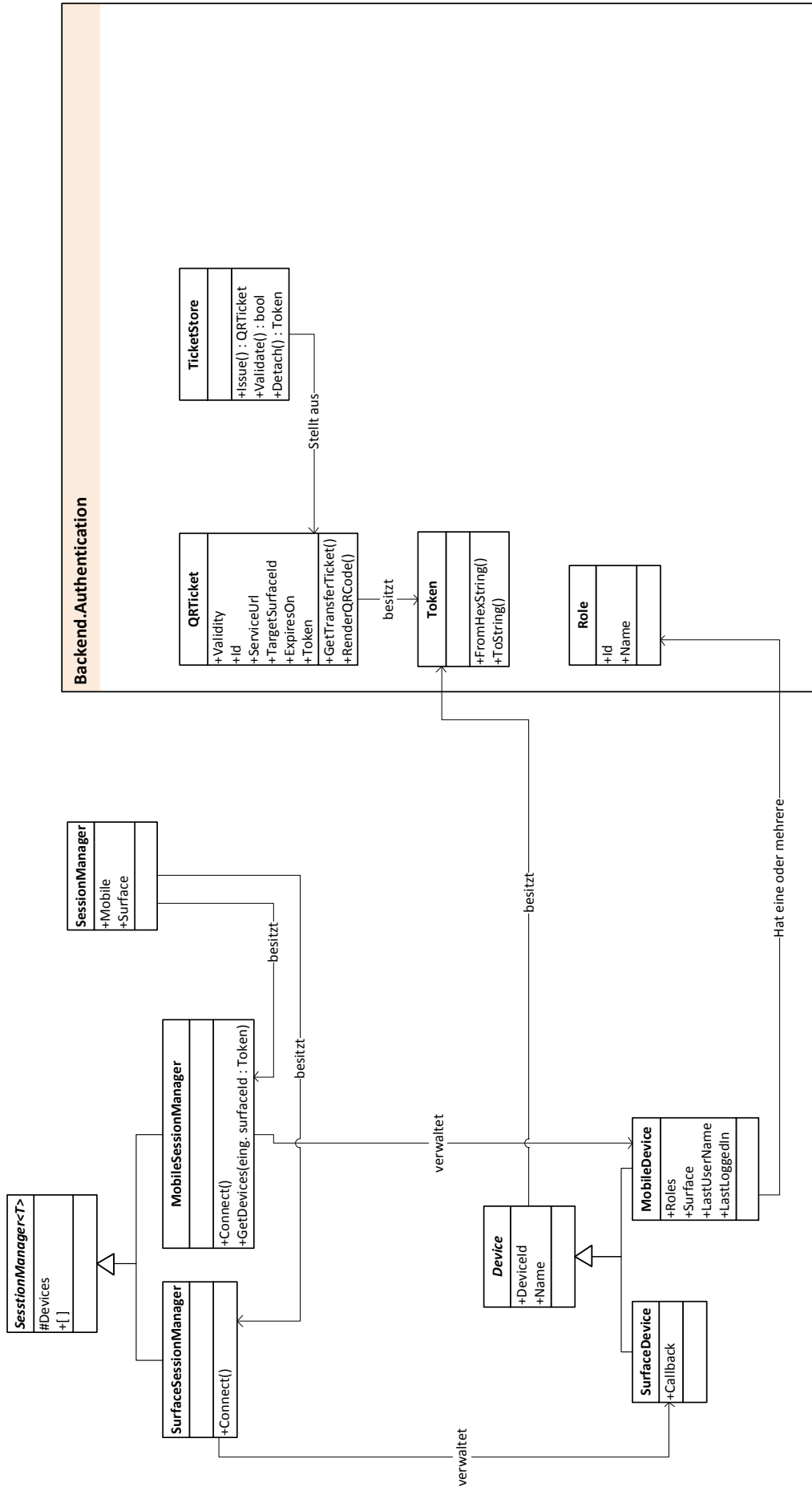
## Backend-Komponente Ticket Store

Ein wesentlicher Bestandteil des Authentifizierungsmechanismus ist das Ausstellen von zeitlich begrenzten Tokens sogenannten Tickets. Diese können zum Einen nur einmalig verwendet werden, zum Anderen müssen sie physikalisch am Surface gelesen werden und somit die Anwesenheit an einem Meeting erfordern.

Der Ticket Store verwaltet als alleinige Instanz alle aktiven Tickets, deren Ausstellung, Validierung und die spätere Vernichtung. Um dies auch auf möglichst tiefer Ebene zu erreichen, wurde der Ticket Store als Singleton implementiert, was die unabsichtliche Spezialisierung oder Offenlegung unerwünschter Funktionalität bei einer späteren Erweiterung von STEP verhindert. Zur Laufzeit stellt der Singleton zusätzlich sicher, dass lediglich eine Instanz dieser zentralen Klasse instanziiert werden kann.

Der Ticket Store wäre selber in der Lage die geforderten Aufgaben zu übernehmen. Damit er jedoch nicht überläuft und selbst nicht auf Wartungsdurchläufe angewiesen ist, bedarf es einem Mechanismus, damit sich ein abgelaufenes Ticket selbst terminiert. Dies ist mittels Inversion of Control zum QR-Ticket gelöst, welches bei Instanzierung einen Terminator Delegate erwartet und diesen nach Ablauf der Ticket-Gültigkeit aufruft. Hier wurde bewusst auf die Umsetzung mittels dem klassischen .NET Event-Modell verzichtet, damit zum Einen eine deterministische Ticket-Ausstellung möglich ist und zum Anderen kein Multicasting erwünscht ist.





## Backend-Komponente Session Manager

Damit auf dem Backend alle verbundenen mobilen Geräte und Surface Tische verwaltet werden können und auch nachvollzogen werden kann welches mobile Gerät gerade und überhaupt mit welchem Surface verbunden ist, bedarf es einer zentralen Einheit.

Folgende erschwerenden Faktoren gilt es zu Beachten:

- State-Injection in WCF generell sehr schwierig und nur mit Einschränkungen möglich.
- Die Service Bindings für mobile Geräte sind by design stateless und low level. Es stehen keinerlei Sitzungsinformationen zur Verfügung.
- Es ist nicht möglich auf Status- bzw. Sitzungsinformationen der Kommunikation zugrundeliegenden Statusinformationen (z.B. HTTP Session ID) zuzugreifen.
- Die Authentifizierung mittels One-Time Ticket ist in der ersten Phase anonym und kann nicht durch ein bekanntes und allgemein implementiertes Authentifizierungsverfahren ersetzt werden. Die Security Principal und Identity Injection von .NET entfällt daher als Möglichkeit ebenfalls.

Diese Problematik führt dazu, jeweils ein Sesseion Manager für Surface und einer für mobile Geräte innerhalb der normalen Application Domain zu pflegen. Die Sesseion Manager sollen vereinfachten Zugriff auf Statusinformationen für das momentane Device bieten. Dies wurde umgesetzt indem bei jeder Kommunikation ein Device seine ID übermittelt, mit der das Backend über die Session Manager die jeweiligen Device Instanzen laden kann.

Da die beiden Session Manager eine grosse gemeinsame Code Basis haben wurden sie mittels einer gemeinsamen, generischen Basisklasse implementiert. Die Klasse SessionManager (nicht generisch) dient als Container für die effektiven Session Manager Instanzen welche mittels Lazy Loding Pattern instanziiert werden.

Der Session Manager wurde so implementiert, dass er die Zugriffe auf die Device Objekte möglichst einfach handhabt. So verfügt der Session Manager zum Beispiel über einen Index-Operator, der das gewünschte Device mittels der DeviceId zurückgibt.

Code-Beispiel, wie ein mobiles Gerät mit der ID mobileDeviceId über den SessionManager suspended wird im Backend:

```
SessionManager.Mobile[mobileDeviceId].Suspend();
```

## Backend-Komponente Trusts

Dieser Teil ist erst als Konzept und konfigurativ im Persistenz-Layer in STEP vorhanden. Die Idee ist die Erstellung von Surface zu Surface Trust-Relationships. Diese sollen die Grundlage bilden für jede Art von Surface zu Surface Kommunikation oder zum Beispiel Meeting-Erweiterung auf weitere "Trusted" Surface Tables.

## 6.3 FUNKTIONEN

### Charmsbar

Für einige Funktionen werden auf dem Surface Buttons benötigt. Einerseits um ein Gerät anzumelden, andererseits um die bereits angemeldeten Geräte aufzulisten. In einem ersten Entwurf wurden die Buttons fix an den Tischrändern dargestellt und mit einem Halbkreis farblich abgetrennt.



Entwurf der fixen Buttons

Der Nachteil dieser Darstellung ist, dass die Buttons ständig angezeigt werden, obwohl sie pro Meeting für jeden Teilnehmer nur einmal benötigt werden. Daher wäre eine Lösung, in welcher die Buttons nur dargestellt werden, falls sie auch wirklich gebraucht werden, optimal.

Während der Projektarbeit wurde die Consumer Preview von Windows 8 vorgestellt, in welcher die Charms Bar zum ersten Mal von Microsoft eingesetzt wurde. Dabei werden selten genutzte Funktionen innerhalb einer Bar dargestellt, welche man von der Seite des Bildschirms her mit einer Wisch-Bewegung sichtbar macht. Diese Bar wird Charms Bar genannt.



Entwurf der Charms Bar

Der Nachteil dieses Ansatzes ist, falls man noch nie mit einer Charms Bar gearbeitet hat, man nicht weiss, dass diese überhaupt existiert. Dies ist allerdings ein generelles Problem mit dem Wechsel auf Touch Input Devices und dürfte für viele User eine Umstellung der Arbeitsweise bedeuten. Da aber Microsoft dieses Konzept mit Windows 8 etablieren will, kann man davon ausgehen, dass sich die Charms Bar nach einer kurzen Umgewöhnungsphase zu einem normalen Werkzeug, wie die Ribbons, entwickeln wird.

Daher haben wir uns dazu entschieden, die Charms Bar für die Darstellung unserer Buttons zu verwenden.



