

Software Engineering & Projektmanagement (SE&PM)

Meilensteinbeschreibung

Stefan Biffi
Barbara Schuhmacher
Erik Gostischa-Franta
Peter Frühwirt

Institut für Softwaretechnik und Interaktive Systeme
Quality Software Engineering, <http://qse.ifs.tuwien.ac.at>

Inhaltsverzeichnis

1. Überblick	3
2. Grober Zeitplan.....	3
3. Termentypen.....	4
3.1. Meilensteine	4
3.2. Review Termine (MR's & IR's).....	4
3.3. Jour-Fixe und Tutortreffen	4
4. Arbeitspaket-Typen & Verantwortlichkeiten im Team	5
4.1. Projektmanagement	5
4.2. Vertikale Verantwortlichkeiten	5
4.3. Horizontale Verantwortlichkeiten	5
5. SCRUM Prozessmodell	7
6. Projektplanung	8
7. Angaben und Vorlagen	8
8. Meilensteine (Produkte).....	8
Meilenstein 0: Kick-Off Treffen	9
Meilenstein 1: Projektdefinition (MR-1).....	9
Meilenstein 2: Hello World (IR-1)	10
Meilenstein 3: Alpha Version (MR-2).....	10
Meilenstein 4: Beta Version (IR-2)	11
Meilenstein 5: Projektabnahme (MR-3)	11

1. Überblick

Dieses Dokument umfasst die im Rahmen der SE&PM Übung verlangten Meilensteine (Produkte), die aus Artefakten (Dokumentation) und/oder Sourcecode bestehen können. Die SE&PM Übung ist eine Teamarbeit bei der Methoden und Produkte der professionellen Softwareentwicklung als Gruppe erarbeitet werden. Es wird von einem 5-6 Personen Team ausgegangen. Die zu verwendeten Technologien sind vorgeschrieben (BPSE-Basic) während die Domäne bzw. Thema des Projekts frei wählbar sind.

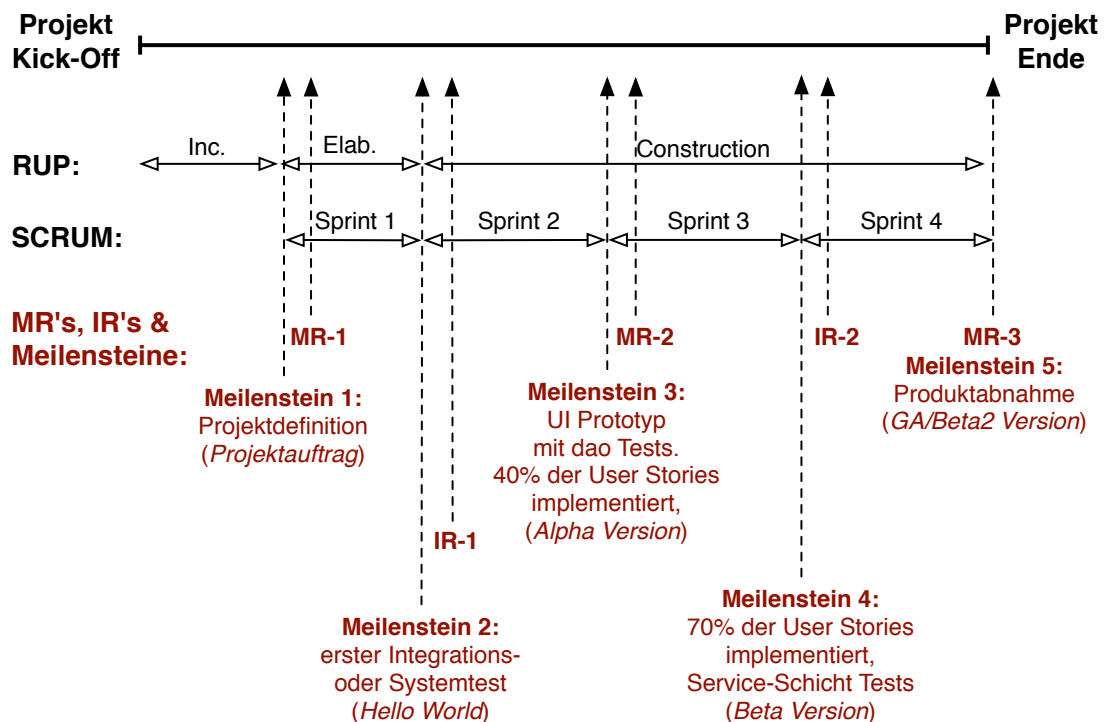
Je nach Art und Umfang des Projektes bzw. Vorgehensmodell können sich die in diesem Dokument beschriebenen Richtlinien ändern. Insbesondere der Status der Artefakte (geschätzter Prozentsatz der Fertigstellung) ist von den verwendeten Vorgehensmodellen (RUP/SCRUM/V-Model XT) und Techniken (Test-Driven, Feature-Driven) betroffen.

