# 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: | 0.3 |
| 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 |

Inhaltsverzeichnis

[Projektantrag 1](#_Toc356419477)

[Einleitung 4](#_Toc356419478)

[1. Zweck des Dokumentes 4](#_Toc356419479)

[Ausgangslage 4](#_Toc356419480)

[2. Anlass und Begründung des Projektes 4](#_Toc356419481)

[3. Problemstellung 5](#_Toc356419482)

[4. Randbedingungen 5](#_Toc356419483)

[5. Situationsanalyse 5](#_Toc356419484)

[6. Erbrachte Vorleistung 5](#_Toc356419485)

[Abgrenzung 6](#_Toc356419486)

[Ziele und Lösungen 6](#_Toc356419487)

[7. Zielvorstellungen 6](#_Toc356419488)

[8. Lösungen 6](#_Toc356419489)

[9. Sicherheits- und Datenschutzaspekte 6](#_Toc356419490)

[Chancen und Risiken 8](#_Toc356419491)

[10. Risikoidentifizierung 8](#_Toc356419492)

[11. Risikoursachenanalyse 8](#_Toc356419493)

[12. Risikobewertung und Risikoquantifizierung, 8](#_Toc356419494)

[Mittelbedarf 9](#_Toc356419495)

[13. Hardware 9](#_Toc356419496)

[14. Software 9](#_Toc356419497)

[15. Personal 9](#_Toc356419498)

[16. Sonstiges 9](#_Toc356419499)

[Planung und Organisation 9](#_Toc356419500)

[17. Projektorganisation 9](#_Toc356419501)

[18. Termine 10](#_Toc356419502)

[19. Prioritäten 11](#_Toc356419503)

[20. Projektmethode 11](#_Toc356419504)

[Wirtschaftlichkeit 11](#_Toc356419505)

[21. Aufwände 11](#_Toc356419506)

[22. Nutzen 11](#_Toc356419507)

# Einleitung

## 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.

# Ausgangslage

## 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))>:

„*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*

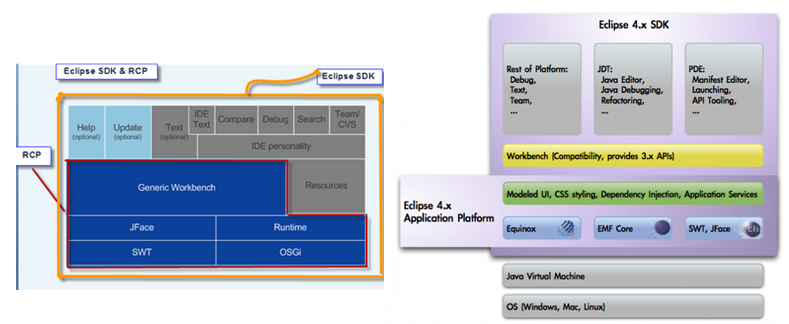
*Weitere Eclipse-Komponenten, wie das Hilfesystem oder das automatische Aktualisierungssystem, können auch eingesetzt werden. Üblicherweise wird das Eclipse-OSGi-Framework Equinox zum Kombinieren („bundling“) der Komponenten eingesetzt.*“

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.



**TODO Bild reinnehmen?**

## 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 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.

## Randbedingungen

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

## 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.

## Erbrachte Vorleistung

Das Know How über Eclipse RCP 4 wurde bereits vor Projektstart aufgebaut.

# 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.

# Ziele und Lösungen

## 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.

Konzepte

* Beschreibung des Aspektes
* Diskussion der Eclipse RCP 4 Lösung
  + Vorteile
  + Vergleich mit Eclipse RCP 3
  + Einschränkungen und Risiken
* Qualität und Testbarkeit

Konkretes Beispiel RCS

* Definition Abnahmekriterien
* Migration
* Test

Ein Aspekt gilt als erfolgreich abgehandelt wenn

* der Handbucheintrag - wie die Migration erfolgen kann - gemacht ist
* der praktische Beweis im Code erbracht worden ist, dass es funktioniert
* Risiko (Evaluation, Beurteilung, Behandlung etc..)
* Vergleich zu anderen möglichen Lösungen?
* Tests/Qualität
* **TODO** Was noch????

Die Erfahrungen im Projekt sollen in einem Bericht geschildert werden.

## 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. Bücher: **TODO**

|  |  |
| --- | --- |
| Eclipse 4 - Rich Clients mit dem Eclipse 4.2 SDK (German Edition) | Teufel, Marc, Helming, Jonas |
| Eclipse 4 Application Development | Vogel, Lars |

## 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. Der Datenschutzaspekt muss noch mit den SBB geklärt werden.

# Chancen und Risiken

**TODO**

## Risikoidentifizierung

Migration wegen unbekannten technischen Problemen nicht möglich

Verlust von Funktionalität

Performanceeinbussen

Verlust von Stabilität

Verlust von Usability

Wartbarkeit nimmt ab

Testbarkeit nimmt ab

Look and Feel wird von Anwendern nicht toleriert

## Risikoursachenanalyse

**TODO**

## Risikobewertung und Risikoquantifizierung,

**TODO**

# Mittelbedarf

## Hardware

PC bzw. Laptop

## Software

Plattform: Java 7, Eclipse RCP

Dokumentation: Microsoft Office

## Personal

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

## Sonstiges

Code Verwaltung: SVN/Repository bei SBB

Buildinfrastruktur: Jenkins / Maven bei SBB

# Planung und Organisation

## Projektorganisation

|  |  |  |
| --- | --- | --- |
| **Rolle** | **Name** | **Email** |
| Projektumsetzung | Mike Rothenbühler | [michael.rothenbuehler@sbb.ch](mailto:michael.rothenbuehler@sbb.ch) |
| Projektbetreuer | Marc Hoffmann | [marc.hoffmann@sbb.ch](mailto:marc.hoffmann@sbb.ch) |
| Experte | Ueli Brawand | [ueli.brawand@besonet.ch](mailto:ueli.brawand@besonet.ch) |

## 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 |
| TBD | 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 |
| **TBD** | Schlussreview |  |
| 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 |

**TODO Mehr Termine??**

## Prioritäten

1. Ein Aspekt nach dem anderen migrieren
2. TODO

## 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 sollen maximal 2 Items in Bearbeitung sein (WIP, Work in progress).

Es soll jeweils eine Fortschrittskontrolle stattfinden **(Wann, in welcher Form??)**

**TODO** Erwähnen Board? Soll die Iteration timeboxed, oder bis ein Aspekt fertig ist sein? ETC..

# Wirtschaftlichkeit

## 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 |

## 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 E4 kann der Lebenszyklus von diversen SBB-Anwendungen verlängert werden. Es entstehen Synergieffekte für andere Projekte.

**Gehe zu Dan Devenonge und frage wieviel RCS ca. gekostet hat und wie lange es noch bestehen wird…**

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