



Projektantrag

Das Ziel des Projektes ist es Aufzuzeigen wie Eclipse RCP 3 Applikationen nach Eclipse RCP 4 migriert werden können

Das vorliegende Dokument ist der Projektantrag der die beteiligten Parteien über den Projektrahmen, die Projektziele und den Projektumfang informieren soll.



Autor:	Mike Rothenbühler
Version:	1.0
Status:	In Arbeit
Ablage:	https://github.com/MikeR13/MAS/blob/master/Deliverables/Projektantrag.docx
Institution:	Hochschule für Technik und Informatik Bern
Verteiler:	Brawand Ueli, Hoffmann Marc, Rothenbühler Mike

Versionkontrolle

Datum	Version	Autor	Bemerkungen
09.05.2013	0.1	MIRO	Erster Wurf
12.05.2013	0.2	MIRO	Weitergefahren mit Planung und Organisation
15.05.2013	0.3	MIRO	Korrekturen / Ergänzungen nach 1. Review
17.05.2013	0.4	MIRO	Korrekturen / Ergänzungen, Risiken
20.05.2013	0.5	MIRO	Korrekturen / Ergänzungen, Risiken
21.05.2013	1.0	MIRO	Erstellung finale Version mit Überarbeitung aller Kapitel

Inhaltsverzeichnis

1.	Einleitung	4
1.1.	Zweck des Dokumentes	4
2.	Ausgangslage	4
2.1.	Anlass und Begründung des Projektes	4
2.2.	Problemstellung	5
2.3.	Randbedingungen	5
2.4.	Situationsanalyse	5
2.5.	Erbrachte Vorleistung	5
3.	Abgrenzung	6
4.	Ziele und Lösungen	6
4.1.	Zielvorstellungen	6
4.2.	Lösungen	6
4.3.	Sicherheits- und Datenschutzaspekte	6
5.	Chancen und Risiken	7
5.1.	Risikoidentifizierung, -bewertung und -quantifizierung	7
5.2.	Risikobehandlung	8
6.	Mittelbedarf	9
6.1.	Hardware	9
6.2.	Software	9
6.3.	Personal	9
6.4.	Sonstiges	9
7.	Planung und Organisation	9
7.1.	Projektorganisation	9
7.2.	Termine	10
7.3.	Prioritäten	11
7.4.	Projektmethode	11
8.	Wirtschaftlichkeit	11
8.1.	Aufwände	11
8.2.	Nutzen	11



1. Einleitung

1.1. Zweck des Dokumentes

Dieser Projektantrag informiert die an „Migration von Eclipse 3.x nach Eclipse 4“ beteiligten Parteien über den Projektrahmen, die Projektziele und den Projektumfang. Das Dokument soll in erster Linie einen Überblick über das Projekt ermöglichen.

2. Ausgangslage

2.1. Anlass und Begründung des Projektes

Die SBB IT haben in mehreren Projekten Eclipse 3.x Rich Client Platform (im Folgenden RCP genannt) im Einsatz. Die neue Version Eclipse 4 RCP wurde bereits in mehreren neueren Projekten eingesetzt und hat sich bewährt.

Eclipse RCP kurz erklärt (aus [http://de.wikipedia.org/wiki/Eclipse_\(IDE\)](http://de.wikipedia.org/wiki/Eclipse_(IDE))):

„Eclipse bietet die Rich Client Platform, welche es Anwendungsentwicklern ermöglicht, basierend auf dem Eclipse Framework, von der Eclipse-IDE unabhängige Anwendungen zu schreiben. Eine Übersicht befindet sich auf der Eclipse-Homepage. <http://eclipse.org/community/rcp.php> Beispielsweise basiert IBM Workplace Clients auf Eclipse RCP.

