



MC ↔ **astor2.0**

Sitzungsprotokolle

Team 4

V 1.2

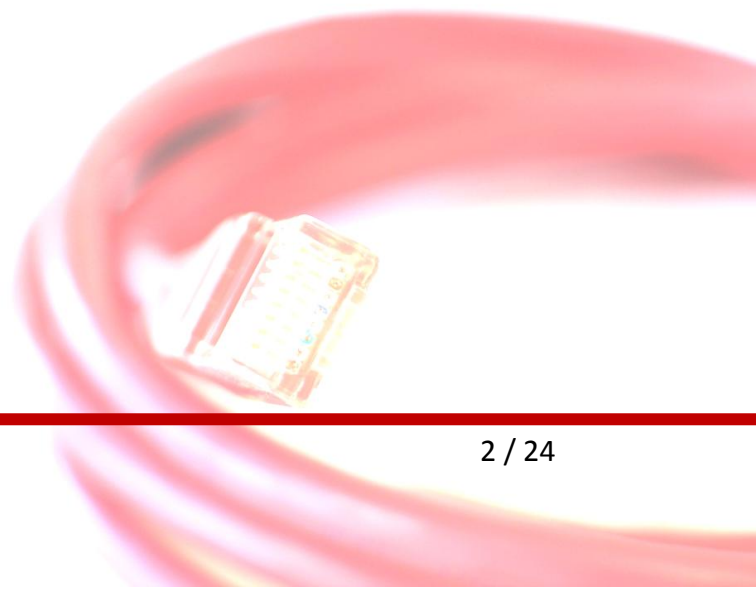
Projekt: MultiCastor 2.0

Auftraggeber: Rentschler & Stuckert
Rotebühlplatz 41/1
70178 Stuttgart

Auftragnehmer: TIT10AID - Team 4 - MCastor2.0

Fabian Fäßler
Filip Haase
Matthis Hauschild
Sebastian Koralewski
Jonas Traub
Christopher Westphal

Rotebühlplatz 41 – Raum 0.10
70178 Stuttgart

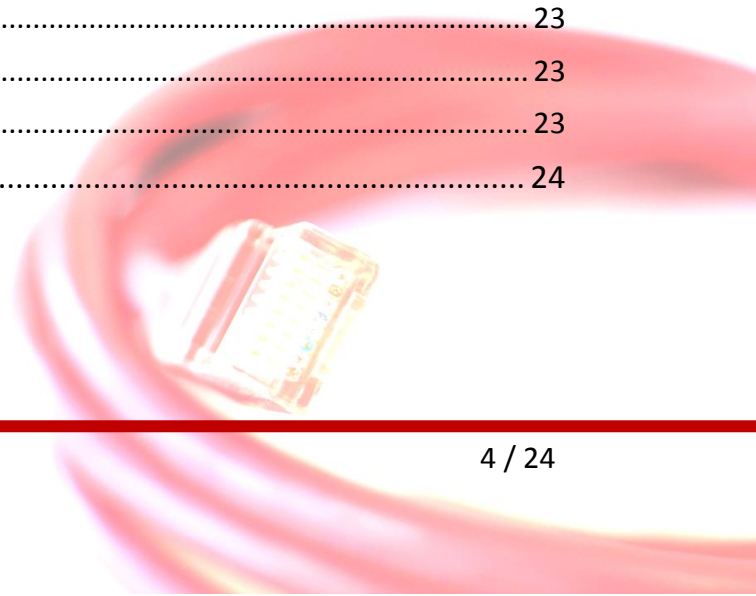


Inhaltsverzeichnis

16.09.2011 - Kickoff	4
Agenda	5
Projektleiter	5
Grobes Umreisen des Vorgehens und Aufgaben	5
Einigung auf gemeinsame Kommunikationswege	5
Einigung auf gemeinsame Tools	5
Meeting Termine	6
20.09.2011 – Kundengespräch	7
23.09.2011 – 1. Teammeeting	8
Agenda	8
Organisatorisches	8
Aufteilung des Teams	8
GIT	8
Vorlagen(Aufgabenanalyse und Verteilung)	8
Quellcode	8
Aufteilung der CRS	8
Sonstiges	9
29.09.2011 – 2. Teammeeting	10
Agenda	10
Logo/Layouts	10
CRS	10
GIT	10
Organisatorisches	11
04.10.2011 – 3. Teammeeting	12
Agenda	12
CRS Besprechung	12
BC Besprechung	12
Organisatorisches	12
07.10.2011 – 4. Teammeeting(Kurzmeeting zur Aufgabenverteilung)	13
11.10.2001 – 5. Teammeeting	13
13.10.2011 – 6. Teammeeting (Git Einrichtung)	13
Agenda	13
SRS/SAS	13

Git	13
18.10.2011 – 7. Teammeeting (Vertiefung SAS/SRS)	14
Agenda	14
1. Protokollvergabe	14
2. Google Kalender	14
3. SRS/SAS.....	14
GDocs:	14
SRS - Allgemein	15
SRS – Zu erstellende Grafiken	15
SRS – Fragen an die Dozenten	15
SAS – Allgemein.....	15
SAS – Fragen an die Dozenten	16
Aufgabenverteilung im SRS/SAS	16
4. Termine	16
5. Codeanalyse	16
6. PIZZA!	16
7. Anderes	17
8. Anlagen.....	17
20.10.2011 – Mitschrift der Unterschiede SRS/SAS	20
27.10.2011 – 8. Teammeeting (Besprechung des Fortschritts)	21
Agenda	21
Protokollvergabe:	21
Berichte – Aktueller Stand	21
Zeitplan	22
Aufgabenverteilung.....	22
SAS.....	22
SRS.....	23
Dokumentation	23
Glossar.....	23
GUI V1.5	23
Dokumentversionen	24

16.09.2011 - Kickoff



Agenda

- Wahl des Projektleiters
- Grobes Umreisen des Vorgehens und der anstehenden Aufgaben
- Einigung auf gemeinsame Kommunikationswege
- Einigung auf gemeinsame Tools
- Meeting Termine

Projektleiter

Zum Projektleiter wurde einstimmig Jonas Traub bestimmt.