Finale Version Charms Bar

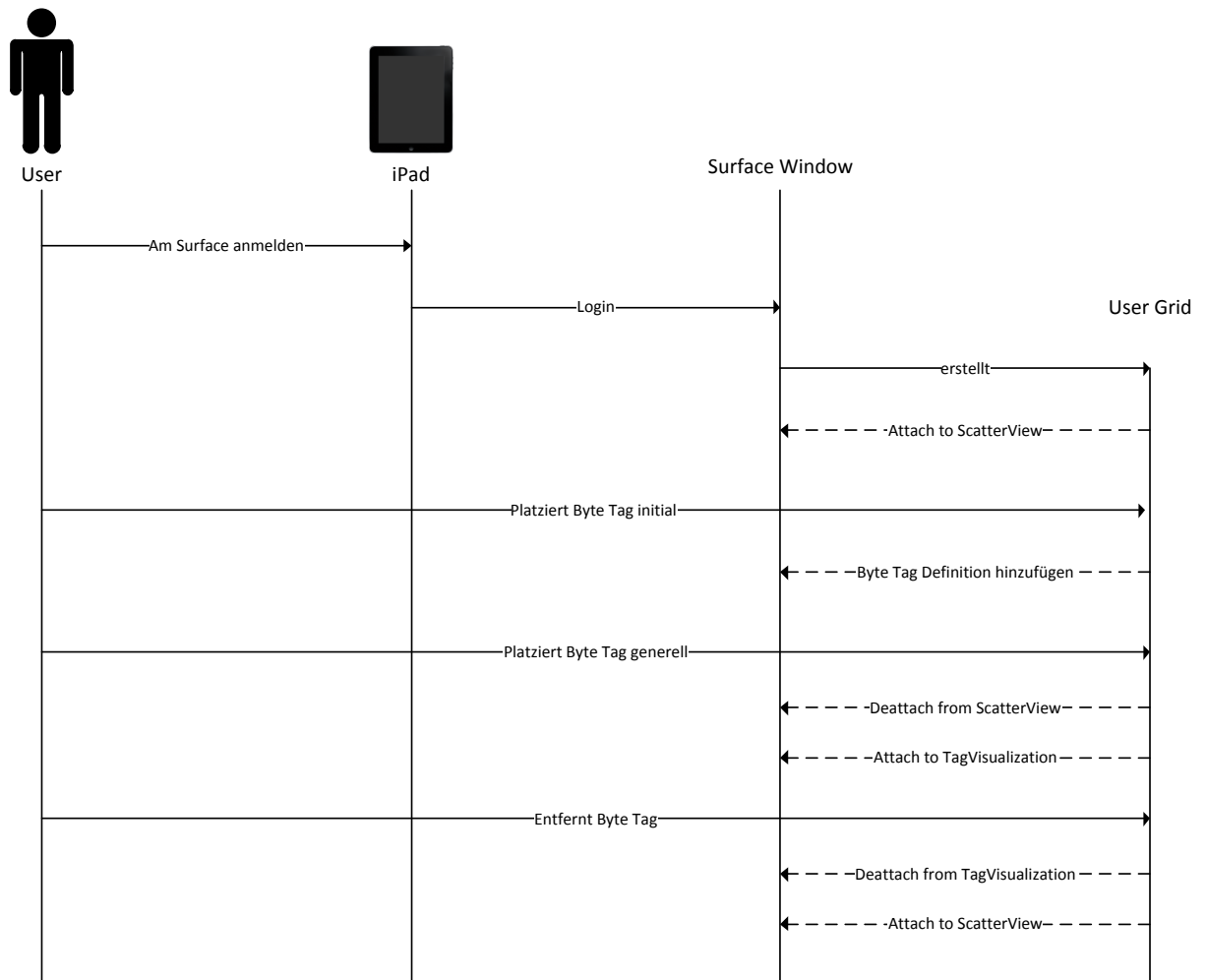
## Devicetracking

Für das Devicetracking auf dem Surface standen zwei Alternativen zur Debatte. Die Surface SDK stellt für die Darstellung mehrerer Elemente die ScatterView-Klasse zur Verfügung, wobei ein einzelnes Device als ScatterViewItem dargestellt wird und mittels Bewegungen gesteuert wird. Als Alternative wird die TagVisualizer-Klasse angeboten, wobei ein einzelnes Device als TagVisualisation dargestellt wird und mit den ByteTags, welche bereits in der Vorstudie genauer analysiert worden sind, bewegt werden.

Der Vorteil des ScatterView Ansatzes ist, dass man die Device-Instanz intuitiv mit den Finger auf dem Surface bewegen kann, der Vorteil des TagVisualisation Ansatzes, dass man egal wo man sich um den Surface Tisch befindet, seine eigene Device-Instanz nach auflegen seines Tags sofort bei sich hat, anstatt sich über den Tisch beugen zu müssen und die Device Instanz zu sich rüber ziehen muss.

Wir haben keinen der beiden Ansätze ausgeschlossen, da beide Vorteile zu wichtig für die Bedienung sind. Daher kann die Device-Instanz zwischen den Darstellungen als ScatterViewItem und TagVisualisation hin und her wechseln.



Bei der Erstellung der Device-Instanz, wird dieses zunächst als ScatterViewItem dargestellt. Sobald auf diese Instanz ein ByteTag aufgelegt wird, wird diese Device-Instanz zusammen mit dem Wert des ByteTag auf dem TagVisualizer registriert und so wird jedes Mal, wenn dieser ByteTag erkannt wird, die Instanz von der ScatterView gelöst und als TagVisualisation dargestellt. Sobald das ByteTag entfernt wird, wird die Instanz wieder der ScatterView hinzugefügt.



## Dokumenthandling auf Surface

In der Workpad Applikation werden die Dokumente in einzelne Kategorien unterteilt. Sobald sich ein iPad sich via Workpad am Surface angemeldet hat, wird der Dokument Index an den Tisch übertragen. Dieser Index ist eine Liste aller Dokumentkategorien und deren Dokumente. Die einzelnen Kategorien werden als ElementMenuItem auf der Device Instanz dargestellt. Ein Nachteil der ElementMenuItems ist, dass man nur bis zu acht Elemente auf der selben Ebene darstellen kann. Da aber in der Workpad Applikation nur die sieben ersten Kategorien dargestellt werden, kann dieser Nachteil in der STEP Applikation vernachlässigt werden.

Zur Darstellung der einzelnen Kategorien standen die Klassen LibraryBar, LibraryStack und Library Container zur Debatte.

	LibraryBar	LibraryStack	LibraryContainer
<b>Bild</b>			
<b>Beschreibung</b>	Die einzelnen Dokumente werden in einer Bar dargestellt und können seitwärts durchgescrollt werden	Die einzelnen Dokumente werden in einem Stack dargestellt, klar erkennbar ist nur das oberste. Um auf die darunter liegenden Dokumente zu greifen zu können, müssen alle darüber liegenden Dokumente angetippt werden um dieses ans Ende des Stack zu platzieren	Der LibraryContainer kann zwischen einer BarView und einer StackView wechseln, welche jeweils wie die LibraryBar- und die LibraryStack-Klasse aussehen
<b>Vorteile</b>	Alle Dokumente können schnell erreicht werden	Nimmt nicht allzu viel Platz in Anspruch	Vereint die Vorteile der LibraryBar und des LibraryStack
<b>Nachteile</b>	Nimmt viel Platz ein	Unter Umständen muss lange durch den Stapel getippt werden, um sein Dokument zu finden	Dokumentenbindung ist geringfügig komplexer als die Alternativen

Der Entscheid fiel auf den LibraryContainer, ausschlaggebend war die Flexibilität.

Sobald eine Kategorie ausgewählt wird, wird diese und deren Inhalt mit Hilfe der LibraryContainer Klasse dargestellt. Zur Erkennbarkeit, welche Kategorien zu welchem Device gehört, werden die LibraryContainer mittels einer dünnen Linie zum dazugehörigen Device-Instanz verbunden. Die einzelnen Dokumente innerhalb der Dokumente können aus dem Library Container heraus gedragt werden und irgendwo auf dem Surface wieder gedropt werden. An dieser Stelle wird das Dokument dann dargestellt.

## Dokumentanzeige

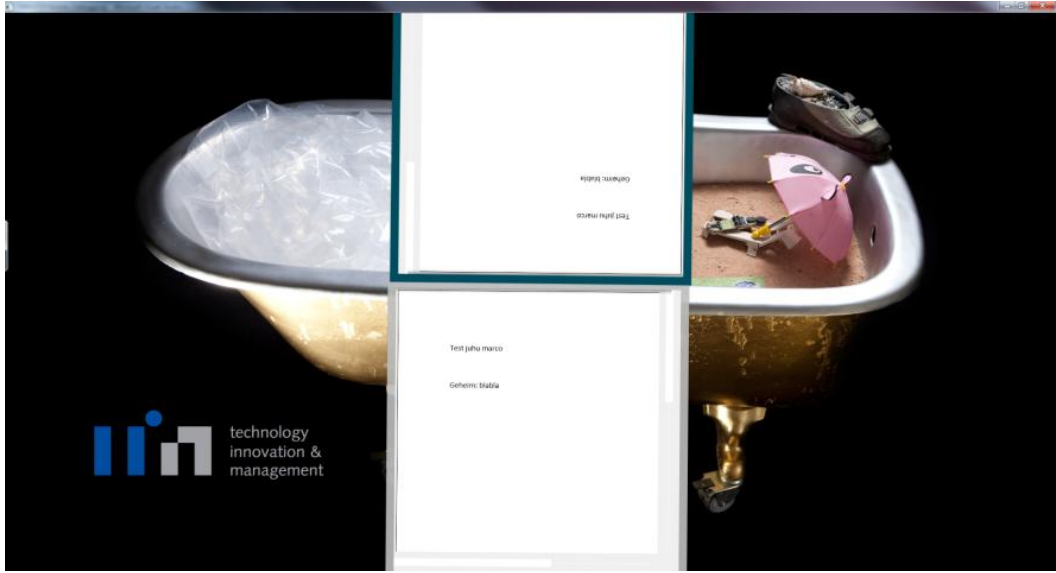
Da wir uns im Rahmen des Descoping dazu entschlossen haben, auf die Editier-Funktionen zu verzichten, kann für die einzelnen Dokumenttypen theoretisch dieselbe Darstellung verwendet werden. Gerade bei den verschiedenen Office Formaten ist es am naheliegendsten, jeweils das von

