# Projektantrag

**BLABLABLA TODO**

|  |  |
| --- | --- |
| Autor: | Mike Rothenbühler |
| Version: | 0.1 |
| 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 |
|  |  |  |  |
|  |  |  |  |

Inhaltsverzeichnis

[Projektantrag 1](#_Toc355881718)

[Einleitung 3](#_Toc355881719)

[1. Zweck des Dokumentes 3](#_Toc355881720)

[Ausgangslage 3](#_Toc355881721)

[2. Anlass und Begründung des Projektes 3](#_Toc355881722)

[3. Problemstellung 3](#_Toc355881723)

[4. Randbedingungen 4](#_Toc355881724)

[5. Situationsanalyse 4](#_Toc355881725)

[6. Erbrachte Vorleistung 4](#_Toc355881726)

[Ziele und Lösungen 4](#_Toc355881727)

[7. Zielvorstellungen 4](#_Toc355881728)

[8. Lösungen 4](#_Toc355881729)

[9. Sicherheits- und Datenschutzaspekte 5](#_Toc355881730)

[Mittelbedarf 5](#_Toc355881731)

[10. Hardware 5](#_Toc355881732)

[11. Software 5](#_Toc355881733)

[12. Personal 5](#_Toc355881734)

[13. Sonstiges 5](#_Toc355881735)

[Planung und Organisation 5](#_Toc355881736)

[14. Projektorganisation 5](#_Toc355881737)

[15. Termine 5](#_Toc355881738)

[16. Prioritäten 6](#_Toc355881739)

[Wirtschaftlichkeit 6](#_Toc355881740)

[17. Kosten 6](#_Toc355881741)

[18. Nutzen 7](#_Toc355881742)

[Konsequenzen 7](#_Toc355881743)

[19. Bei Realisierung 7](#_Toc355881744)

[20. Bei verspäteter Realisierung 7](#_Toc355881745)

[21. Bei Nichtrealisierung 7](#_Toc355881746)

[Antrag 7](#_Toc355881747)

[22. Bisherige Entscheide 7](#_Toc355881748)

[23. Projektantrag 7](#_Toc355881749)

# 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.*“

Eclipse RCP 4 bietet ein sehr viel einfacheres 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.

Die Migration auf die neue Version ist somit auch für bestehende Eclipse RCP Applikationen durchaus ein Thema. Eclipse 3.x wird aktuell noch unterstützt, es könnte jedoch schon bald nicht mehr der Fall sein.

## 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

Das Projekt RCS (Rail Control System) wird für die Realisierung der Migrationsansätze genommen (**?? TODO**).

Der Client von RCS weist folgende Kennzahlen auf:

* 70 Plugins
* über 10 Hauptfenster
* Dutzende von Dialogen
* 270‘000 Zeilen Code

**TODO RCS kurz erklären?**

## Erbrachte Vorleistung

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

# 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.

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
* 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.

## 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.

# 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: Hudson

# 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 | TODO |

## Termine

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Datum** | **Termin** | **Beteiligte Personen** | **Rolle** | **Status ?? TODO** |
| 02.05.2013 | Kickoff Meeting | Brawand Ueli | Experte | Erledigt, 02.05.2013 |
| Hoffmann Marc | Betreuer |
| Rothenbühler Mike | Projektumsetzer |
| 21.05.2013 | Abstract | Hoffmann Marc | Reviewer | In Arbeit |
| Rothenbühler Mike | Realisator |
| 21.05.2013 | Projektantrag | Hoffmann Marc | Reviewer | In Arbeit |
| Rothenbühler Mike | Realisator |
| 16.06.2013 | Statusbericht | Brawand Ueli |  | Wie oft jetzt? |
| Hoffmann Marc |  |
| Rothenbühler Mike |  |
| 30.06.2013 | Statusbericht | Brawand Ueli |  | Wie oft jetzt? |
| Hoffmann Marc |  |
| Rothenbühler Mike |  |
| TBD | Review | Brawand Ueli |  |  |
| Hoffmann Marc |  |
| Rothenbühler Mike |  |
| 14.07.2013 | Statusbericht | Brawand Ueli |  | Wie oft jetzt? |
| Hoffmann Marc |  |
| Rothenbühler Mike |  |
| 28.07.2013 | Statusbericht | Brawand Ueli |  | Wie oft jetzt? |
| Hoffmann Marc |  |
| Rothenbühler Mike |  |
| 04.08.2013 | Statusbericht | Brawand Ueli |  | Wie oft jetzt? |
| 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 |  |  |

Evtl. noch Rolle der „Teilnehmer“

**TODO Mehr Termine??**

## Prioritäten

1. Ein Aspekt nach dem anderen migrieren
2. TODO

# Wirtschaftlichkeit

## Kosten

**????**

## Nutzen

Eclipse RCP 4 bietet ein sehr viel einfacheres 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.

Die Migration auf die neue Version ist somit auch für bestehende Eclipse RCP Applikationen durchaus ein Thema. Eclipse 3.x wird aktuell noch unterstützt, es könnte jedoch schon bald nicht mehr der Fall sein.

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

# Konsequenzen

## Bei Realisierung

## Bei verspäteter Realisierung

## Bei Nichtrealisierung

# Antrag

## Bisherige Entscheide

## Projektantrag

Irgendwo noch Abgrenzung rein!?

Qualitative und quantitative Kriterien