Grobes Umreisen des Vorgehens und Aufgaben

Zu den wichtigsten Aufgaben gehörte die Zuteilung der Rollen innerhalb des Teams. Es wurde sich auf 6 Rollen geeinigt:

- Projektleiter
- Leading Engineer
- Documentation
- Engineer/Tester, Expert on STAF/STAX
- Engineer/Tester, Expert on Usability
- Engineer/Tester, Expert on MMRP

Damit sich jeder dazu in der Lage ist sich genauer mit den noch offenen Rollen und den damit verbundenen Aufgaben auseinandersetzen, wurde beschlossen die Verteilung auf das nächste Meeting zu verschieben.

In diesem Zusammenhang wurde beschlossen dass sich jeder bis zum nächsten Meeting genau mit dem Programm beschäftigt haben muss (Herunterladen, Sender/Receiver und Loopback ausprobieren und Configfiles anschauen) und als „Nice To Have“ auch schon einen Blick in den Sourcecode werfen sollte.

Zum Kundengespräch am 20.09.2011 sollte jeder seine offenen Fragen sammeln und einreichen. Das Gespräch führt stellvertretend Jonas Traub.

Einigung auf gemeinsame Kommunikationswege

Zur einfachen gemeinsamen Kommunikation wurde eine Facebook Gruppe von Filip Haase eingerichtet, damit dort Unklarheiten zeitnah diskutiert werden können.

Es wurde beschlossen hier *vorerst* auch alle Dokumente (wie Protokolle, Sammlung der Bugs) zu speichern bis dies durch eine Software zur Versionsverwaltung ersetzt wird.

Für die Planung der Meetings wurde beschlossen dass Jonas Traub zu den Meetings samt Agenda über *Google-Calender* einlädt. Änderungswünsche in der Agenda und am Termin können dann sofort mitgeteilt und geändert werden.

Einigung auf gemeinsame Tools

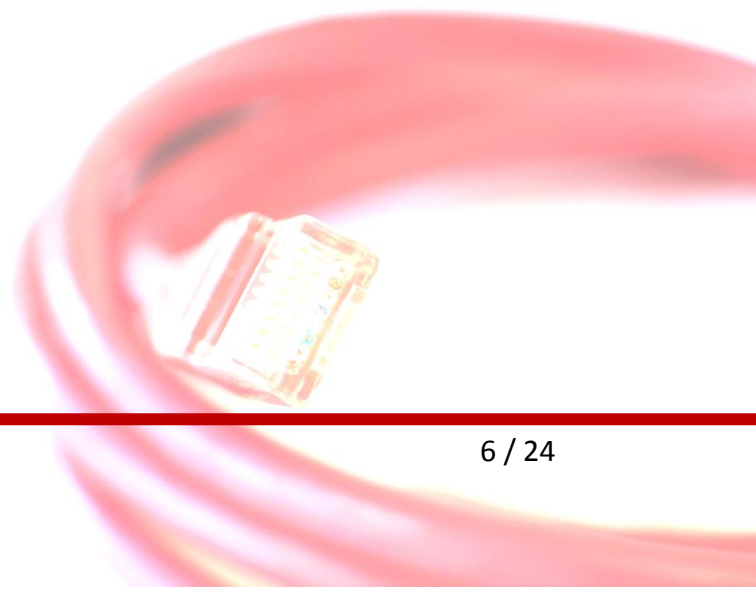
Als Integrierte Entwicklungsumgebung wurde sich auf Eclipse geeinigt, damit es nicht zu Import/Export Schwierigkeiten kommen kann.

Als nächster Punkt wurde über eine Versionsverwaltungssoftware diskutiert. Zur Wahl standen SVN und GIT. Es wurde sich auf GIT geeinigt da Fabian Fäßler bereits über Erfahrung mit GIT hat und sich dazu bereit erklärt hat einen Server einzurichten und eine Einführung in GIT für alle zu geben. Es wurde beschlossen dass der Server bis zum nächsten Termin eingerichtet sein soll.

Meeting Termine

Es wurde beschlossen vorerst keine regelmäßigen Meetings(z.B. Dienstag Abend 18.00 Uhr jede Woche) anzusetzen sondern die Termine dynamisch dem Workflow anzupassen.

Termin zum 1. Teammeeting: **23.09.2011**, ca **12.00** Uhr (nach FS-Vorlesung)



20.09.2011 – Kundengespräch

Die Fragen wurden von den Vertretern des Entwicklerteams gestellt. Die Antworten kommen vom Kunden (Hr. Stuckert und Hr. Rentschler).

Q: Wir haben alle Dokumente bekommen und haben Fragen. Usecases die dort beschrieben sind. Sind die neu oder alt? **A:** Lastenheft sind von einer Version 1 Multicastor CRS kopiert und eingefügt, also fehlen die neuen Anforderungen

Q: Was ist die Zielgruppe? **A:** Alle Personen die von Berufswegen etwas mit Netzwerken zu tun haben. Die Zielgruppe bleibt gleich wie bei der Version 1. Es soll aber trotzdem ein sehr ausführliches Manual für Neueinsteiger vorhanden sein.

Q: Die Usability (GUI) soll überarbeitet werden. Wie kann man sich das genau vorstellen? Gibt's schon bereits Konzepte? **A:** Ein Paar Punkte wurden bereits genannt, aber die genauen Usability Punkte sollen selbst definiert werden.

Q: Die Fenster sollen ausblendbar sein? **A:** Es geht um die Tabs. Die sollen ein- und ausblendbar sein. (Bsp: Fenster Menüpunkt mit Häkchen) [...] Es soll auch überlegt werden wie die Buttons neu angeordnet werden können. [...] Sender und Receiver Tabs sehen zu ähnlich aus. Außerdem sollen sie anhand ihrer Titel identifiziert werden können.

Q: Darf das Layout auch vollständig überarbeitet werden? (Z.B. Tabs wie im Firefox aus den Fenstern ziehen) **A:** Generell darf alles geändert werden was für sinnvoll erachtet wird. Allerdings würde es viel Zeit brauchen die für andere, wichtigere Aufgaben gebraucht wird.

Q: Sollen verschiedene Sprachen unterstützt werden? **A:** Am besten English und Deutsch.

Q: Soll etwas an den Config Files geändert werden? **A:** Am besten soll zwischen Programm und Testkonfiguration unterschieden werden, und diese sollen getrennt werden.