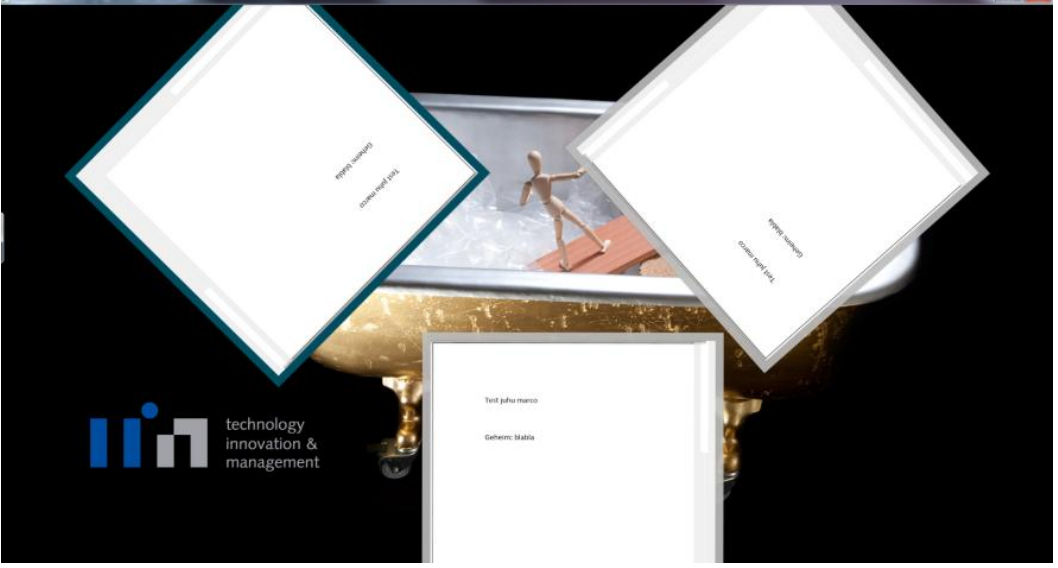
Microsoft entwickelte Dateiformat für Dokumente XPS (XML Paper Specification) zu verwenden. Für das Dateiformat PDF gibt es ebenfalls 3rd-Party Software, welche eine Konvertierung zu einem XPS ermöglichen. Als alternativer Ansatz stand auch via WinForm Support in WPF mit Hilfe des WindowsFormHost-Controls zur Debatte (Quelle: <http://www.screencast.com/t/JXRhGvzvB>). Da man jedoch somit zwei unterschiedliche DocumentViewer Klassen benötigt hätte und diese beide hätte pflegen und stylen müssen, haben wir uns dazu entschlossen, eine 3rd-Party Library zu verwenden und somit die PDF Dokumente zu XPS zu konvertieren.

Um die XPS Dokumente darzustellen, wird die Klasse System.Windows.Controls.DocumentViewer verwendet. Diese wird innerhalb eines ScatterViewItems dargestellt und kann somit nach belieben gedreht, vergrößert und bewegt werden. Damit man das Dokument auch scrollen kann, wird innerhalb des ScatterViewItems auch ein SurfaceScrollViewer dargestellt. Die Mutation des ScatterViewItem kann daher nur am Rande des ScatterViewItems vollzogen werden, die Bewegungen innerhalb des DocumentViewers werden als ScrollBefehle interpretiert.

Damit man nicht aus Versehen sein Dokumente bewegt, anstatt zu scrollen, kann man die ScatterViewItems auch an Ort und Stelle fest pinnen.

Zur Problematik, wenn mehrere Personen sich Dokumente auf dem Tisch ansehen wollen, standen folgende Möglichkeiten zur Auswahl:

Spiegelmodus		
	Beschreibung	Der Tisch wird horizontal, vertikal oder sowohl horizontal als auch vertikal gespiegelt. Somit sieht der Benutzer gegenüber genau dasselbe wie derjenige, welcher mit dem Tisch interagiert.
	Vorteil	Übersichtlichkeit
Nachteil	Sobald mehr als zwei Benutzer um den Tisch stehen, stimmt die Positionierung nicht mehr für alle Benutzer	

Lokale Dokumenten Kopie		
	Beschreibung	Von den einzelnen Dokumenten wurde jeweils eine Kopie erstellt, somit kann für jeden Benutzer seine eigene Kopie erstellt werden, die er nach belieben ausrichten kann.
	Vorteil	Jeder Benutzer kann sein Dokument nach belieben ausrichten
Nachteil	Unübersichtlichkeit, falls zu viele Kopien erstellt werden	

Da der Spiegelmodus ab drei Personen und mehr keinen Vorteil mehr bringt, haben wir uns dagegen entschieden und die Lösung mit den lokalen Kopien verfolgt. Dabei lassen sich die Dokumentkopien mit ihrem Host linken, so dass jegliche Scroll-Bewegung auf dem Dokumentenhost auch auf seine Kopien dargestellt wird. Somit kann gewährleistet werden, dass für alle Benutzer stets dieselbe Stelle des Dokuments angezeigt wird.

### Dokumentkonvertierung

Da die Dokumente zur Anzeige alle im Dateiformat XPS verfügbar sein müssen, wurde für alle möglichen Dateitypen ein Converter geschrieben, welcher jeweils das Dokument in ein XPS File konvertiert. Für die Office Dokumente wird hierbei die Microsoft.Office.Interop Library verwendet.

Beim Dateiformat PDF wurde auf die 3rd-Party Library von Aspose zurückgegriffen. Dieser Entscheid war mit einigen Problemen verbunden, den die Aspose Library schreibt das XPS File mit dem Language Wert "und." für undefiniert. Dies führte zu einigen Problemen, daher wird das PDF Dokument sicherheitshalber als temporäres XPS Dokument zwischengespeichert und danach wird dieses temporäre Dokument wieder gelesen und mit gesetztem Language Wert erneut gespeichert. Möglicherweise könnte dieser Umstand mit einer besseren Library umgangen werden.

Für die Dokumentkonvertierung wird ein Dictionary verwendet, welches als Key eine File-Extension und als Value eine Klasse, welche IDocumentConverter implementiert, erwartet. So kann die Liste der unterstützten Dokumententypen mit dem Schreiben eines neuen Converters sehr einfach erweitert werden.



## Dokument-Caching

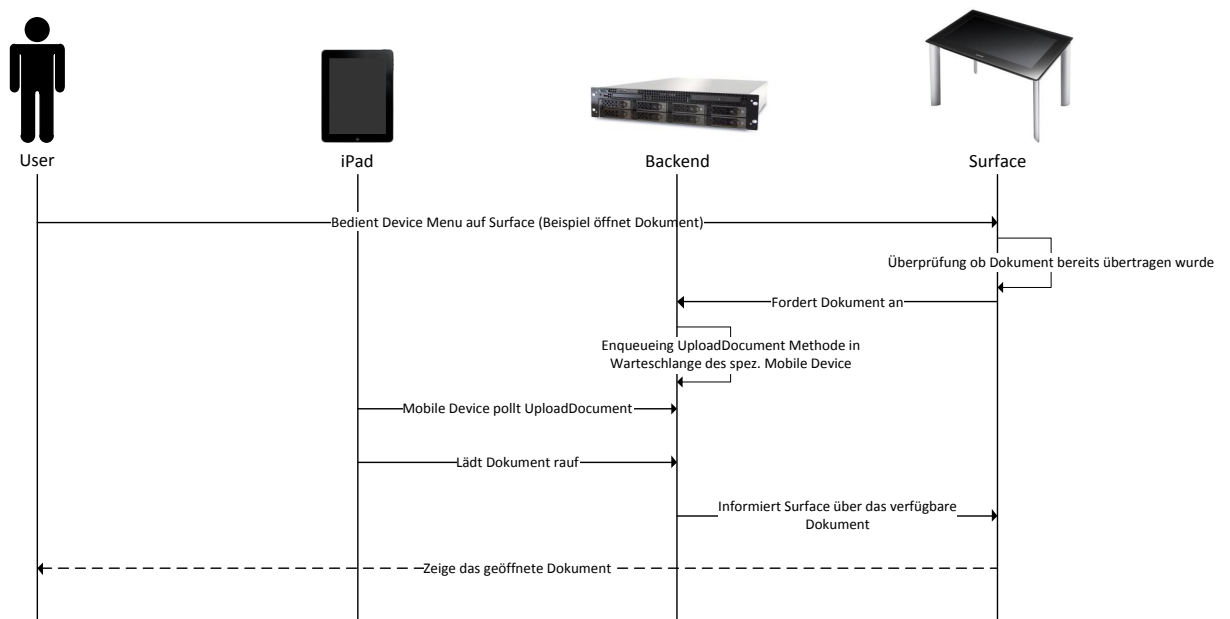
Um die Dokumente zwischenspeichern gibt es den sogenannten SuperiorStorage, welcher für jedes Device einen Ordner erstellt, in welchem dessen Dokumente einerseits im original Format und andererseits auch im XPS Format zwischen gespeichert werden. Sobald sich das Device abmeldet, wird dessen Ordner mit allen Dokumenten gelöscht. Sobald STEP selber terminiert wird, wird für jedes Device dessen Ordner mitsamt den zugehörigen Dokumenten gelöscht.

Die Dokumente werden anhand ihrer Revision hinterlegt, d.h. sobald eine neue Revision eines Dokuments hochgeladen wird, wird auch diese verwendet.

## Dokument-Übertragung

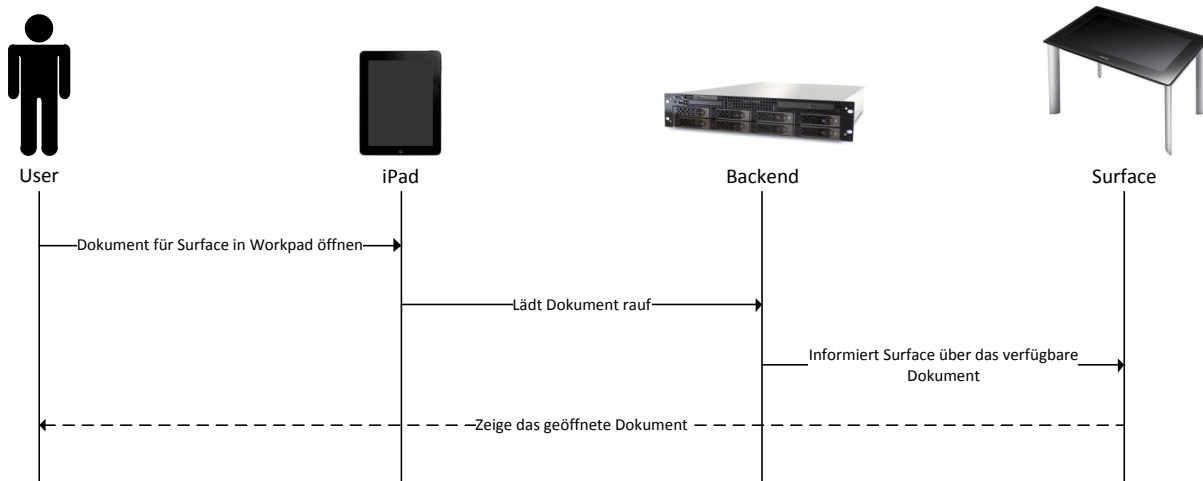
Um ein Dokument an den Surface zu Übertragen werden dem Benutzer zwei Möglichkeiten angeboten: Er kann das Dokument auf dem Surface öffnen, oder er kann auf dem iPad auswählen, dass das Dokument auf den Surface übertragen wird.