Das Prozessmodell, das bei den Laborübungen zu Software Engineering eingesetzt wird, ist ein iteratives und inkrementelles, das an den SCRUM Prozess angelehnt ist. Weiters werden die Phasen und Rollen des Rational Unified Process (RUP) verwendet. Nähere Informationen dazu im Abschnitt *SCRUM Prozessmodell*

2. Grober Zeitplan

Der Grobe Zeitplan für die Übung schreibt nur vor in welcher Folge die Termine abzuhalten sind. Der Zeitplan zeigt wie der Rational Unified Process (RUP) und SCRUM im Rahmen der SE&PM LU kombiniert werden, sowie die fix vorgegebenen Meilensteine und zu vereinbarende (variable) Reviews mit Tutor.

Eine konkrete Einteilung der Gruppentreffen (Jour-Fixe) und Treffen mit dem Tutor, die hier nicht eingezeichnet sind, obliegt Ihrer Verantwortung (siehe Abschnitt *Terminotypen*). Teilen Sie sich aber die Arbeitszeit gut ein.



3. Termintypen

3.1. Meilensteine

Die Namen sowie Abgabetermine der Meilensteine sind fix vorgegeben.

1. Meilenstein „ *Projektdefinition* “: Projektauftrag
 - Abgabe am 20.04.2010
2. Meilenstein „ *Hello World* “: erster Integrationstest oder Systemtest
 - Abgabe am 01.05.2010
3. Meilenstein „ *Alpha* “: UI Prototyp mit DAO Tests, 40% der User Stories Implementiert
 - Abgabe am 22.05.2010
4. Meilenstein „ *Beta* “: 70% der User Stories implementiert mit Service-Schicht Tests
 - Abgabe am 06.06.2010
5. Meilenstein „ *General Availability Version* “: Produktabnahme, MR-3
 - Abgabe am 26.06.2010

Jeder Meilenstein hat ein Produkt zu folge. Zu Projektanfang (Meilenstein 1: „*Projektdefinition*“) besteht das Produkt nur aus dem Projektauftrag, aber schon zum zweiten „*Hello World*“ Meilenstein sollen verfeinerte Artefakte, die aus dem Projektauftrag entstehen, sowie der erste Sourcecode abgegeben werden.

Bereits der erste Meilenstein wird über das SCM (Mercurial) abgegeben, indem der Projektauftrag mittels Maven Site ausgearbeitet und mit Mercurial eingchecked wird. Die BitBucket Plattform (<http://bitbucket.org/>) soll hierzu verwendet werden. Für jeden Meilenstein soll ein sog. Tag im Mercurial Repository gesetzt werden.

3.2. Review Termine (MR's & IR's)

Jeder Meilenstein wird in einem Review untersucht. Die Review-Termine werden mit dem Tutor fixiert und sollten wenn möglich ein paar Tage nach jedem Meilenstein stattfinden - damit Zeit bleibt einen ordentlichen Statusbericht anzufertigen.

Im Rahmen der SE&PM LU, teilen sich die Reviews in 3 Management Reviews (MR's) mit dem Tutor und Assistenten, und 2 Interne Reviews (IR's) nur mit dem Tutor. Zu jedem Review, wird das Produkt des jeweiligen Meilensteins analysiert und bewertet.

Für alle Review Termine gilt Anwesenheitspflicht für alle Teammitglieder. Sollte ein Teammitglied für eine Review verhindert sein, muss der Tutor so bald als möglich informiert werden.

3.3. Jour-Fixe und Tutortreffen

Zusätzlich zu den MR's und IR's, trifft sich die Gruppe ohne Tutor zu einem fix ausgemachten wöchentlichen Treffen (Jour Fixe). Tutortreffen finden in der Regel auch jede Woche statt und sind mit dem Tutor zu fixieren.