Q: Ist für das geforderte Regressionstest-System JUnit ausreichend? **A:** JUnit soll nur auf Codeebene verwendet werden. Wir wollen aber dass das Programm komplett getestet wird. [...] Dafür soll STAF/STAX benutzt werden. [...] CLI, LOG, CONF und ETH muss über STAF/STAX getestet werden.

Q: Soll der MulticastSocket von java.net verwendet werden? Oder pcap? **A:** Never touch a running system. [...] IGMP, MLD ist Layer 3. MMRP ist Layer 2. Für Layer 2 Multicaster muss die MAC Adresse und nicht die IP Adresse in die GUI.

Q: Soll es eine Oberfläche für IPv4, IPv6, Layer2 und 3 geben? **A:** Nein. Layer2 und Layer3 sollen getrennt werden. Weil es auch andere Config-Files gibt.

Q: Welche Java und Betriebssystemvoraussetzungen sollen beachtet werden? **A:** Java 1.6, BS: Linux und Windows (Mac optional)

Q: Was ist mit Zuverlässigkeit genau gemeint? **A:** Es muss erkennen wenn es an Ressourcengrenzen kommt. [...] Ein Netzwerkabriss muss abgefangen und geloggt werden. [...] Außerdem sollen sinnvolle „Defaultwerte“ in die Formulare eingefüllt werden.

Q: Wie sieht es mit dem Zeitplan und den Präsentationen aus? **A:** Das Projekt ist in Phasen eingeteilt. Bald müssen erste Anforderungen angegeben werden, genaue Termine stehen in den Zeitplänen. Dieses Semester gibt es 2 Präsentationen (einmal SRS und einmal CRS)

Q: Sollen wir Snake im Programm lassen? **A:** Völlig egal

23.09.2011 – 1. Teammeeting

Agenda

- Organisatorisches
- Aufteilung des Teams
- GIT
- Vorlagen(Aufgabenanalyse und Verteilung)
- Quellcode
- CRS/BC

Organisatorisches

- Das Meeting am Donnerstag wird zwischen dem Anlageseminar und der „WebEngineering“-Vorlesung gehalten.
- Nachfrage bezüglich des Abgabetermins für die Präsentation, was genau als Business Case definiert ist und ob die Gruppenverteilung in Ordnung ist
- Wichtige Termine zur Übersicht:
 - 4.10 Abgabe CRS
 - 3.11 Abgabe SRS und SAS

Aufteilung des Teams

- | | |
|---|----------------------|
| • Projektleiter: | Jonas Traub |
| • Dokumentationsverfasser: | Mathis Hauschild |
| • Leading Engineer: | Filip Haase |
| • Engineer/Tester, Expert on STAF/STAX: | Sebastian Koralewski |
| • Engineer/Tester, Expert on Usability: | Christopher Westphal |
| • Engineer/Tester, Expert on MMRP: | Fabian Fäßler |

GIT

Git wurde von Fabian Fäßler aufgesetzt und kann in Eclipse integriert werden. Da sich an der DHBW aber leider auf seinen Server „getunnelt“ werden muss, und man dazu noch einen SSH Tunnel für GIT braucht für den privaten Server, wird noch eine geschicktere Lösung wie **GIT auf DHBW Server** oder **GITHub** gesucht. Aufgrund des Zeitmangels wurde die GIT-Einführung auf das nächste Meeting verlegt.

Vorlagen(Aufgabenanalyse und Verteilung)

- Dokumentvorlagen für Präsentation und Dokumente(Jonas Traub)
- Logo für das Team (Fabian Fäßler)
- Inoffizieller Teamname: MCastor2.0

Quellcode

Jeder hat bereits in den Quellcode geschaut, eine gemeinsame Einarbeitung in den Quellcode wird allerdings auf einen unbestimmten Zeitpunkt nach Präsentation und Abgabe des CRS gelegt.

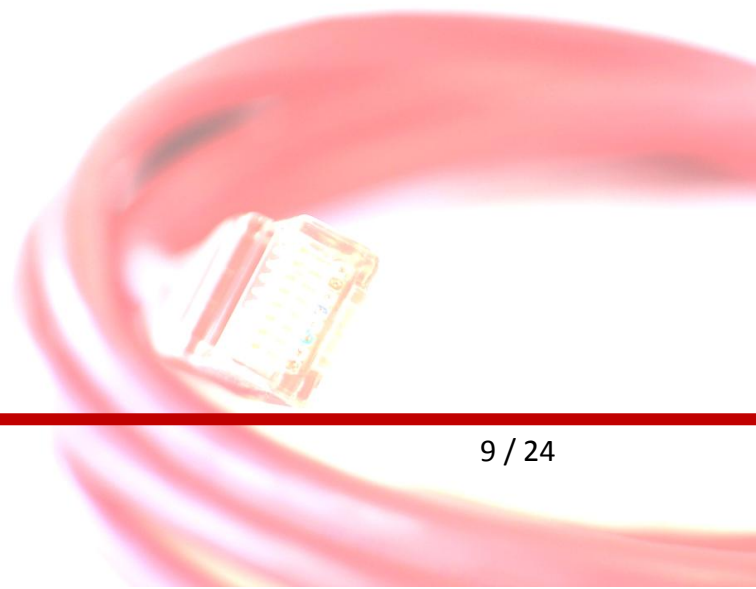
Aufteilung der CRS

- Produkteinsatz: Sebastian Koralewski

- Produktfunktion: Filip Haase
- Produktdaten: Mathis Hauschild
- Produktcharakteristik Christopher Westphal

Sonstiges

Gefundene Bugs sollen in einem Facebook Dokument abgelegt werden dass dafür angelegt wurde. Bis sie dann ins SRS eingetragen werden.



29.09.2011 – 2. Teammeeting

Agenda

- Logo/Layouts
- CRS
- Organisatorisches
- GIT-Einführung

Logo/Layouts

Fabian hat das Logo fertiggestellt und es wurde vom Team als gut befunden.

Jonas hat die Dokumentationsvorlagen fertiggestellt, das Logo von Fabian erhalten und eingebaut und es wurde vom Team als gut befunden.