Die folgenden Komponenten (Plug-ins) werden mindestens für ein Eclipse-Rich-Client-Platform-Programm benötigt:

- Eclipse Core Platform – steuert den Lebenszyklus einer Eclipse-Application
- Standard Widget Toolkit (SWT)
- JFace

Weitere Eclipse-Komponenten, wie das Hilfesystem oder das automatische Aktualisierungssystem, können auch eingesetzt werden. Üblicherweise wird das Eclipse-OSGi-Framework Equinox zum Kombinieren („bundling“) der Komponenten eingesetzt.“

Das Rail Control System (RCS) ist eine der grössten Eclipse RCP Anwendungen bei den SBB. RCS soll mindestens noch 10 Jahre im Einsatz sein und kontinuierlich ausgebaut werden. Eclipse 3 wurde deprecated, es findet keine Weiterentwicklung mehr statt. RCS muss also auf Eclipse RCP 4 migriert werden.

Eclipse RCP 4 bietet ein moderneres Programmiermodell an als sein Vorgänger. Die vielfältigen APIs aus RCP 3.x wurden deutlich reduziert und vereinheitlicht. Die über das gesamte API verteilten Singletons wurden entfernt.

Neben der Vereinfachung wurden moderne Konzepte wie Dependency Injection und Declarative Styling eingeführt.

Die Implementation von Rich Client Applikation in Eclipse RCP wird mit der Version 4 flexibler und deutlich vereinfacht. Die Produktivität der Programmierer steigt, die Testbarkeit und die Wartung der Applikationen wird erleichtert.

2.2. Problemstellung

Da sich mit der Version 4 einiges an Eclipse RCP geändert hat ist eine Migration nicht einfach so zu bewerkstelligen. Es gibt aus der Community (noch) nicht viele Berichte zu gelungenen Migrationen, geschweige denn eine Anleitung wie eine solche Migration erfolgreich durchgeführt werden kann.

Es sollen Erkenntnisse gewonnen werden, wie eine erfolgreiche Migration durchgeführt werden kann, ohne dass die bestehende Applikation in den Punkten

- Funktionalität
- Performance
- Stabilität
- Usability
- Look and Feel

negativ beeinflusst wird. Die Arbeit an der bestehenden Applikation soll auch während der Migrationszeit möglich sein. Hierfür müssen Lösungen erarbeitet werden.

Mit den Erfahrungen und Ergebnissen aus der Master Thesis soll eine Migration auch für grosse Projekte relativ einfach möglich sein.

2.3. Randbedingungen

Das Projekt wird im Rahmen der Semesterarbeit und der Diplomarbeit durchgeführt. Dafür gelten die an der HTI üblichen Bedingungen.

2.4. Situationsanalyse

Um die erarbeiteten Migrationsvorschläge in der Praxis zu verifizieren, wird die Migration des RCP-Clients exemplarisch durchgeführt.

Bei dem RCS Client handelt es sich um die grösste Eclipse RCP Applikation der SBB mit folgenden Kennzahlen:

- 70 Plugins
- über 10 Hauptfenster
- Dutzende von Dialogen
- 270'000 Zeilen Code

RCS wird zur Disposition des Zugverkehrs auf dem gesamten Streckennetz der SBB verwendet.

2.5. Erbrachte Vorleistung

Das Knowhow über Eclipse RCP 4 wurde bereits vor Projektstart aufgebaut.



3. Abgrenzung

Aus Zeitgründen wird es nicht möglich sein während der Projektdauer das gesamte RCS auf Eclipse RCP 4 zu migrieren. Es werden ausschliesslich die ausgewählten Aspekte migriert.

4. Ziele und Lösungen

4.1. Zielvorstellungen

Das Ziel des Projektes ist es mindestens 5 Aspekte von Eclipse RCP 3 auf Eclipse 4 RCP zu migrieren. Als Aspekt wird zum Beispiel das plugin.xml, die Selektion oder auch Dependency Injection betrachtet.