Generell wird zur Übertragung der Dokumente mit einer PriorityQueue gearbeitet. Wenn sich ein mobiles Gerät erfolgreich angemeldet hat und der Dokument-Index übertragen wurde, werden als Low Priority Tasks die Übertragung der Thumbnails in Auftrag gegeben. Sobald der Benutzer ein Dokument aus einer Dokumentkategorie (sprich aus dem LibraryContainer) beginnt zu ziehen, wird ein Upload Befehl als High Priority Task an das iPad gesendet. Somit wird gewährleistet, dass ein geordnetes Dokument vor den Thumbnails übertragen wird und dass nur jene Dokumente hochgeladen werden, welche wirklich benötigt werden.



Das Sequenzdiagramm zeigt den Ablauf der Dokumentübertragung. Die Übertragung der Thumbnails wird in diesem Diagramm ausgeklammert.

Wählt ein Benutzer die alternative Variante und sendet ein Dokument vom iPad aus an den Surface, wird das Dokument direkt per Webservice auf das Backend übertragen. Ist es dort angekommen, wird es weiter an den Surface geschickt. Eine automatische Darstellung auf dem Surface direkt nach der Übertragung wird momentan noch nicht unterstützt.



Das Sequenzdiagramm zeigt die alternative Variante, indem ein Dokument vom iPad aus geöffnet wird. Die gezeigte automatische Darstellung wird momentan noch nicht unterstützt.

## Anmeldeprozess

Um den Anmeldeprozess für ein mobiles Gerät zu initiieren, tippt der Benutzer den "Add"-Button auf der Chamrs Bar an. Dadurch wird eine MobileDevice-Instanz erstellt und ein One-Time Ticket wird vom Backend angefordert. Solange der Surface das Ticket noch nicht erhalten hat, wird die MobileDevice-Instanz mit einer Loading Animation dargestellt.

Das One-Time Ticket enthält ein QR-Code, welcher auf der MobileDevice-Instanz dargestellt wird. Zusätzlich wird der Benutzer informiert, wie lange sein Ticket noch gültig ist. Die Information über die Ticket-Gültigkeitsdauer wird numerisch als auch grafisch dargestellt. Verstreicht diese Frist ohne Interaktion des Benutzers, wird die MobileDevice-Instanz gelöscht.

Um das iPad anzumelden muss der QR-Code von der Workpad Applikation gelesen werden solange das Ticket gültig noch ist. Der QR-Code wird gelesen, indem die Kamera des iPad über den auf dem Surface dargestellten Code gehalten wird.

### Der QR-Code enthält

- Host: Die URL zum Backend Webservice
- Token: Die ID des One-Time Tickets
- Surface: Die ID des Surface

Der Token, die Surface ID, die iPad ID sowie der Benutzername des iPad Users werden nun an den Webservice Backend gesendet. Sobald das Backend diese Information erhält und validiert hat, wird der entsprechende Surface informiert über die Anmeldung des mobilen Geräts. Nun wird das iPad in der entsprechenden MobileDevice-Instanz dargestellt.

Nach erfolgreicher Anmeldung sendet das iPad den Dokumenten Index an das Backend.

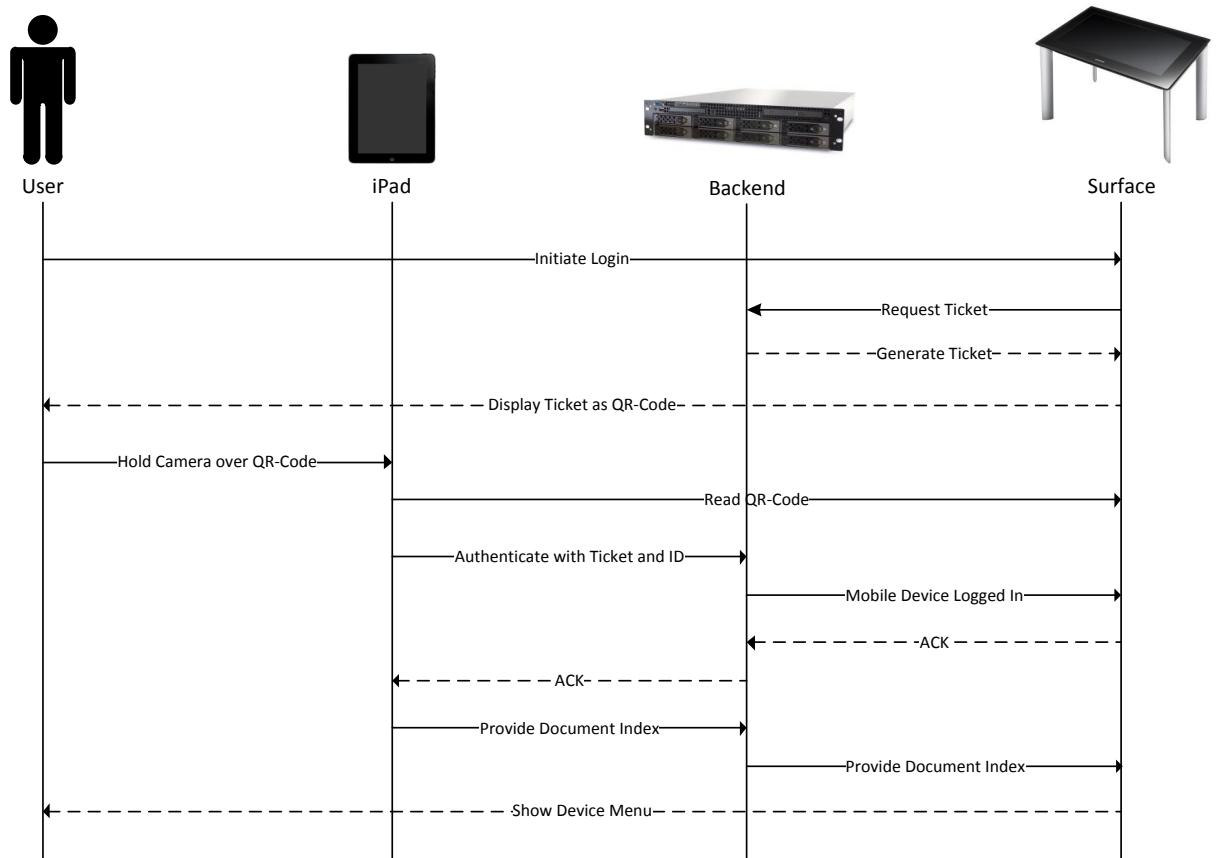
### Beispiel eines QR Codes



HOST=http://localhost/STEP/MobileDeviceService.svc

TOKEN=45BAAD81E5DDF119C1906F3DB1245715EA6A728551FEFC9437D5056F8E4485C9

SURFACE=819D2371817020BF1613EFFDDF7503FACFE6448263F6199D3BB76524B3840083



### Suspend-Mechanismus

Sobald auf dem iPad die Workpad Applikation in den Hintergrund tritt, wird STEP informiert, dass das MobileDevice in den Suspend Modus gewechselt hat. Daraufhin wird die Device Instanz und alle ihre Dokumentkategorien innerhalb von STEP ebenfalls suspended, d.h. es können keine Dokumente mehr geöffnet werden und keine Dokumentkategorien angeschaut werden. Dies muss zwingend so umgesetzt werden, da vom iPad keine Dokumente mehr hochgeladen werden können, solange die Workpad Applikation inaktiv ist. Somit wird verhindert, dass die PriorityQueue der Workpad Applikation aufgefüllt wird, solange die Workpad Applikation diese nicht abarbeiten kann.

Sobald die Workpad Applikation wieder in den Vordergrund tritt, wird die Resume Methode aufgerufen. Damit wird die Device Instanz und die zugehörigen Dokumentkategorien wieder resumed.

### Abmeldeprozess

Das MobileDevice kann sich über mehrere Methoden abmelden. Entweder kann das Device über die Logout-Funktion der Workpad Applikation abmelden. Hierbei wird der Logout Befehl an den Surface gesendet, dieser beendet die Device-Instanz und löscht dabei jegliche Dokumente des Benutzers aus dem Zwischenspeicher.

Eine alternative Möglichkeit sich abzumelden ist, das Gerät über das Device Menü auf dem Surface direkt abzumelden. Hier wird auf dem Surface das Device Menü und jegliche Dokumentkategorien in den Suspendstatus gesetzt und in der PriorityQueue des Workpad wird der Befehl zur Abmeldung enqueued. Der Logout Befehl wird an den Surface gesendet, sobald die Workpad Applikation den Logout Request gepollt hat. Somit wird die Device Instanz erst nach Rückmeldung des iPads definitiv gelöscht.

Die letzte Möglichkeit des Abmeldens ist eher als Sicherheitsmassnahme zu verstehen. Sollte sich ein Device über fünf Minuten im Suspend-Status befinden, wird die Device Instanz aus Sicherheitsüberlegungen terminiert und die zugeordneten Dokumente gelöscht.

### **Queueing Mechanismus**

Da es mit dem iPad keine Möglichkeit gibt ein Two-Way Binding aufzubauen, werden die Methodenaufrufe, welche das iPad aufrufen sollte in eine Queue geschrieben. Diese Queue wird von der Workpad Applikation kontinuierlich gepollt. So weiss die Workpad Applikation, wann sie ein Dokument hochladen oder ein Device abmelden soll.

Dabei wird die Queue in zwei Prioritäten unterteilt. Als Low Priority werden die Upload Befehle der Thumbnails eingetragen, alle anderen Tasks werden als High Priority hinterlegt. So kann sichergestellt werden, dass der Upload eines Dokuments nicht durch Uploads von Thumbnails hinausgezögert oder gar blockiert wird.

### **Sessiontracking**

Die Sessions werden zentral auf dem Backend gepflegt und verwaltet. (vgl. Session Manager Beschreibung im Kapitel Architektur und Komponenten)

Die Aktivität des Mobile Device wird über den Suspend Mechanismus und die Polling Aktivität getrackt. Wenn das Gerät suspended wird, wird es aus Sicherheitsgründen nach einer gewissen Zeit automatisch abgemeldet. Für mehr Informationen siehe Funktionsbeschreibung des Suspend-Mechanismus.