CRS

Die CRS Aufgaben wurden aufgeteilt allerdings kamen allerdings einige Fragen auf. [Herr Stuckert kam zufälligerweise am Meeting vorbei, deswegen wurde er direkt nach einigen Dingen gefragt]

Q: Sind die UseCases bereits vollständig, oder sollen noch neue UseCases hinzugefügt werden(z.B. Staf/Stax) **A:** Ja, generell sind bisherige UseCases fast vollständig, können aber noch einzelne Punkte hinzugefügt werden.

Q: In welchem Grad müssen sie noch detaillierter geschrieben werden(da sie bis jetzt sehr allgemein gehalten sind)? **A:** Man kann es exakter machen, aber nicht zu exakt. Sollte allgemein gültig sein. Allerdings sollte Unterschied zwischen Layer 2 und Layer 3 Protokollen erwähnt werden. Requirements sollten noch genauer ausformuliert werden. SRS am Ende ganz genau.

Q: Wie soll die Hilfe aussehen? Soll es ein neues/extra Fenster geben oder Sie eingebettet werden? **A:** Vorerst sollte darauf geachtet werden dann nichts gemacht wird das unnötig Zeit verschwendet.

[Herr Stuckert verabschiedete sich wieder]

Aufgrund der ungeklärten Urheberrechten werden alle Grafiken neu erstellt. Um ein einheitliches Design zu schaffen wird dies von Fabian Fäßler übernommen.

GIT

Die Git-Einführung wurde von Fabian Fäßler gehalten.

- Git herunterladen und installieren: <http://git-scm.com/>
- EGit und JGit über Eclipse Update/Install standart installieren. URL:<http://download.eclipse.org/egit/updates>
- Anschließend Window-> Show View -> Git/Git Repositories
- > Git Repository Tab sollte hinzugefügt sein
- Clone Git Repository sollte angezeigt werden
- Bei URI einfügen: it##DHBW Benutzername##@login.dhbw-stuttgart.de:/home/blackhole.it02/it10025/repo/MultiCastor
- Connection: Protocol ssh
- Willkommensnachricht wegklicken ("unerlaubter Systemzugriff...")
- Lokalen Speicherort eintragen -> Finish
- > Anschließend sollten Files geladen werden und im Git RepositoryTab angezeigt
- Neues JavaProjekt anlegen und JUnit4 hinzufügen(Libraries, Add Library JUnit4)

- Über Rechte Maustaste auf src-> delete src folder löschen
- Über Rechte Maustaste auf Projekt -> Properties -> Build Path Git Ordner/Sourcecode hinzufügen und als src benennen

Zum bearbeiten datei bearbeiten.. im Repository View zu Datei navigieren -> rechte maustaste Add...

Zum Pullen des Masters Rechte Maustaste auf "das ganz oben im Repository View" und pull

Zum Push in den Masters Rechte Maustaste auf "das ganz oben im Repository View" push und Custom URI von Hand eintragen

Theoretische Arbeitsweise von GIT:

- Jeder hat sein eigenes Git Repository
- Beim Arbeiten an Dokumenten entstehen immer neue Branches(Branch ^= Ast), die extra laufen
- Es gibt eine Hauptbranch namens Master in die die eigenen Änderungen eingemergt werden.
- Alle Branches(inklusive Master) liegen auf dem eingen System. D.h. auch wenn mit Master gemergt wird muss er damit es für Alle erreichbar ist auf den Server(liegt auf DHBW Servern) gepusht werden.

Organisatorisches

- Ansprechpartner:
 - Test/Requirements: Hr. Rentschler
 - Moduldesign/Projektmanagement: Hr. Stuckert
- Es muss eine Mail an Herrn Stuckert geschrieben werden bezüglich des Reviews des CRS(Filip Haase).
- Nächster Meetingtermin: Dienstag 04.10.2011
- Fertigstellung der einzelnen CRS-Teile soll bis zum 03.10.2011 20:00 Uhr geschehen und zu Mathis Hauschild geschickt werden, damit dieser bis zum 04.10.2011 das vollständige Dokument erstellen kann.
- Business-Case: Wird von Sebastian Korlewski ebenfalls bis 03.10.2011 20:00 Uhr ausgearbeitet und dann ebenfalls an Mathis zum Review und Feinschliff geschickt.
- Präsentation: Wird von Mathis und Jonas ausgearbeitet und gehalten.

04.10.2011 – 3. Teammeeting

Agenda

- CRS Besprechung
- BC Besprechung
- Organisatorisches

CRS Besprechung

Das CRS ist von Mathis Hauschild zusammengeführt worden, ein Inhaltsverzeichnis eingefügt und an entscheidenden Stellen wurde noch ein wenig ausformuliert. Außerdem kamen noch einzelne Fragen auf.

- Es wird nur auf IPv4 eingegangen und IPv6 wird nicht erwähnt, die Frage ist ob es vlt. Allgemeiner gehalten werden soll? *[Diskussion im Team]* Es soll alles, inklusive der Bilder, so allgemein gehalten das weder IPv4 noch IPv6 erwähnt werden.
- Sollen die Requirements besser dargestellt werden? *[Diskussion im Team]* Die Requirements sollen in einer Tabelle dargestellt werden.
[Allgemeines Gelächter über das Wochenende. Außerdem Einführung des „Schweige-Fuchses“ um die Teaminterne Kommunikation zu optimieren]
- Was für ein Name soll dem „neuen Config-File“ gegeben werden? Vorerst wird es „Predefined-Value-Configuration-File“ genannt.

Das CRS soll bis zum 05.10.2011 von Mathis fertiggestellt werden um es dann zum Review einzureichen.

BC Besprechung

Der Business Case wurde von Sebastian erstellt und von Mathis überflogen. Dabei kamen folgende Fragen auf:

- Es fehlt ein vordefiniertes Budget.
- Wie berechnen wir im Risiko Management das Geld?

-> Zahlen werden von Jonas berechnet und eingetragen, dafür liefert Filip ihm eine Übersicht über alle Arbeitspakete. Anschließend wird der Text von Sebastian nochmal überflogen und zum Review eingereicht.

Organisatorisches

- Wir werden nach einer geeigneten OpenSource Lizenz suchen. GPL darf wenn es die Autoren sagen verändert werden. Jonas informiert sich genauer welche Möglichkeiten offen stehen.
- Zeilenumbrüche im Logfile sollen unter Windows und Linux-Betriebssystem leserlich gemacht werden
- Ab jetzt werden Dokumente im kleineren besprochen, um anschließend die Finale-Version im Plenum zu präsentieren.