Für jeden Aspekt soll folgendes durchgeführt werden:

Konzept

- Beschreibung des Aspektes
- Diskussion der Eclipse RCP 4 Lösung
 - o Vorteile
 - o Vergleich mit Eclipse RCP 3
 - o Einschränkungen und Risiken
 - o Qualität und Testbarkeit im Vergleich zu Eclipse RCP 3

Konkretes Beispiel RCS

- Definition Abnahmekriterien
- Migration
- Test

Die Erfahrungen im Projekt sollen in einem Bericht geschildert werden.

4.2. Lösungen

Die Lösung eines Aspektes wird jeweils während einer Iteration erarbeitet. Nach dem heutigen Wissenstand gibt es noch keine möglichen Lösungsvarianten. Nach Möglichkeit sollen jedoch dokumentierte „Best practices“ zum Einsatz kommen (zum Beispiel aus „*Eclipse 4 - Rich Clients mit dem Eclipse 4.2 SDK*“ von Marc Teufel und Jonas Helming oder „*Eclipse 4 Application Development*“ von Lars Vogel).

4.3. Sicherheits- und Datenschutzaspekte

Der Code der SBB darf nicht an Dritte gelangen. Um dies zu verhindern wird der Code ausschliesslich im SBB Netz bearbeitet und dann höchstens Ausschnitte daraus publiziert.



5. Chancen und Risiken

5.1. Risikoidentifizierung, -bewertung und -quantifizierung

Id	Risiko	Eintritts- wahrscheinlichkeit 0 - 15 % unwahrscheinlich 15 - 30 % eher unwahrscheinlich 31 - 50 % möglich 51 - 100 % wahrscheinlich	Begründung	Auswirkung des Risikos aufs Projekt	Begründung
				8 kritisch 4 gross 2 klein 1 vernachlässigbar	
1	Migration wegen unbekannten technischen Problemen nicht möglich	20 %	Mit dem heutigen Wissenstand darf davon ausgegangen werden, dass es keine technische Probleme geben sollte.	8	Das Projekt könnte nicht durchgeführt werden, die Migration wäre gescheitert
2	Zeitplan kann nicht eingehalten werden	40 %	Der Zeitplan ist sehr sportlich ausgelegt, es steht eine sehr kurze Zeit für die Bearbeitung des Projektes zur Verfügung.	4	Einzelne Aspekte könnten nicht migriert werden.
3	Funktionalität geht verloren	40 %	Eclipse RCP 4 bietet heute noch nicht alle Funktionen von Eclipse RCP 3 an.	8	Es darf unter keinen Umständen Funktionalität verloren gehen, das würde vom Auftraggeber nicht akzeptiert werden.
4	Performanceeinbussen	50 %	Es gibt Berichte darüber, dass Eclipse 4 teilweise mit Performanceproblemen zu kämpfen hat.	8	Die Applikation muss mindestens genauso performant sein wie heute. Etwas anderes wird vom Auftraggeber nicht akzeptiert
5	Verlust von Stabilität	20 %	Es sind heute keine Probleme betreffend Stabilität von Eclipse RCP 4 bekannt.	8	Eine abstürzende Applikation oder dergleichen ist absolut inakzeptabel
6	Verlust von Usability	10 %	Geringe Wahrscheinlichkeit da die Eclipse IDE seit Version 3.8 standardmässig auf E4 ausgeliefert wird.	8	RCS ist das wichtigste Arbeitswerkzeug der Disponenten und muss somit eine hohe Usability aufweisen.
7	Wartbarkeit nimmt ab	10 %	In der Wartbarkeit von Eclipse RCP 4 Applikationen erwarten wir eine eklatante Verbesserung gegenüber Eclipse RCP 3.	4	Das Projekt darf in der Wartung keinesfalls teurer werden, da die Wartungskosten bereits heute zu hoch sind.
8	Testbarkeit nimmt ab	10 %	Auch betreffend Testbarkeit dürfte Eclipse RCP 4 um einiges besser dastehen als Eclipse RCP 3.	4	Um die Qualität der Applikation zu gewährleisten muss diese auch testbar sein bzw. bleiben.
9	Look and Feel wird von Anwendern nicht toleriert	10 %	Das Look and Feel wird wohl ziemlich gleich bleiben.	4	Das Look and Feel sollte gleich bleiben da Änderungen gegebenenfalls zu Schulungen und Anpassungen von Handbüchern führen.