4. Arbeitspaket-Typen & Verantwortlichkeiten im Team

Das Produkt jedes Meilensteins setzt sich aus Dokumentation und/oder Sourcecode zusammen. Diese Artefakte entstehen auf Grund von 3 verschiedenen Arten von Tätigkeiten im Rahmen der Projektarbeit:

4.1. Projektmanagement

Z.B. Artefakte (Dokumentation)

Ein Artefakt ist ein Dokument (Bild und/oder Text) welches im Rahmen der Übung verlangt wird. Einige Artefakte beziehen sich auf Projektmanagement Aspekte, andere dienen dem Technischen Design der Software. In SE&PM sollen alle Artefakte mittels Maven Site erstellt werden.

Artefakte können grob in 3 Kategorien unterteilt werden:

1. *Artefakte des laufenden Projektmanagement*: z.B. Projektplan, Risikoanalyse, Stundenlisten, Statusberichte und Besprechungsprotokolle.
2. *Funktionale und Nichtfunktionale Anforderungsdokumente* wie z.B. Anwendungsfälle.
3. *Design Artefakte*: z.B. Komponentendiagramm und UI-Skizzen.

4.2. Vertikale Verantwortlichkeiten

Z.B. Technische Entwicklung, Programmierung der Produktfeatures (Sourcecode).

Jedes Teammitglied muss am Softwareprodukt mitprogrammieren und auch Tests dazu schreiben. Was programmiert wird, definiert jedes Team selber im Projektauftrag (MR-1). Welche Teammitglieder welche Features ausprogrammieren, muss spätestens beim IR-1 geklärt werden. Jene Aufteilung kann sich aber klarerweise unter Verwendung von Agilen Entwicklungsmethoden, im Laufe der Zeit ändern.

4.3. Horizontale Verantwortlichkeiten

Z.B. Team Koordinator, Technischer Architekt, Tester (Dokumentation & Sourcecode)

Werden beim Kickoff Treffen vereinbart und beim MR-1 fixiert. Die Horizontalen Verantwortlichkeiten sind:

- Projekt/Risiko Management (Rolle: Team Koordinator)
- Dokumentations-Management (Rolle: Dokument Beauftragter)
- Test Management (Rolle: Test Beauftragter)
- Architektur Management (Rolle: Technischer Architekt)
- Build- und Release-Management (Rollen: TA Stv. + TK Stv.)

Verantwortlichkeiten im Team

- **Horizontale Verantwortung:** Experte und Ansprechperson für eine bestimmte Hauptaufgabe, wie z.B. **Implementierung (Technischer Architekt, TA)**.
- **Vertikale Verantwortung:** Jede Person muss bei der Entwicklung des Endprodukts mitwirken und hat pro Iteration gewisse Artefakte bei den Experten „abzuliefern“. (**Team Koordinator, TK**)

		Tom (TK)	Jerry (TA)
Rolle	Hauptaufgabe	Feature "Import"	Feature "Export"
TK	Planung & Tracking	Stundenliste, Statusb.	Stundenliste, Statusbericht
TA	Entwurf	Komponente "Import"	Komponenten "Vorschau" und "Export"
TA	Entwurf	User Interface	User Interface / Darstellung
TA	Entwurf	Datenbank-Schema	Datenbank-Schema
Tester	Testplanung	Integrations Tests	Integrations- und Systemtests

Beispiel: Der Team Koordinator Tom muss bei der Implementierung die User Interface und Datenbank Guidelines befolgen, welche vom Technischen Architekten im Entwurf & Design erarbeitet werden. Genauso muss der Technische Architekt, in seiner Vertikalen Rolle, regelmäßig zu den Stundenlisten und Statusberichten beitragen, welche dem Team Koordinator bei Planung & Tracking helfen.

5. SCRUM Prozessmodell

Das Prozessmodell das in SEPM eingesetzt wird ist ein agiles Modell welches an SCRUM angelehnt ist.