07.10.2011 – 4. Teammeeting(Kurzmeeting zur Aufgabenverteilung)

Nach dem Review von 06.10.2011 und einer Mail mit allen Anforderungen von Herr Stuckert ergaben sich folgende Aufgaben:

- Protokolle abgabefertig machen und in die Vorlage bringen: Filip
- Präsentation: Jonas/Matthis
- Änderungen am CRS die sich aus dem Review ergeben haben: Jonas
- Neue Grafik zu STAF/STAX: Fabi
- BC Überarbeitung aufgrund des Reviews: Sebastian
- Gant-/Netzplan aus den Arbeitspaketen: Christopher

11.10.2011 – 5. Teammeeting

Fazit von der aktuellen Arbeit:-Alle sind zufrieden mit den Ergebnissen und der Zusammenarbeit in der Gruppe.

Planung der **Aufgaben**:

- SRS Layout durchlesen,damit man eine Vorstellung hat, um was es sich genau handelt bzw wie eins erstellt wird. Am besten zu Donnerstag 13.10.2011 **spätestens** aber zu Dienstag 18.10.2011
- Donnerstag 13.10.2011 wird git aufgesetzt.
- Quellcodeanalyse wird am 18.10.2011 bei Jonas gemacht. -Pizzaessen ebenfalls am 18.10.2011 bei Jonas.

13.10.2011 – 6. Teammeeting (Git Einrichtung)

Agenda

- SRS/SAS
- GIT

SRS/SAS

SAS / SRS bis Dienstag lesen! Gedanken machen!

Git

Voraussetzung: egit

Package Explorer -> Import -> Git -> Projects from Git -> Multicaster -> Clone:

URI: <http://mcastor@github.com/Samuirai/MultiCastor.git> (nutzer: mcastor; passwort:

ldvmc2011) -> Next -> Finish

Dann kommt eine Liste mit "MultiCastor" -> Next -> Finish ... nun sollte im Package Explorer das Projekt zu sehen sein.

Link zu github: <https://github.com/Samuirai/MultiCastor> (Benutzer und Passwort siehe oben)

Name + E-Mail eintragen in gitconfig:

```
$ git config --global user.name "Firstname Lastname"
```

```
$ git config --global user.email your_email@youremail.com
```

```
git-config --l -> alle Config-Einstellungen anzeigen
```

18.10.2011 – 7. Teammeeting (Vertiefung SAS/SRS)

Agenda

1. Protokollvergabe
2. Google Kalender
3. SRS/SAS – Allgemeines
 - 3.1. Google Docs
 - 3.2. Austauschen über Inhalte
 - 3.3. Verteilung der Aufgaben
4. Weitere Termine
5. Codeanalyse
6. PIZZA!
7. Anderes
8. Anhang

1. Protokollvergabe

Matthis Hauschid übernimmt die Protokollierung

2. Google Kalender

Es wird ein gemeinsamer Google Kalender angelegt, in den alle SWE-relevanten Termine eingetragen werden

3. SRS/SAS

Die Dokumente sollten in zwei Wochen fertig sein

GDocs:

- Die Dokumente aufgeteilt in Kapitel in Google Docs anlegen -> jeder hat aktuelle Version
- Versionsverwaltung: jedes Teildokument hat eigene Versionen...am Ende in einem Baum wird die gesamte Versionierung aufgezeigt

SRS - Allgemein

- Grafiken: UML mit dem Tool, Rest: Skizze an Fabi (returns awesome image)
- Analyse Vorgängerdokument -> evtl. Übernahme einiger Teile
- Punkt 2.2: Glossar auslagern in GDocs (siehe Fragen 4) -> jeder, der Fachbegriffe nutzt, **muss** sie ins Glossar eintragen
- Punkt 2.3: vorherige Gruppe hat den Punkt ausgelassen (siehe Frage 5)
- Punkt 2.4: Um diesen Punkt zu lösen, wurde die Vorlesung vom 18.10 anhand der digitalen Notizen rekapituliert
- Punkt 3.2: Use-Cases aus CRS 1zu1 übernehmen.
 - UC 1 - 3 können teilweise aus Vorgängerversion des SRS übernommen werden (*Layer-2-Funktionalität hinzufügen*)
- Punkt 4.2: Feedback zum CRS abwarten: wenn Kritik, dass es nicht drin ist, kommt es rein, andernfalls wird es weggelassen

SRS – Zu erstellende Grafiken

- Punkt 2: zwei UML Diagramme: Abläufe im Einsatzbereich, Darstellung der systemrelevanten Abläufe im Einsatzbereich
- Punkt 2.3: UML Klassendiagramm - noch unklar, was und wie (siehe Frage 5)
- Punkt 2.4: Aktivitätendiagramm erstellen
- Punkt 3.2: Use-Case-Diagramm mit dem Tool erstellen (Änderungen bedenken, die beim Besprechen der Präsentation aufgetreten sind)

SRS – Fragen an die Dozenten

- Ist das SRS wirklich auf 30 Seiten begrenzt
- Wo ist der konkrete Unterschied der Zielbestimmung im CRS/SRS?
- Punkt 2.1: Handelt es sich um die kurzgefassten Strukturen aus dem CRS?
- bzgl. Glossar: Ist ein globales Glossar für alle Dokumente in Ordnung (insbesondere SRS/SAS)?
- Punkt 2.3: Uns ist nicht schlüssig, was im UML dargestellt werden soll.
 - Grobe Struktur des Programms?
 - Abhängigkeiten von Fachbegriffen aus dem Glossar?
 - Beschreibung des Problembereich wie in Punkt 2.1 - wo wäre der Unterschied?
- Punkt 4: Unterschiede zwischen CRS / SRS (/ SAS Punkt 2.2)? Können die Punkte so übernommen werden?
- Punkt Anhang: sind Anhänge zwingend erforderlich oder optional und zählt Anhang zur Seitenzahl?