5.2. Risikobehandlung

Id	Risiko	Risikobehandlung
1	Migration wegen unbekannten technischen Problemen nicht möglich	Es ist umfangreiches Eclipse Knowhow im Projekt vorhanden und es bestehen Kontakte zur Eclipse-Community.
2	Zeitplan kann nicht eingehalten werden	Die Behandlung eines Aspektes soll timeboxed erfolgen. Das heisst, dass pro Aspekt eine maximale Anzahl an Stunden zur Verfügung steht und diese Zeit darf nicht überschritten werden. Wenn ein Aspekt innerhalb dieses Zeitraumes nicht erfolgreich migriert werden konnte, dann ist das nun einmal so. Die Erfahrungen werden dokumentiert und es soll kritisch hinterfragt werden, ob man die Migration des Aspektes unter Umständen andersartig hätte angehen müssen.
3	Funktionalität geht verloren	Um dieses Risiko zu minimieren, soll jeweils vor und auch nach der Migration die Funktionalität überprüft und dokumentiert werden. Die beiden Überprüfungen sollten zum selben Resultat führen. . Bestehende funktionale Testfälle müssen durchgeführt werden.
4	Performanceeinbussen	Im RCS Client sind diverse Performanceprüfungen eingebaut. Bestimmte ausgewählte Kennzahlen sollen vor und nach der Migration erfasst und miteinander verglichen werden. Falls Differenzen bestehen werden diese kritisch hinterfragt und, wenn nötig, Massnahmen eingeleitet. Bestehende nichtfunktionale Testfälle müssen durchgeführt werden.
5	Verlust von Stabilität	Nach der Migration eines Aspektes soll jeweils durch gezielte manuelle Tests festgestellt werden, ob die Applikation nach wie vor stabil läuft.
6	Verlust von Usability	Bei Änderungen der Bedienung müssen diese gemeinsam mit den Anwendern verifiziert werden.
7	Wartbarkeit nimmt ab	Die Wartbarkeit soll auch jeweils vor und nach der Migration eines Aspektes bestimmt und verglichen werden. Hier erwarten wir eigentlich eine Verbesserung.
8	Testbarkeit nimmt ab	Hier soll die Testabdeckung durch JUnit-Tests geprüft und verglichen werden. Tendenziell erwarten wir auch hier testbareren Code
9	Look and Feel wird von Anwendern nicht toleriert	Das „Look and feel“ soll vor und nach der Migration identisch sein, oder wird mit der Usability-Expertin abgestimmt werden.



6. Mittelbedarf

6.1. Hardware

PC bzw. Laptop

6.2. Software

Plattform: Java 7, Eclipse RCP
Dokumentation: Microsoft Office

6.3. Personal

Projektrealisator: Mike Rothenbühler
Projektbetreuer: Marc Hoffmann
Experte: Ueli Brawand

6.4. Sonstiges

Code Verwaltung: SVN/Repository bei SBB
Buildinfrastruktur: Jenkins / Maven bei SBB
Dokumentenverwaltung und Tasktracking: <https://github.com/MikeR13/MAS/>

7. Planung und Organisation

7.1. Projektorganisation

Rolle	Name	Email
Projektschulung	Mike Rothenbühler	michael.rothenbuehler@sbb.ch
Projektbetreuer	Marc Hoffmann	marc.hoffmann@sbb.ch
Experte	Ueli Brawand	ueli.brawand@besonet.ch