### **QR-Code**

Zur Generierung der QR-Codes wird die Zebra Crossing Library verwendet. Diese wird mit Hilfe einer Klasse, die das Interface IQRCodeAdapter implementiert, adaptiert. So kann auch eine andere Library zur Generierung der QR-Codes verwendet werden und man ist nicht zwingend auf diese Library oder QR-Codes im Allgemeinen angewiesen. Es wäre durchaus möglich, die von Surface üblicherweise verwendeten Microsoft Tags zu gebrauchen. Da wir im Rahmen dieses Projekt aber auch auf die iPad Integration angewiesen sind, haben wir uns für die gebräuchlicheren QR-Codes entschieden.

## 6.4 SCHLUSSFOLGERUNGEN

### Resultate

Deliverable	Ziel	erfüllt
<b>Verbindungsaufbau mit mobilem Endgerät</b>		
	Null, ein oder mehrere mobile Endgeräte können am Surface angemeldet werden.	erfüllt
	Auf dem Surface wird ein Menü pro angemeldetem Gerät angezeigt.	erfüllt
	Ein Gerät kann über den in Verbindungsaufbau mit mobilem Endgerät beschriebenen Vorgang angemeldet werden.	erfüllt
<b>Kommunikation mit dem mobilen Endgerät Deliverable</b>		
	Die Kommunikation muss verschlüsselt stattfinden.	erfüllt, konfigurativ
	Die Verbindung muss in der Lage sein, ein Dokument bis 2MB innerhalb von einer Sekunde zu übertragen.	in unsere Testumgebung erfüllt, abhängig von Netzwerkbandbreite
	Das mobile Endgerät soll via IP-Protokoll mit dem Surface kommunizieren.	erfüllt
	Es wird vorausgesetzt, dass sich die Surface-Applikation und das mobile Endgerät via IP erreichen können.	Voraussetzung, kein eigentliches Ziel
<b>Kommunikation mit Dokument Management System</b>		
	Es ist eine Voraussetzung, dass das mobile Gerät die Credentials für das Dokument Management System hinterlegt hat, oder diese Möglichkeit anbietet.	angepasst aus Sicherheitsgründen
	Die Credentials werden über eine sichere Verbindung an den Surface übertragen.	angepasst aus Sicherheitsgründen
	Die Surface Applikation kann sich zum Dokument Management System mit den gelieferten Credentials verbinden.	angepasst, das Dokument wird vom iPad hochgeladen
<b>Session Management und Device Tracking</b>		
	Ein angemeldetes Gerät muss nach einer definierten Zeit automatisch abgemeldet werden. (Session-Timeout)	erfüllt
	Es muss ein Mechanismus realisiert werden, der den vorzeitigen Sessionabbruch verhindert.	erfüllt
	Die Session für ein Gerät kann vom Benutzer manuell terminiert werden.	erfüllt
	Tracking über einen getaggten Gegenstand ist optional und nicht im Scope.	erfüllt
<b>Dokument Handling auf Surface</b>		

Im Menü pro Gerät werden Dokumente, die zur Verwendung mit dem Surface markiert wurden, aufgelistet.	erfüllt
Ein Benutzer kann ein aufgelistetes Dokument öffnen und Surface zeigt dieses an.	erfüllt
Der Benutzer kann auf dem mobilen Endgerät ein Dokument direkt auf den Surface übertragen.	erfüllt
Dokumenttypen die unterstützt werden sollen: Word (docx), Excel (xlsx), Power-Point (pptx) oder PDF (pdf)	erfüllt
Eine Unterstützung von Bild-Formaten wie zum Beispiel JPEG, Bitmap oder PNG ist optional.	nicht erfüllt
Die Dokumente werden nur als Read-Only geöffnet. Eine Editier-Funktion wird nicht angeboten.	erfüllt
Die maximale Anzahl Personen, die STEP gleichzeitig verwenden wird durch die Grösse des Surface auf acht Personen limitiert.	Anforderung

**Die Kommunikation muss verschlüsselt stattfinden**

Dies muss in der Konfiguration definiert werden, die Verbindung funktioniert über http und auch über https.

**Die Verbindung muss in der Lage sein, ein Dokument bis 2MB innerhalb von einer Sekunde zu übertragen**

Dies wurde in der Testumgebung erfüllt. Die Geschwindigkeit ist natürlich abhängig von der Bandbreite und wird beeinträchtigt durch eventuell vorhandenen FireWall, Virens Scanner etc.

**Es ist eine Voraussetzung, dass das mobile Gerät die Credentials für das Dokument Management System hinterlegt hat, oder diese Möglichkeit anbietet**

Wurde aus Sicherheitsgründen gedroppt, da man die Sicherheit der Verbindung zwischen iPad und Surface nicht 100% gewährleisten kann

**Die Credentials werden über eine sichere Verbindung an den Surface übertragen**

Wurde aus Sicherheitsgründen gedroppt, da man die Sicherheit der Verbindung zwischen iPad und Surface nicht 100% gewährleisten kann

**Die Surface Applikation kann sich zum Dokument Management System mit den gelieferten Credentials verbinden**

Wurde aus Sicherheitsgründen gedroppt, da man die Sicherheit der Verbindung zwischen iPad und Surface nicht 100% gewährleisten kann

## Probleme

Durch Verzögerung, verursacht durch fehlende Ressourcen im Workpad Team, musste für Testing Zwecke ein Test Projekt TI8M.STEP.Mobile erstellt werden, welches die iPad Interaktion simuliert. Dadurch, und allgemein durch die Abhängigkeit von der Workpad Applikation, ging viel Zeit verloren, welche für andere Aufgaben aufgewendet hätte werden können.

Die Stabilität zwischen der Verbindung von Surface zu iPad ist verbesserwürdig. Da die Implementation von Workpad parallel zur Entwicklung des STEP Projektes erfolgte, wurde STEP primär mit TI8M.STEP.Mobile getestet. Da die Tests mit dem iPad selbst erst in der Endphase des Projektes durchgeführt werden konnten und die Ressourcen beim Workpad Team ebenfalls begrenzt sind, konnte keine zufriedenstellende Stabilität erreicht werden.

Ebenfalls Probleme verursacht hat die Konvertierung von PDF zu XPS Dokumenten. Abhängig davon, welche Metadaten innerhalb des PDF gesetzt sind und welche nicht, werden im XPS Dokument zum Teil fehlerhafte Tags gesetzt, welche auf dem Surface im schlimmsten Fall zum Absturz führen können. Dieser Fehler ist bei Microsoft bereits gemeldet worden und befindet sich immer noch im Review Stadium (Quelle: <http://connect.microsoft.com/VisualStudio/feedback/details/622563/exception-while-click-cannot-find-non-neutral-culture-related-to-und-scatterview-xps-in-documentviewer>).

## Mögliche Weiterentwicklungen

Die STEP Lösung beinhaltet viele Interessante Konzepte, welche durch den modularen Aufbau sehr viel Entwicklungspotenzial bieten.

Weiterentwicklung	Priorität	Aufwand
<b>Bearbeitung für Dokumente</b>	10	hoch
<b>Stabilitätsissues beheben</b>	10	mittel
<b>Windows 8 unterstützen</b>	9	hoch
<b>Surface Identity encrypted ablegen um Security zu erhöhen</b>	9	mittel
<b>iPhone unterstützen</b>	8	mittel
<b>Windows Phone unterstützen</b>	8	hoch
<b>Integration von Kinect</b>	7	hoch
<b>Surface zu Surface Interaktion</b>	7	hoch
<b>Andorid Phone unterstützen</b>	6	hoch
<b>Farbschemata in der Konfiguration anpassbar</b>	6	gering
<b>Rollen um rollenspezifische Funktionen erweitern</b>	5	mittel
<b>Darstellung von Bildern</b>	4	niedrig
<b>Darstellung von Movies</b>	4	mittel
<b>Zusätzliche Dokumenttypen hinzufügen</b>	2	niedrig
<b>Alternative Ticket Library hinzufügen(Microsoft Tags)</b>	1	niedrig

### 7.1 MARCO DANIELE

Schon während der Bewerbungsphase der Studienarbeit hatten wir im Team unterschiedlichste Ideen, was man alles machen könnte, doch sehr früh war bereits klar, dass wir von der Technologie her etwas im Microsoft Umfeld machen wollten. Nachdem wir in der Liste von Markus Stolze gesehen hatten, dass ein Projekt mit dem Surface 1.0 in Zusammenarbeit mit dem Technorama ausgeschrieben war, haben wir uns gleich darauf mit ihm in Verbindung gesetzt, um unsere Ideen zu präsentieren. Die erste Idee war, ein Spiel für auf dem Surface umzusetzen, welche mit der Begründung, dass das Technorama eher nicht an einem Spiel interessiert sein wird und dass man "soviel Spass erst im Master-Studium haben darf", abgelehnt.

Danach haben wir die Idee, eine Multi-Tasking Applikation für den Surface, vorgeschlagen, in welcher es zu Beginn einen simplen Task zu erfüllen gilt, aber mit der Zeit werden immer mehrere kleine Tasks parallel zu einander gestartet, welche es alle zu erfüllen gilt. Als wir Herrn Stolze mit der Idee überzeugt haben, hat sich herausgestellt, dass das Angebot vom Technorama nicht mehr aktuell war und der Surface Tisch daher auch nicht zu Verfügung stand und so hiess es wieder "go back to square one".

Parallel zu unseren Studienarbeitsbemühungen, hatten ich und Michael in einem Team-Meeting bei unserem Arbeitgeber, der ti&m, erfahren, dass man über die Möglichkeit nachdenkt, sich einen neuen Surface 2.0 anzuschaffen, falls Projekt-Ideen vorhanden wären. Wir haben dann unsere Idee einer Meeting Management Plattform vorgestellt. Von Seiten der ti&m wurde die Anforderung gestellt, dass als Endgeräte primär auf ein iPad hingearbeitet werden soll.

Diese Idee wurde dann von Herrn Stolze akzeptiert, ausschlaggebend war nicht zuletzt, dass ein Businesspartner im Hintergrund stand.

Während der Vorstudie stellten sich die technischen Mängel des Surface 2.0 heraus und viele unserer Ideen mussten bereits in einem frühen Stadium begraben werden, was sich in unserem Descoping niederschlägt.

In einem der monatlichen Teammeetings bei der ti&m wurde die Workpad Applikation vorgestellt, welche genau unserem Bedürfnis eines iPad Clients entspricht. Daher haben wir uns mit der Workpad Entwicklerin in Verbindung gesetzt und unsere Idee vorgeschlagen. Daraufhin wurde sehr viel Zeit aufgewendet um diese Schnittstelle sobald wie möglich bereit zu stellen.