SAS – Allgemein

- kurzer Einschub: nach Evaluierung unseres Workflows, Einigung aufs Minimalprinzips
- Punkt 3: Struktur wird von HP-Vorgängern (Punkt 2.1) übernommen (weniger Farben)
- Punkt 5: FF kümmert sich um STAF/STAX, SK nimmt MMRP, Rest entwickelt GUI.
 - FH koordiniert/verwaltet den Entwicklungsfortschritt sowie die Modulbeschreibung

- JT koordiniert das Terminliche
- Punkt 6: halbe Woche vor interner Fertigstellung von SAS gemeinsam durch Punkt 6 gehen und schauen, was abgedeckt ist und was noch fehlt

SAS – Fragen an die Dozenten

- Ist die Seitenanzahl unbegrenzt, oder wurde nur vergessen, sie anzugeben?
- In welchem Detailgrad ist die Grafik unter “Allgemeine Hinweise” zu erstellen und wo im Dokument ist sie zu platzieren?
- siehe Frage 3 bei SRS
- Punkt 2.1: Vorgänger haben nur MC-Klassendiagramm gemacht. Sie sind nicht auf Nachbarsysteme eingegangen, sollen wir dies machen oder nicht?

Aufgabenverteilung im SRS/SAS

- anhand dieses Protokolls schickt JT die Mail mit den Fragen an die Dozenten
- FF erstellt Grafiken
- FH koordiniert Modulerstellungen im SAS (Delegation an Teammitglieder)
- SK übernimmt SRS Punkt 2 - Produkteinsatz
- JT übernimmt SRS Punkt 3 - Produktfunktionen (teilweise Delegation an FH, SK)
- FH übernimmt SRS Punkt 4 - Produktcharakteristiken
- MH übernimmt SAS Punkt 1 - Einführung
- CW übernimmt SAS Punkt 2 - (wenn was gemacht werden muss)
- Restliche Punkte des SAS können so nicht aufgeteilt werden, Aufteilung:
 - FF: STAF/STAX in Punkt 3 - 5
 - SK: MMRP in Punkt 3 - 5
 - CW: Module der GUI (Punkt 5)
- MH & JT am Montag bis Mittwoch: Umsetzung ausgesuchter GUI-Punkte

4. Termine

- Termine siehe GKalender
- Präsi: Präsentatoren werden beim Teammeeting am 03.11 festgelegt.
- **WICHTIG:** terminliche Absprache mit den Dozenten, evtl. Präsentation nach hinten verlegen (80 % der Zeit für Planung, Analyse atm absolut nicht gegeben)

5. Codeanalyse

Wurde auf Grund von Zeitmangel verlegt

6. PIZZA!

- es werden zwei mehrheitsfähige Familienpizzen bestellt
 - erste: Salami (easy decision)

- zweite: nach ausführlicher Evaluation aller möglichen Möglichkeiten wurde Salami Schinken Hackfleisch gewählt

7. Anderes

- Link zum Git-Repo: <http://mcastor@github.com/Samuirai/MultiCastor.git>
- Github-Issues zur Verwaltung der Programmieraufgaben verwenden (<https://github.com/Samuirai/MultiCastor/issues>), andere Termine via GKalender
- Die E-Mail an Herr Rentschler und Herr Stuckert mit den ausgearbeiteten Fragen wurde von Jonas am 19.10.2011 versendet. (siehe Anlagen)

8. Anlagen

E-Mail an H. Rentschler / H. Stuckert vom 19.10.2011

Antworten aus Vorlesung vom 25.10.2011 unter den Punkten notiert

Guten Tag Herr Rentschler,
Guten Tag Herr Stuckert,

wir saßen Gestern in unserem Team zu einem langen Meeting zusammen, in dem wir das bestehende Programm und die Dokumente SAS und SRS analysiert haben. Darüber hinaus haben wir für die kommende Zeit Aufgaben im Team aufgeteilt.

Bei diesem Prozess sind jedoch noch einige Fragen aufgekommen um deren Beantwortung wir sie bitten möchten:

- Ist das SRS wirklich streng auf 30 Seiten begrenzt? Wir haben im CRS alleine 12 Usecases spezifiziert. Wenn wir diese mit ähnlichem Umfang, wie in Ihrer Vorlage beschreiben (jeweils ca. 3 Seiten), wäre allein hierdurch die 30 Seiten Grenze überschritten. An welchen Stellen sollen wir ggf. Inhalte einsparen?

Inhalt ist nicht begrenzt, aber kein "Blabla".

- In CRS, SAS und SRS ist jeweils mit gleicher/ähnlicher Struktur der Punkt Zielbestimmung enthalten. Wo soll der konkrete Unterschied zwischen den Beschreibungen liegen oder kann die identische Beschreibung jeweils wieder verwendet werden?

Sehr ähnlich, fast Copy an Paste (Umfang: 1/2-3/4 Seite)

Beispielhaft wurde unsere CRS-Inhalt betrachtet ("sehr schwierig noch was zu ändern").

Ergebnis: Inhaltlich identisch. → Keine Änderung notwendig, C&P okay

- zu Punkt 2.1 im SRS: Können hier die Strukturen aus dem CRS kurzgefasst und wieder verwendet werden? Falls nicht, wo liegt der wesentliche inhaltliche Unterschied.

Es wird ein Testtool benötigt, dass MMRP spricht... Grafik von Tafel dazu (Saba)

- bzgl. dem Glossar: Wir würden gerne ein globales Glossar für alle Dokumente anlegen und mit entsprechenden Fußnoten auf dieses verweisen. Ist dies in Ordnung (insbesondere bei SRS/SAS)?

JA

Bitte immer Dateinamenskonvention beachten!

- Punkt 2.3 im SRS: Uns ist hier nicht schlüssig, was genau im UML-Diagramm dargestellt werden soll. Folgende Möglichkeiten sind unserer Ansicht nach denkbar:
 - Grobe Struktur des Programms → NEIN
 - Abhängigkeiten von Fachbegriffen aus dem Glossar
 - Beschreibung des **Problembereich wie in Punkt 2.1** - wo wäre der Unterschied

AW: Usability, MMRP, (STAF/STAX). Ggf. auch Screenshots Okay - AKTIVITÄTENDIAGRAMM: Workflow. & Problembereiche herausstellen.

- Wo genau liegen die Unterschiede zwischen CRS (Punkt 4)/ SRS (Punkt 4)/ SAS Punkt (2.2)? Können wir hier ggf. die bestehenden Inhalte aus dem CRS wieder verwenden?