7.2. Termine

Datum	Termin	Beteiligte Personen
02.05.2013	Kickoff Meeting	Brawand Ueli
		Hoffmann Marc
		Rothenbühler Mike
21.05.2013	Abstract	Hoffmann Marc
		Rothenbühler Mike
21.05.2013	Projektantrag	Hoffmann Marc
		Rothenbühler Mike
16.06.2013	Statusbericht	Brawand Ueli
		Hoffmann Marc
		Rothenbühler Mike
30.06.2013	Statusbericht	Brawand Ueli
		Hoffmann Marc
		Rothenbühler Mike
Noch zu definieren	Review	Brawand Ueli
		Hoffmann Marc
		Rothenbühler Mike
14.07.2013	Statusbericht	Brawand Ueli
		Hoffmann Marc
		Rothenbühler Mike
28.07.2013	Statusbericht	Brawand Ueli
		Hoffmann Marc
		Rothenbühler Mike
04.08.2013	Statusbericht	Brawand Ueli
		Hoffmann Marc
		Rothenbühler Mike
Noch zu definieren	Schlussreview	Brawand Ueli
		Hoffmann Marc
		Rothenbühler Mike
12.09.2013	Eingabe Präsentationsbedürfnisse	Hoffmann Marc
		Rothenbühler Mike
12.09.2013	Abgabe Dokumentation/Anhänge	Hoffmann Marc
		Rothenbühler Mike
12.09.2013	Upload Poster	Hoffmann Marc
		Rothenbühler Mike
20.09.2013	Schlusspräsentation	Brawand Ueli
		Hoffmann Marc
		Rothenbühler Mike

7.3. Prioritäten

1. Identifikation der Aspekte und Erstellung einer priorisierten Liste
2. Ein Aspekt nach dem anderen den Prioritäten nach gemäss Kapitel 4.1. behandeln.
3. Die Timebox für die Bearbeitung eines Aspektes muss eingehalten werden. Die Einhaltung des Zeitplanes ist somit höher einzustufen als die erfolgreiche Migration eines Aspektes. In jedem Fall soll jedoch die Dokumentation des Aspektes erstellt werden.

7.4. Projektmethode

Das Projekt wird mit Kanban durchgeführt. Der Betreuer bestimmt das aktuelle Backlog und priorisiert die Backlog Items. Das Backlog kann jederzeit vom Betreuer angepasst und neu priorisiert werden.

Es dürfen maximal 2 Items gleichzeitig in Bearbeitung sein (WIP, Work in progress).

Es soll jeweils eine Fortschrittskontrolle stattfinden. Nach jedem Sprint wird mit dem Betreuer das Ergebnis diskutiert und gegebenenfalls Massnahmen bestimmt.

Eine Iteration, welche der Bearbeitung eines Aspektes gleichkommt, dauert 2 Wochen.

8. Wirtschaftlichkeit

8.1. Aufwände

Die Kosten werden nicht konkret beziffert. Es entsteht folgender Personalaufwand:

Wer	Was	Berechnung Aufwand	Total Aufwand (in h)
Buildmanager	Unterstützung beim Einrichten des Workspace	1 * 4 h	4
Marc Hoffman	Unterstützung während ganzer Projektdauer	20 Wochen * 2 h	40
Mike Rothenbühler	Projektrealisierung privat	20 Wochen * 14 h	280
	Projektrealisierung auf Projektkosten	20 Wochen * 4 h	80
Gesamt			404

Der Bedarf an Arbeitsmitteln ist im Kapitel Mittelbedarf aufgeführt. Alle Mittel sind bereits vorhanden, es entstehen somit keine zusätzlichen Kosten.

8.2. Nutzen

Die Migrationsanleitung ist Grundlage für die Migration von RCS und anderen RCP-Projekten der SBB. Durch die gesammelten Erfahrungen können die Migrationen effizienter durchgeführt werden.

Durch die Migration auf Eclipse RCP 4 kann der Lebenszyklus von diversen SBB-Anwendungen verlängert werden. Es entstehen Synergieeffekte für andere Projekte.

RCS kann - wie heute geplant – noch bis 2025 im Einsatz bleiben.

Der wirkliche Nutzen soll am Ende des Projektes in reflexiver Weise beurteilt werden.