Während der Studienarbeit mussten wir feststellen, dass es nicht nur Vorteile hat, wenn man beim Arbeitgeber die Studienarbeit absolvieren kann. Die meisten Mitarbeiter und vorallem Projektleiter hatten kein Verständnis dafür, dass man zwar vier Tage im Büro ist, aber nur zwei davon wirklich für die Arbeit aufwenden kann und die anderen zwei eigentlich für die Studienarbeit reserviert sind. Trotz mehrmaligem darauf hinweisen, dass man erst in zwei Tagen wieder Zeit für diese Offerte oder jenes Projekt zur Verfügung hat. Dennoch kam es häufig vor, dass man trotzdem die fünf Minuten für den Arbeitgeber gefunden hat, welche dann meistens doch eine halbe bis zwei Stunden in Anspruch nahmen. Daher haben wir für den Dokumentationsteil, in welchem wir nicht mehr zwingend auf den Surface angewiesen waren, die Räumlichkeiten gewechselt, so dass wir in Ruhe arbeiten konnten.

Rückblickend lässt sich sagen, dass die Arbeit mit einer neuen Technologie wie dem Surface sehr spannend war und bereits von Seiten der ti&m mehrer Offerten welche Surface Applikationen enthalten initiiert worden sind. Dadurch lässt sich sagen, dass die gelernten Lektionen aus der Semesterarbeit mich in meiner beruflichen Zukunft unterstützen werden. Ebenfalls hat die Arbeit am Projekt viel Spass gemacht.



## 7.2 MICHAEL MEIER

Das Team mit Marco Daniele und Silka Simmen war für mich eine wohlvertraute Umgebung. Marco kenne ich aus mehrjähriger Berufspraxis bei diversen Firmen sowie diversen Software-Engineering Projekten als Engineering Consultant. Silka kenne ich als Netzwerkspezialistin weniger von Software-Projekten, dafür aber aus dem privaten und IT-Betriebsumfeld.

Das Team ist aus meiner Sicht sehr schlagkräftig. Nicht nur darum, weil wir uns hervorragend ergänzen und uns unseren Stärken sowie Schwächen gegenseitig bewusst sind, sondern vorallem auch darum weil wir es vermögen die Motivation im Team zu erhalten.

So war die Formation des Teams schnell erledigt, die Auswahl des Themas für die Semesterarbeit hingegen gestaltete sich etwas umständlicher. Die grosse Chance in einem Projekt für das berufsbegleitende Studium sehe ich vorallem darin, dass diese Projekte wesentlich kleiner, einen enger eingegrenzten Scope haben und kein direkter, finanzieller Impact davon ausgeht. So suchten wir für die Realisierung der Semesterarbeit ein Projekt, welches eine Prise technisches Risiko, gepaart mit neuartigen Anwendungstechnologien bietet und natürlich auch Spass macht. Technologisch haben wir das im dazumals erst angekündigten Microsoft Surface 2 Tisch gefunden. Nach den ersten Gesprächen mit dem Technorama Winterthur, welches ein solches Projekt ausgeschrieben hatte, stellte sich heraus, dass die Durchführung aus terminlichen Gründen nicht klappt. Mit dem Versuch gescheitert, ein eigen-lanciertes Projekt umzusetzen, stellten wir rasch fest, dass nur Projekten mit Business Partner Türen für die Umsetzung offen stehen. So machten wir das Naheliegenste und lancierten ein Projekt mit der Wunschtechnologie und einem adequate Business-Nutzen zusammen mit dem Arbeitgeber von Marco und mir.

Mit der Projektabwicklung war ich sehr zufrieden. Trotz der hohen Auslastung auf privatwirtschaftlicher Seite konnten wir unsere Ziele erreichen.

Natürlich war es für unseren Arbeitgeber zusätzlich erschwerend zwischen für die Semesterarbeit reservierten und der regulären Arbeitszeit zu unterscheiden. So ist es oft vorgekommen dass ich kurzfristig für Kick-Offs und Beratungsmandate in andere Projekte parallel abgezogen wurde. Rückblickend würde ich deshalb auf eine Arbeit in diesem Kontext verzichten.

Etwas schade war auch, dass die am Anfang zugesicherte Unterstützung bzw. Ressourcen durch das Workpad Team eher knapp und zeitlich verspätet Einzug im Projekt gefunden hat. Dies hat uns dazu gezwungen umständlichere Umgehungs-lösungen zu implementieren. Das Package T18M.STEP.Mobile ist vollständig und ausschliesslich aufgrund dieses Umstandes entstanden. Da dieser Part eigentlich nicht zur offiziellen Aufgabenstellung gehört und auch keine Kernfunktionen im eigentlichen Application Lifecycle wahrnimmt haben wir uns bei der Umsetzung auch gewisse qualitative Freiheiten gelassen.

Die Zusammenarbeit mit der Fachhochschule bzw. den direkten Betreuern habe ich als sehr erfrischend und unkompliziert empfunden. Sie haben uns in diesem eher kreativen und neuartigen Projekt optimal unterstützt. Auch in positiver Erinnerung bleibt der flexible Charakter der Arbeit, da wir erst nach einer detaillierten Vorstudie innerhalb der Semesterarbeit abschätzen konnten, was technologisch nun tatsächlich mit dem neuen Surface 2 Tisch möglich ist. Ein teilweises Descoping und Ausweichen auf andere Lösungsstrategien ermöglichte es uns nicht nur die definierten Ziele zu erreichen, sondern auch gleich eine gewisse Grundarbeit an der Plattformlösung STEP zu verrichten. Diese erlaubt die Weiterentwicklung sowie Nutzung für verwandte Anwendungsszenarien auf flexible Art und Weise.

Es hat Spass gemacht.

### 7.3 SILKA SIMMEN

Gegen Ende des siebten Semesters haben wir im Team uns erste Gedanken zur Themenwahl der bevorstehenden Studienarbeit gemacht. Das Team hat sich schnell gefunden, da wir zu dritt bereits während der gesamten Studienzeit immer wieder in Teamarbeiten gut zusammen gearbeitet haben. Das Team ergänzt sich aus meiner Sicht sehr gut, da die einzelnen Personen verschiedene Kompetenzen und Stärken mitbringen, die gut miteinander harmonieren.

Bei der Suche nach einem geeigneten und spannenden Thema sind wir auf eine interessante Projektidee von Herrn Stolze gestossen: Die Realisierung einer Besucherapplikation auf einem Microsoft Surface Tisch für das Technorama Winterthur. Umgehend haben wir uns für diese Idee beworben, leider stellte sich bald heraus, dass das Technorama zum damaligen Zeitpunkt keine Ressourcen zur Verfügung hatte um ein solches Projekt zu betreuen.

Somit war eine interessante Idee vorhanden, jedoch fehlte ein Sponsor für einen Surface Tisch. Meine beiden Teamkollegen arbeiten bei einer Firma, die Partner von Microsoft Schweiz ist. Dort angefragt war schnell klar, dass grosses Interesse bezüglich einer Zusammenarbeit besteht, um ein Projekt mit einem Surface umzusetzen. Schnell war die Idee bezüglich der Integration mobiler Endgeräte mit einem Surface 2.0 geboren. Nach der definitiven Zusage der ti&m AG haben wir uns wieder bei Herrn Stolze gemeldet, der in der Zwischenzeit jedoch ausgelastet war bezüglich Betreuung von Arbeiten für das kommende Semester.

Zum Glück hatte Herr Bläser noch Kapazität für eine Arbeit und war einverstanden mit der Idee und der Zusammenarbeit mit dem Business Partner. Für mich persönlich war dies ein glücklicher Umstand, da ich somit Einblick in eine andere Firma und deren Arbeitsmethoden erhalte.

Respekt hatte ich zu Beginn vor der Technologie, da ich mit .NET erst wenige Projekte umgesetzt habe und somit wenig Erfahrung mitbringe. Da meine Teamkollegen aber erfahrene .NET Entwickler sind, war ich überzeugt, dass dieser Umstand für mich eher eine Chance als ein Risiko darstellt.

Zu Beginn der Arbeit folgte sogleich die erste Ernüchterung: Im Rahmen der Vorstudie hat sich herausgestellt, dass der Surface 2.0 einige technische Schwachstellen aufweist und unsere Ideen nicht so umgesetzt werden können, wie wir uns dies ursprünglich vorgestellt hatten.

Das war zwar schade, aber wir haben schnell alternative Konzepte ausgedacht, wie mit den gegebenen Möglichkeiten eine Integration eines iPad mit einem Surface erreicht werden kann.

Der weitere Verlauf des Projektes war aus meiner Sicht geradlinig, einzig die Koordination mit dem Workpad Team nahm mehr Zeit in Anspruch als wir ursprünglich geplant hatten. Eine weitere Schwachstelle sehe ich beim Architekturentscheid, das Backend vom Surface zu trennen, dieser war meiner Meinung nach und rückblickend betrachtet nicht ganz optimal. Die zusätzliche Komplexität kann im implementierten Anwendungsfall nicht ganz durch den Vorteil der Modularität aufgewogen werden.

Mit dem Endresultat bin ich trotzdem sehr zufrieden. Meiner Meinung nach hat STEP eine schöne und solide Architektur und bietet eine gute Basis für allfällige Weiterentwicklungen. Wir konnten zeigen, dass eine Integration von mobilen Geräten mit dem Surface gut möglich ist. Es hat Spass gemacht dieses Konzept umzusetzen und dass es am Schluss funktioniert hat ist sehr motivierend. Die Studienarbeit war rückblickend sehr spannend, herausfordernd und durch eine gute Zusammenarbeit im Team geprägt. Die umfangreiche Vorstudie ist aus meiner Sicht besonders gut gelungen. Mich freut, dass die ti&m AG zufrieden ist und das Projekt weiterverfolgt wird.

### 8.1 INSTALLATIONSANLEITUNG ENTWICKLUNGSUMGEBUNG

Im Rahmen des Projekts STEP wird nur Source Code abgegeben und kein Installer. Deshalb ist die hier beschriebene Installationsanleitung für Entwickler gedacht, welche sich mit .NET, C# und SQL auskennen. Zurzeit gibt es noch kein MSI Package und somit müssen die einzelnen Installationsschritte noch manuell vorgenommen werden.

#### Anforderungen:

- Microsoft Visual Studio® 2010
- .NET Framework 4.0
- Surface SDK 2.0
- Microsoft XNA® Framework Redistributable 4.0
- MS SQL 2008 (vorzugsweise mit Default Instanz)

#### Backend Solution aufsetzen:

1. Als erstes muss die DB aufgesetzt werden. Dafür gibt es im Backend.Database ein Script (install.cmd) welches die Erstellung der Datenbank übernimmt.
2. Auf dem Backend Server muss im IIS Manager der Ordner Backend.Service als Applikation auf localhost/STEP konfiguriert werden.
3. Ebenfalls im IIS Manager des Backend Servers soll ein Application Pool erstellt werden, welcher über die Credentials verfügen sollte, welche Zugriff auf Backend.Service und auf die erstellt Datenbank hat.
4. Die Backend Solution soll nun als Administrator geöffnet werden. Sollte bei der Datenbank Erstellung nicht die Default Instanz verwendet worden sein, muss im web.config der Connection String dementsprechend angepasst werden.
5. Solution builden.