Ist in unserem Fall das selbe: Eventuell umformulieren.

- zum Anhang des SRS: sind alle Anhänge zwingend erforderlich oder optional und zählt der Anhang zur Seitenzahl?

*optional - **Bild auf jeden Fall:** Kästchenbildchen wie das Paket aufgebaut ist. (Aus Standart. Auf jeden Fall in Anhang. Auszug aus Spezifikation)*

- Ist die Seitenanzahl im SAS unbegrenzt bzw. muss lediglich das Verhältnis zwischen Grafiken und Text eingehalten werden?

Ja

- In welchem Detailgrad ist die Grafik unter "Allgemeine Hinweise" im SAS zu erstellen und wo im Dokument ist sie zu platzieren?

Gar nicht. Sie dient nur zur Abstrahierungsbeschreibung.

- zu Punkt 2.1 im SAS: Unsere Vorgänger haben nur MC-Klassendiagramm angelegt. Sie sind nicht auf Nachbarsysteme eingegangen. Sollen wir die Nachbarsysteme hier betrachten oder nicht?

Nachbarsysteme darstellen: Kommunikation funktional beschreiben.

- Noch eine organisatorische Frage: Haben wir es richtig verstanden, dass die Dokumente in der Klausurwoche abgegeben und präsentiert werden sollen?

Eigentlich sollte ja ein Großteil der Zeit in Analyse, Planung und Design investiert werden. Bei dem gesetzten Termin würde der größte Teil der Zeit (komplettes 4. Semester) in die Implementierung fließen.

E-Mail schreiben mit Terminanfrage für Review und Feedback.

Vielen Dank schon vorab für ihre Antworten.

Gerne können sie ihre Antworten auch an die anderen Gruppen schicken, denn sie sind vermutlich für alle interessant.

Wir würden uns jedoch freuen, wenn sie unsere verwendete Arbeitszeit zur Ausarbeitung der Fragen bzw. der detaillierten Analyse der Dokumente dann gesondert positiv in ihre Benotung einfließen lassen.

Viele Grüße
Jonas Traub

Team 4 - MCastor2.0

Antwort von Herr Rentschler vom 22.10.2011

(CC: Kursverteiler, H. Stuckert)

Hallo Herr Traub,

es ist kein Problem, das SRS mit mehr als 30 Seiten zu erstellen.

Aber bedenken sie bitte, was das SRS enthalten soll:

Eine Beschreibung des zu entwickelnden Systems -welches die Anforderungen im CRS realisieren soll- und keine detaillierte Wiedergabe des Lösungswegs, wie Sie dahingekommen sind!

Sie können das SRS auch später noch verfeinern (immer Dokumenten-Version hochzählen!), aber zur Abgabe sollte schon die wesentliche Systembeschreibung aus Black-Box-Sicht erkennbar sein.

Bitte nicht copy&paste zwischen den Dokumenten. Immer auf die jeweilige Abstraktionsebene fokussieren: CRS enthält die Beschreibung des WAS, SRS die des Black-Box-WIE (Workflows, GUI-Dialoge Interfaces), etc.) und die SAS die des White-Box-WIE (SW-Architektur)!

Im SAS einer StandAlone-PC-Applikation haben Beschreibungen von Nachbarsystemen nichts zu suchen!

Am Dienstag gerne mehr dazu!

Viele Grüße,
Markus Rentschler

Antwort auf Antwort von Herr Rentschler vom 24.10.2011

(CC: Kursverteiler, H. Stuckert)

Guten Tag Herr Rentschler,
Guten Tag Herr Stuckert,

Vielen Dank für Ihre kurze Vorabantwort per Mail. Wir freuen uns auf ihre detaillierten Erläuterungen, zu unseren Fragen im Einzelnen, in der kommenden Vorlesung.

Viele Grüße
Jonas Traub

Team 4 - MCastor2.0

Von Herrn Rentschler am 25.10.2011 gewünschte Terminanfrage

Guten Tag Herr Rentschler,
Guten Tag Herr Stuckert,

wie eben in der Vorlesung besprochen sende ich ihnen folgende Terminanfrage:
Wir würden gerne Zeitnah das angekündigte Feedback zu unserem CRS und der Präsentation erhalten, damit wir Kritikpunkte noch bei der Erstellung von SAS/SRS berücksichtigen können. Wann können wir dieses Feedbackgespräch ansetzen?

Viele Grüße
Jonas Traub

Team 4 - MCastor2.0

20.10.2011 – Mitschrift der Unterschiede SRS/SAS

[nach Beschreibung von Herr Stuckert 20.10.2011]

- Jeweils Ergebnis der Analysephase
- SRS:
 - Anforderungen und Schnittstellen zum Benutzer
 - wird dem Kunden vorgelegt
 - Vertragsgrundlage (Exakte Beschreibung)
 - Funktionale Anforderung "raus holen"
- SAS:
 - Bekommt Kunde nie zu sehen
 - Coderichtlinien
 - Abhängigkeiten
 - Alles was für Entwickler wichtig ist
 - Entwicklersicht und nicht Kundensicht
 - MVC Architektur?
 - Client Server Architektur?
 - Anwendungs-Gesamtkonzept

- Mit den beiden Dokumenten kann jeder Entwickler “in seine Komponente gehen und versuchen sie Forderungen umzusetzen.”