Das zentrale und wichtigste Element von SCRUM ist der Sprint, der eine definierte Zeitdauer hat und als die "Umsetzung einer Iteration" bezeichnet werden kann. Jeder Sprint muss mit der Fertigstellung einer neuen Funktionalität enden. Ist das Sprintziel nicht erreichbar wird der Sprint abgebrochen und ein Neuer mit eigenem Ziel gestartet (was in der LVA eine Ausnahme sein soll). Zu Beginn eines Sprints findet ein Sprint Planning Meeting statt, bei dem die Aufgaben für den bevorstehenden Sprint basierend auf den priorisierten Anforderungen des Product Backlog festgelegt und im Sprint Backlog, der aus einzelnen Backlog Items besteht, fixiert werden.

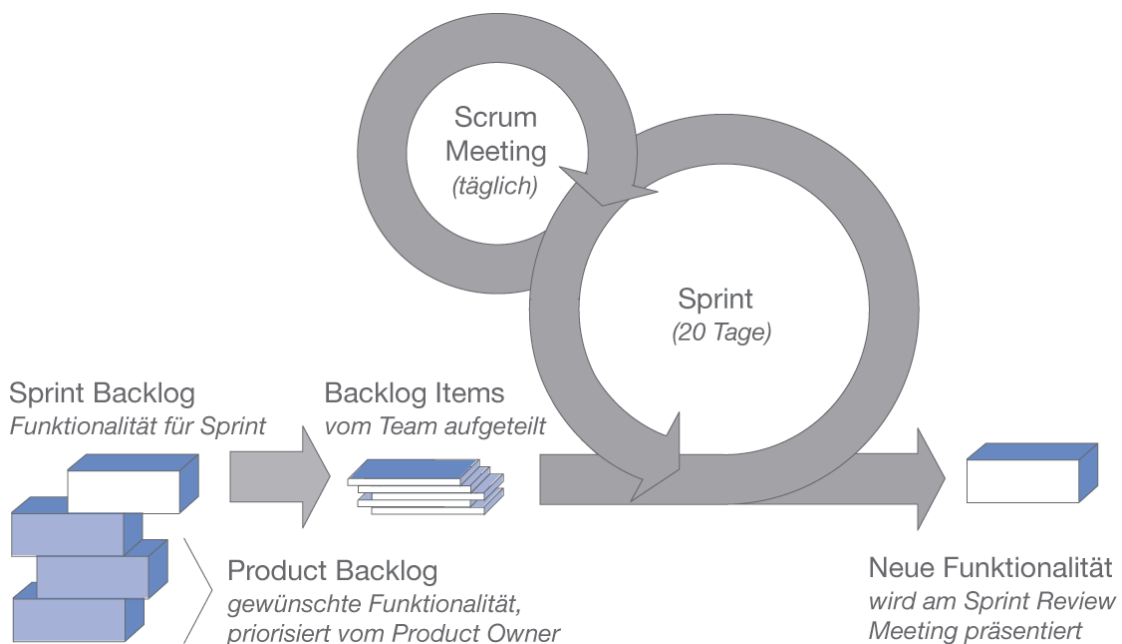
Das Product Backlog besteht aus User Stories, denen Tasks zugeordnet sind, die meist während der Sprintplanung erstellt werden. Weiters kann das Product Backlog auch Fehlerberichte (Bugreports) enthalten, welchen wiederum Tasks zugeordnet sind bzw. sein können.

Das Team verpflichtet sich damit zur Umsetzung der in den Sprint Backlog aufgenommenen Funktionalität. Des Weiteren wird in diesem Meeting ein Sprint Goal, ein bestimmtes Ziel, definiert, das die Ergebnisse des Sprints festlegt.

Eine sog. Iceberglis ist nichts weiter als der Product Backlog mit dem Sprint Backlog zusammen in einer Tabelle veranschaulicht, siehe Projektauftrag von der BPSE Webseite: <http://bpse.ifs.tuwien.ac.at/medium/projektauftrag.html#Iceberglis>

Überdies findet zeitnahen und regelmäßigen Abständen ein SCRUM Meeting statt (mindestens beim Jour-Fixe der Meilensteinabgabe), das der Team Koordinator leitet. Dieses Treffen hat zweierlei Aufgaben: einerseits den Fortschritt des Projektes zu überwachen und andererseits Hindernisse zu erkennen und zu beseitigen.