#### Surface Solution:

1. Konfiguration anpassen, dass als Endpoint die Adresse des Backend Servers verwendet wird.
2. Mit Hilfe von Pre Build Commands wird die Config File an die benötigten Stellen kopiert.

### 8.2 BENUTZERHANDBUCH

#### Charms Bar

Die Charms Bar kann von jeder Seite des Tisches hinein gewischt werden. Auf der Charms Bar können die grundlegenden Funktionen wie User Liste darstellen und neuen User anmelden ausgeführt werden.



Charms Bar

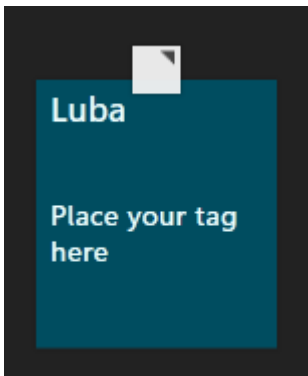
## User Anmelden

Wenn man sich mit dem iPad an dem Surface Tisch anmelden will, kann man über die Charms Bar mit Hilfe des Buttons "Add" ein One-Time Ticket anfordern. Dieses Ticket wird als QR-Code auf dem Surface dargestellt und muss von der Workpad Applikation gelesen werden um sich am Surface anzumelden.

Das One-Time Ticket hat eine beschränkte Gültigkeit, die Dauer die es noch gültig ist wird einerseits in Sekunden dargestellt, als auch grafisch als Balken der kleiner wird.



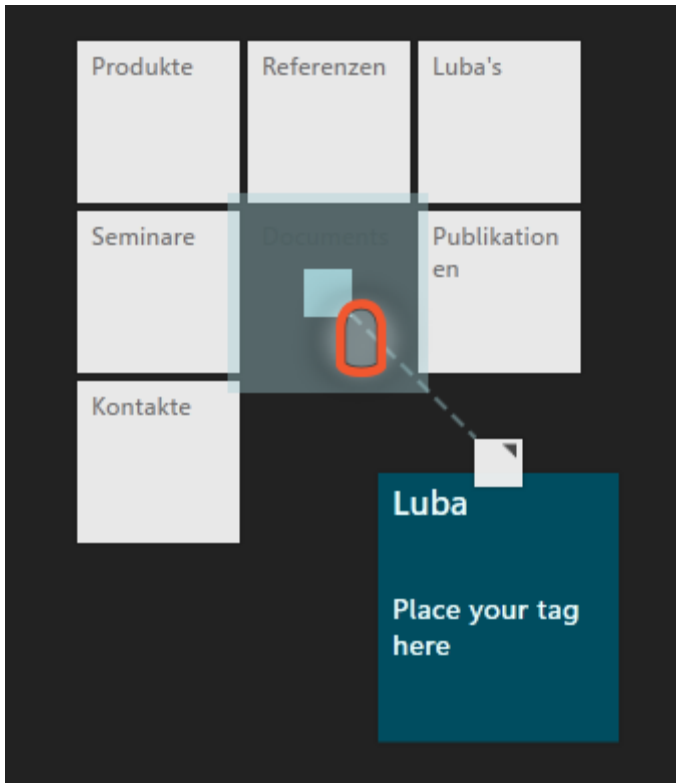
Ablaufendes One-Time Ticket



Benutzer hat sich erfolgreich angemeldet

## Dokument Kategorien

Auf dem User Menu kann man die selben Dokument Kategorien wie auf dem Workpad einsehen. Sobald man sich für eine Kategorie entschieden hat, werden die einzelnen Dokumente dieser Kategorie dargestellt und können geöffnet werden.



Übersicht über alle Kategorien



Eine Kategorie wurde ausgewählt

## Dokumentanzeige

Sobald ein Dokument aus der Dokumentenkategorie gezogen wird und irgendwo auf dem Surface gedroppt wird, wird das entsprechende Dokument auf dem Surface geöffnet.



Offenes Dokument

Falls mehrere Personen um den Tisch stehen und nicht alle optimal das Dokument einsehen können, kann von jedem Dokument eine Kopie erstellt werden. Jeder Benutzer kann die Kopie für sich so positionieren, damit er eine einwandfreie Aussicht darauf genießen kann.



Kopie-Befehl wird gesendet



Kopie wurde erstellt

Damit das Meeting einem geordnetem Verlauf folgen kann und nicht jeder Benutzer beliebig in seiner Kopie herum scrollt, gibt es die Möglichkeit, dass alle Kopien eines Dokuments sich an ihr

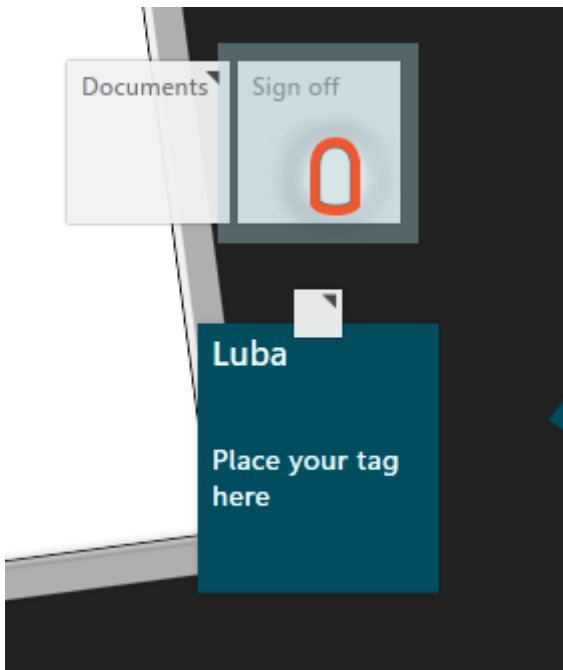
Ursprungsdokument binden. So werden jegliche Scroll-Bewegungen welche im Ursprungsdokument getätigt werden in allen Kopien durchgeführt. Es wird somit gewährleistet, dass alle Meetingteilnehmer sich stets an der selben Stelle des Dokuments befinden und über das selbe diskutieren.



Verlinkungsbefehl vom Ursprungsdokument an seine Kopien

### Abmelden

Abmelden kann man ein Gerät via Workpad Applikation oder über das User Menü auf dem Surface direkt.



Abmeldebefehl



## 9.1 GLOSSAR

### **ADO.net**

"ADO.NET stellt die direkteste Methode des Datenzugriffs innerhalb von .NET Framework bereit."

Website: <http://msdn.microsoft.com/de-de/library/e80y5yhx.aspx>

### **.NET Framework**

.NET bezeichnet eine von Microsoft entwickelte Laufzeitumgebung für Computerprogramme. Sie umfasst eine Sammlung von Klassenbibliotheken, eine API (Programmierschnittstelle) und etliche Dienstprogramme (Services). .NET ist im vollen Umfang nur für Windows verfügbar, aber viele Programme laufen z. B. mit Hilfe des Mono-Projektes oder dotGNU auch auf Unix basierten Betriebssystemen.

### **Gantt-Diagramm**

Ein Diagramm-Typ, welches den Projekt-Zeitplan aufzeigt. Es ist ersichtlich, wann welche Phasen und Aktivitäten im Projekt vorgesehen sind, dazu gehört die Visualisierung des Start- und Abschlusszeitpunktes.

Wikipedia: [Gantt-Chart](#)

### **Microsoft.NET**

Bezeichnet eine von Microsoft entwickelte Laufzeitumgebung für Computerprogramme.

Website: <http://msdn.microsoft.com/en-gb/netframework/default.aspx>

Wikipedia: [Microsoft .NET](#)

### **Multitouch**

Fähigkeit, gleichzeitig mehrere Berührungen zu erkennen. Beispielsweise bei Windows 7 oder iPhone.

Wikipedia: [Multitouch](#)

### **MSDN**

Microsoft Developer Network

### **SDK**

Ein Software Development Kit ist eine Sammlung von Werkzeugen und Anwendungen, um eine Software zu erstellen.

Wikipedia: [Software Development Kit](#)

### **WCF**

WCF steht für Windows Communication Foundation.

Website: [http://msdn.microsoft.com/en-us/library/ms735119\(v=VS.90\).aspx](http://msdn.microsoft.com/en-us/library/ms735119(v=VS.90).aspx)

## **WPF**

WPF ist ein Grafik-Framework, welches mit dem [.Net Framework 3.0](#) eingeführt wurde. Es stellt ein umfangreiches Modell bereit, bei welchem die Präsentations- und Geschäftslogik getrennt wird. WPF-Anwendungen können sowohl Desktop- als auch Web-Anwendungen sein und benutzen, wenn möglich, auch Hardwarebeschleunigung. Das Framework versucht, die verschiedenen Bereiche, die für die Präsentation wichtig sind (Benutzerschnittstelle, Zeichnen und Grafiken, Audio und Video, Dokumente, Typographie), zu vereinen.

Website: <http://msdn.microsoft.com/en-us/library/ms754130.aspx>

Wikipedia: [Windows Presentation Foundation](#)

## **XAML**

XAML ist eine neue, von Microsoft entwickelte allgemeine Beschreibungssprache für die Oberflächengestaltung von Anwendungen.

Die Sprache ist Case-sensitive, das heisst es wird zwischen Gross- und Kleinschreibung unterschieden. XAML beschreibt zudem die Anwendungsoberflächen für [WPF](#) deklarativ. Dadurch wird versucht das Aussehen von der Logik zu trennen.

Website : <http://msdn.microsoft.com/en-us/library/ms752059.aspx>

Wikipedia: [Extensible Application Markup Language](#)

## **XML**

Auszeichnungssprache zur Darstellung hierarchisch strukturierter Datensätze in Form von Textdaten.

Wikipedia: [Extensible Markup Language](#)

## **XNA**

Steht für Xna is Not Acronymed. Kapselt die Microsoft eigne Umgebung, Technologie, SDKs sowie IDE für die Spieleentwicklung.

Website: <http://msdn.microsoft.com/en-us/aa937791.aspx>

## **XPS**

XPS steht für XML Paper Specification und ist ein Dateiformat für Dokumente von Microsoft entwickelt.