27.10.2011 – 8. Teammeeting (Besprechung des Fortschritts)

Agenda

- Verteilung des Protokolls
- Berichte
- Zeitplan
- GUI V1.5

Protokollvergabe:

Das Protokoll wird von Filip Haase verfasst

Berichte – Aktueller Stand

Jeder beschreibt kurz, was er gemacht hat und wie der aktuelle Stand ist: Insbesondere natürlich das, was für die Arbeit der Anderen wichtig sein könnte

Jonas:

- Produktcharakteristiken fertiggestellt
- Mit Matthis GUI soweit überarbeitet was sie als Issues zugeteilt hatten(Titeländerung/neue Tabs)

Matthis:

- noch keine Issues geclosed da er mit Jonas zusammen erstmal Fenster zusammenlegung gemacht hat
- Vorschlag Sachen von FaceBook auf GoogleDocs zuverlegen

Chris:

- GUI: Button Issue oberflächlich erledigt(wird noch technisch gemacht)
- About Issue closed (Noch Bild wird aber noch in Text abgeändert)
- LogFile Bug geclosed

Filip:

- Gui Issues geclosed
- Bugs geclosed

Seba:

- Hat Bachelorarbeit zu MMRP durchgelesen und hat Verständnis zu den nötigen Aspekten Dokumente
- Muss noch einzelne Fragen zu JoinAll/JoinEmpty an Rentschler/Stuckert stellen da sich hier die Vorlage von Ihnen nicht mit dem eigentlichen Standards übereinstimmt
- Wird alles voll fertig gemacht bis Dienstag(UseCases und SAS Inhalt)

Fabi:

- Hat sich stark mit STAF/STAX und PATH Variablen beschäftigt
- Demo funktioniert jetzt
- STAF: Man kann Staf Clients laufen lassen und mit Hilfe eines STAF Service ein TestProgramm auf einem Client laufen lassen und die Ausgabe bekommt man auf dem Server gesammelt. Durch JAVA kann man die Kommunikation aufrechterhalten damit die Handler nicht immer sofort beendet werden.

Zeitplan

Bis Dienstag 01.11.2011 - **1200** alles fertig bei Matthis.

Aufgabenverteilung

Matthis portiert alle wichtigen Dokumente von Facebook auf GoogleDoc. Dabei gilt für Bugs, dass nur wenn der Bug behoben werden soll, der Bug als Issue angelegt wird.

SAS

Gliederung SAS:

1. Einführung [MH] Überarbeitet der Einleitung. Die 3 Hauptpunkte werden noch eingebaut
Allgemeine Einleitung

2. Systemstruktur Übersicht(vgl. SAS Vorgänger Punkt 2.1)[MH]

3. Benutzeroberfläche [CW] (vgl. SAS Vorgänger und unserem SRS)

3.1. Einführung

3.2. Technische Dokumentation

3.3. Systemdesign und Programmstruktur

4. Configuration [MH]

- Inkrementelles Laden/Speichern

- Multilingualität(hier auch Manual PDF aufnehmen)

- Trennung von Programm und User Config

5. MMRP [SK]

5.1. Einführung

5.2. Technische Dokumentation

5.3. Systemdesign und Programmstruktur

6. STAF/STAX [FF]

6.1. Einführung

6.2. Technische Dokumentation

6.3. Systemdesign und Programmstruktur

7. Programmreaktion im Fehlerfall[FH]

8. Systemübersicht[CW] Systemübersicht Inhalt übernehmen und **Umschreiben**

9. Technische Konzepte [JT](nochmal durchschauen ob alles abgedeckt ist. Di/Mi wenn alles bei Jonas ist)

SRS

[MH] Einleitung: Es muss Einleitung erstellt werden und ein Satz zu Dokumentation(Alles unter /LF50/) hinzugefügt werden soll.

[SK] Punkt 2 muss noch dazu: Alles unter /LF20/ Programmreaktion im Fehlerfall einfügen

[SK] UseCases machen

[SK] Anhang zu MMRP machen Thema"MMRP for dummies" (Aufjedenfall: Auszug aus Spezifikation und Paketaufbau-Bild siehe Protokoll vom 18.)

[JT] Alles unter /LF70/(Programmkonfigurationen) in SRS noch in die UseCases einfügen

[FH] UseCases fertig machen

[FF] Anhang zu STAF/STAX hinzufügen. Thema:"Was ist STAF/STAX?/ for dummies)

JEDER zu seinem Teil noch Vorlage von Rentschler/Stuckert anschauen und abgleichen und schauen, dass es pünktlich bis Dienstag fertig ist.

Dokumentation

Matthis übernimmt Protokollzusammenfassung und Einfügen ins Layout. Das Gleiche gilt für Anlagen – insbesondere Glossar

Glossar

Das Glossar wird bis Dienstag fertig gestellt.Begriffe werden mit 1-2 Sätzen beschrieben. Insgesamt werden etwa 60 Begriffe erwartet.

GUI V1.5

Es kam die Frage nach dem Termin zur Fertigstellung der Version 1.5 auf. Dieser wird nächstes Semester sein.

Dokumentversionen

Versionsnr	Datum	Autor(en)	Kommentar
V 0.1	12.08.2011	Stuckert/ Rentschler	Dokument angelegt
V 0.2	16.09.2011	Haase	Sitzungsprotokoll Kickoffmeeting erstellt
V 0.3	20.09.2011	Fäßler	Protokoll Kundengespräch erstellt
V 0.4	23.09.2011	Koralewski	Sitzungsprotokoll Teammeeting 1 erstellt
V 0.5	27.09.2011	Traub	Template/Formatvorlage erstellt
V 0.6	29.09.2011	Westphal	Sitzungsprotokoll Teammeeting 2 erstellt
V 0.7	30.09.2011	Haase	Git-Installation Protokoll erstellt
V 0.8	04.10.2011	Fäßler	Sitzungsprotokoll Teammeeting 4 erstellt
V 0.9	07.10.2011	Traub	Sitzungsprotokoll Teammeeting 5 erstellt
V 1.0	09.10.2011	Haase	Zusammenführung der Protokolle und Fertigstellung des Dokuments mit allen Protokollen bis zum 09.10.2011
V 1.1	28.10.2011	Hauschild	Protokolle vom 11.10 bis zum 18.10, verfasst von SK, CW, FH & MH hinzugefügt
V 1.2	29.10.2011	Hauschild	Protokolle vom 20.10 & 27.10, verfasst von JT & FH hinzugefügt

Lizenz/License:

© Fäßler, Haase, Hauschild, Koralewski, Traub, Westphal

Dokumentversion/Document Version:

Titel: Sitzungsprotokolle

Version: V 1.2 (15. November 2011)

Autoren/Authors Projektteam/Project team:

- Jonas Traub (Projektleiter)
- Filip Haase (Leading Engineer)
- Matthias Hauschild (Documentation)
- Fabian Fäßler (Engineer/Tester, Expert on STAF/STAX)
- Christopher Westphal (Engineer/Tester, Expert on usability)
- Sebastian Koralewski (Engineer/Tester, Expert on MMRP)