Website: <http://msdn.microsoft.com/en-us/windows/hardware/gg463431.aspx>

## 9.2 LITERATURVERZEICHNIS

QR-Codes

<http://www.denso-wave.com/qrcode/index-e.html>

Surface 2.0 SDK – Architecture

<http://msdn.microsoft.com/en-us/library/ff727809.aspx>

Surface 2.0 SDK - Tagged Object Recognition

<http://msdn.microsoft.com/en-us/library/ff727854.aspx>

Surface 2.0 SDK - Tagged Objects and Tag Visualizations

<http://msdn.microsoft.com/en-us/library/ff727801.aspx>

WPF window: displaying an embedded PDF

<http://www.screencast.com/t/JXRhGvzvB>

Bug-Report: Exception while click: "Cannot find non-neutral culture related to 'und'." ( ScatterView + XPS in DocumentViewer)

<http://connect.microsoft.com/VisualStudio/feedback/details/622563/exception-while-click-cannot-find-non-neutral-culture-related-to-und-scatterview-xps-in-documentviewer>

# Aufgabenstellung Studienarbeit für Marco Daniele, Michael Meier und Silka Simmen:

## Meeting Management mit Microsoft Surface 2

### 1. Auftraggeber und Betreuer

Diese Studienarbeit findet in Zusammenarbeit mit der Firma *Technology Management & Innovation AG, Zürich* statt.

#### **Ansprechpartner Auftraggeber:**

- Stefan Pfenninger, Principal, [Stefan.Pfenninger@bsgroup.ch](mailto:Stefan.Pfenninger@bsgroup.ch)

#### **Betreuer HSR:**

- Prof. Dr. Luc Bläser, Institut für Software, [lblaeser@hsr.ch](mailto:lblaeser@hsr.ch)

### 2. Ausgangslage

Mit Hilfe des interaktiven und digitalen Microsoft Surface 2 Tisches sollen Business Meetings in Bezug auf gemeinsame Bearbeitung von Dokumenten effizient unterstützt werden. Ein Meeting-Teilnehmer soll seine Dokumente auf dem Tisch eines Sitzungsraums darstellen können, indem er zuerst sein Mobiltelefon mit dem entsprechen geöffneten Dokument auf den Tisch legt. Der Tisch hat im Backend Zugriff auf ein Document Management System wie zum Beispiel MS Sharepoint. Die Dokumente können dann gemeinsam am Tisch gelesen, bearbeitet und mit Notizen vermerkt werden. Am Schluss werden die Änderungen an den Dokumenten wieder zurück im Document Management System zurückgespeichert.

Um diesen Ablauf zu realisieren, bedarf es der Erarbeitung von Lösungen in verschiedenen Aspekten:

- 1) *Anzeigen von Dokumente vom Mobiltelefon am Tisch:* Konzeption und Umsetzung eines geeignetes Vorgehen, um Dokumente vom Mobilgerät am Tisch öffnen zu können, z.B. durch direkte Übertragung mit Bluetooth, durch optische Erkennung des Mobiltelefons und anschliessenden Instruktionen zum Upload an ein Document Management System des Tisches bzw. andere Ansätze.
- 2) *Meeting Session Programm:* Entwicklung einer Anwendung auf dem Surface Table, welche die Durchführung einer Sitzung mit Dokumenten leitet. Dies umfasst das Starten einer Sitzung, das Uploaden/Auswählen von Dokumenten, Darstellung der Dokumente in geeigneten Viewers oder Editors, das Abspeichern der Dokumente und schliesslich das Beenden der Sitzung.

- 3) *Office und PDF Viewer am Tisch*: Einsatz von Standard-Anwendungen am Tisch, um die Dokumente anzeigen und bearbeiten zu können sowie die Integration dieser in das Meetings Session Programm.
- 4) *Document Management System*: Verwaltung und Zugriff von Dokumenten in einem bestimmten Document Management System im Backend, welcher unter anderem vom Tisch zugegriffen wird.

## 1. Ziele und Aufgabenstellung

Die Aufgabe dieser Arbeit ist es, eine Lösung für das Meeting Management am Microsoft Surface 2 Tisch zu entwickeln.

Folgende spezifische Ziele werden vorgegeben:

- Aufnahme der Anforderungen für das Meeting Management basierend auf dem Surface 2 Tisch.
- Entwurf eines Lösungskonzeptes zur Umsetzung des Meeting Ablaufs mit Hilfe von dem Surface 2 Tisch (Abläufe, technische Analyse, erster Prototyp)
- Entwicklung der Anwendung für den Surface 2 Tisch, welche den Meeting Ablauf umsetzt. Dies umfasst insbesondere das Öffnen der Dokumente am Tisch mit dem Mobiltelefon, das Darstellen und Bearbeiten der Dokumente am Tisch sowie das Abspeichern am Schluss.
- Integration aller nötigen Komponenten (Mobile-Surface Interaktion, Meeting Session Programm, Document Management System) zur funktionsfähigen Gesamtlösung.

## 2. Zur Durchführung

Mit dem HSR-Betreuer finden in der Regel zweiwöchentliche Besprechungen statt. Zusätzliche Besprechungen sind nach Bedarf durch die Studierenden zu veranlassen. Besprechungen mit dem Auftraggeber werden nach Bedarf durchgeführt.

Alle Besprechungen sind von den Studenten mit einer Traktandenliste vorzubereiten und die Ergebnisse in einem Protokoll zu dokumentieren, das dem Betreuer und dem Auftraggeber per E-Mail zugestellt wird.

Für die Durchführung der Arbeit ist ein Projektplan zu erstellen. Dabei ist auf einen kontinuierlichen und sichtbaren Arbeitsfortschritt zu achten. An Meilensteinen gemäss Projektplan sind einzelne Arbeitsergebnisse in vorläufigen Versionen abzugeben. Über die abgegebenen Arbeitsergebnisse erhalten die Studierenden ein vorläufiges Feedback. Eine definitive Beurteilung erfolgt auf Grund der am Abgabetermin abgelieferten Dokumentation.

### 3. Dokumentation

Über diese Arbeit ist eine Dokumentation gemäss den Richtlinien der Abteilung Informatik zu verfassen (siehe <https://www.hsr.ch/Allgemeine-Infos-Diplom-Bach.4418.0.html?&L=0>). Die zu erstellenden Dokumente sind im Projektplan festzuhalten. Alle Dokumente sind nachzuführen, d.h. sie sollten den Stand der Arbeit bei der Abgabe in konsistenter Form dokumentieren. Die Dokumentation ist vollständig auf CD/DVD in 3 Exemplaren abzugeben. Auf Wunsch ist für den Auftraggeber eine gedruckte Version zu erstellen.

### 4. Termine

Siehe auch Terminplan auf <https://www.hsr.ch/Termine-Diplom-Bachelor-und.5142.0.html?&L=0>

- 20.02.12 Beginn der Studienarbeit, Ausgabe der Aufgabenstellung durch die Betreuer.
- 29.05.12 Die Studierenden senden folgende Dokumente der Arbeit per Email zur Prüfung an ihre Betreuer.
  - Kurzfassung
  - A0-Poster
 Vorlagen sowie eine ausführliche Anleitung betreffend Dokumentation stehen unter den allgemeinen Infos Diplom-, Bachelor- und Studienarbeiten zur Verfügung.
- 01.06.12 Die Studierenden senden die vom Betreuer abgenommene und freigegebene Kurzfassung als Word-Dokument an das Abteilungssekretariat (cfurrer(at)hsr.ch)
- 01.06.12 Abgabe des Berichtes an den Betreuer bis 17.00 Uhr

### 5. Beurteilung

Eine erfolgreiche Studienarbeit erhält 8 ECTS-Punkten (1 ECTS Punkt entspricht einer Arbeitsleistung von ca. 25 bis 30 Stunden). Für die Modulbeschreibung der Studienarbeit siehe [https://unterricht.hsr.ch/staticWeb/allModules/19456\\_M\\_SAI.html](https://unterricht.hsr.ch/staticWeb/allModules/19456_M_SAI.html)

Gesichtspunkt	Gewicht
<b>1. Organisation, Durchführung</b>	1/5
<b>2. Berichte (Abstract, Mgmt Summary, techn. u. persönliche Berichte) sowie Gliederung, Darstellung, Sprache der gesamten Dokumentation</b>	1/5
<b>3. Inhalt *)</b>	3/5

\*) Die Unterteilung und Gewichtung von 3. Inhalt wird im Laufe dieser Arbeit festgelegt.

Im Übrigen gelten die Bestimmungen der Abt. Informatik zur Durchführung von Studienarbeiten.

Rapperswil, den 11. Januar 2012

---

Der verantwortliche Dozent



Prof. Dr. Luc Bläser  
Institut für Software  
Hochschule für Technik Rapperswil



## Vereinbarung

### 1. Gegenstand der Vereinbarung

Mit dieser Vereinbarung werden die Rechte über die Verwendung und die Weiterentwicklung der Ergebnisse der Studienarbeit *Meeting Management mit Microsoft Surface 2* von *Marco Daniele, Michael Meier und Silka Simmen* unter der Betreuung von *Prof. Dr. Luc Bläser* geregelt.

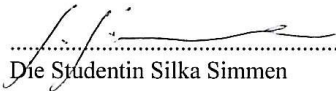
### 2. Urheberrecht

Die Urheberrechte stehen der genannten Studentin und den genannten Studenten zu.

### 3. Verwendung

Die Ergebnisse der Arbeit dürfen sowohl von der genannten Studentin und den genannten Studenten, von der HSR wie von TI&M nach Abschluss der Arbeit verwendet und weiter entwickelt werden

Rapperswil, den 21.12......

  
.....  
Die Studentin Silka Simmen

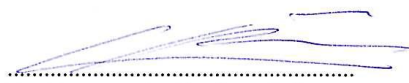
Rapperswil, den 21.2.12.....

  
.....  
Der Student Michael Meier

Rapperswil, den 21.2.12.....

  
.....  
Der Student Marco Daniele

Rapperswil, den 21.2.12.....

  
.....  
Der Betreuer der Studienarbeit  
Prof. Dr. Luc Bläser

Rapperswil, den.....

.....  
Der Studiengangleiter / die Studiengangleiterin




## 9.5 ERKLÄRUNG ÜBER DIE EIGENSTÄNDIGE ARBEIT



### Erklärung über die eigenständige Arbeit

Hiermit erkläre ich die vorliegende Arbeit „Meeting Management mit Microsoft Surface 2“, sofern nicht speziell deklariert, selber und ohne fremde Hilfe durchgeführt zu haben.

  
\_\_\_\_\_  
Marco Daniele, 31.5.2012

  
\_\_\_\_\_  
Michael Meier, 31.5.2012

  
\_\_\_\_\_  
Silka Simmen, 31.5.2012