Am Ende eines Sprints erfolgt eine Meilensteinabgabe und ein Review Meeting, bei dem die neue Funktionalität, sozusagen das Inkrement, präsentiert und das Sprint Goal überprüft werden. Auf dieses Inkrement soll optimalerweise (im Zuge der LVA) aufgebaut werden. Daraufhin beginnt nach der Festlegung des neuen Sprint Backlog der nächste Sprint.



Vorgehensstrategie von SCRUM nach Schwaber und Beedle (2002)

6. Projektplanung

Die Planung sämtlicher nicht-organisatorischen Arbeitspakete (Implementierungstasks) erfolgt über BitBucket (<http://bitbucket.org/>).

Sämtliche große organisatorischen Tätigkeiten wie z.B. das Erfassen der Use Cases (Anwendungsfälle) oder User Stories, Architekturplanung, Rollout, ... muss traditionell in einem GANTT Chart erfasst werden. Nicht erfasst werden müssen sehr kurze Tätigkeiten wie z.B. Stundenlisten erstellen, Review- oder Meetingprotokolle schreiben, usw...

Die gesamte persistente Kommunikation muss offen für alle Teammitglieder und den Tutor über eine Mailingliste erfolgen, verwenden Sie dazu Google Groups (<http://groups.google.com>).

7. Vorlagen und Tutorials

<http://bpse.ifs.tuwien.ac.at/src-package/BPSE.zip>

BPSE-Basic Example, BPSE-Site Dokumentation, Design-Patterns aus dem BPSE-Buch

<http://bpse.ifs.tuwien.ac.at/arch-basic.html>

Basic Example Übersicht

<http://bpse.ifs.tuwien.ac.at/technology/tech-maven-basics.html>

Maven Basics

<http://bpse.ifs.tuwien.ac.at/technology/tech-maven-site.html>

Maven Site

http://bpse.ifs.tuwien.ac.at/technology/spring_jdbctemplates.html

Spring JDBC Templates

<http://bpse.ifs.tuwien.ac.at/technology/tech-scm-mercurial.html>

Mercurial Tutorial

8. Meilensteine (Produkte)

Alle Produkte (Dokumente und Code) müssen im Versionierungssystem verwaltet werden und stehen jederzeit in der aktuellsten Version für alle Teammitglieder und dem Tutor zur Verfügung. Die Struktur für die Verwaltung der Produkte ist im Versionierungssystem durch die Verwendung von Maven und Maven Site vorgegeben.

Es ist erwünscht, dass sämtliche Dokumente die sinnvoll in einem Wiki abgebildet werden können, wie z.B. Meetingprotokolle oder Stundenlisten im BitBucket Wiki erstellt werden. Dafür sind zu Beginn geeignete Vorlagen im Wiki zu erstellen! Alle anderen Artefakte sollen mit Maven Site abgebildet werden.

Der folgende Abschnitt beschreibt im speziellen welche Produkte zu welchen Meilensteinen fertig gestellt werden sollen. Erklärung zur Spalte „Status“ in den folgenden Tabellen:

- *aktualisiert (%)*: Das Artefakt wurde bereits vorher in einer ersten Version erstellt soll verbessert werden. Der Prozentsatz spezifiziert den ungefähren Fertigstellungsgrad des Artefakts bei der jeweiligen Meilensteinabgabe.
- *abgeschlossen*: Das Artefakt ist fertig und sollte zumindest von der Struktur her nicht mehr verändert werden.

Meilenstein 0: Kick-Off Treffen

Nr.	Artefakt
1	Artefakte des Laufenden Projektmanagement
	Besprechungsprotokoll + Rollenverteilung
2	Projektvorschlag, PV
	Ausgangssituation
	Projektbeschreibung
	Zielgruppen
	Featureliste
	Domänenmodell

Meilenstein 1: Projektdefinition (MR-1)

Nr.	Artefakt	Status
1	Artefakte des Laufenden Projektmanagement (Maven Site od. bitbucket Wiki)	
	Besprechungsprotokolle	aktualisiert
	Stundenlisten	aktualisiert
	MR-1 Statusbericht	abgeschlossen
2	Projektauftrag, PA (Maven Site + Mercurial Tag)	
	Projektbezeichnung, Projektteam & Rollenverteilung	abgeschlossen
	Ausgangssituation	abgeschlossen
	Projektbeschreibung	abgeschlossen
	Zielgruppen	abgeschlossen
	Domänenmodell	abgeschlossen
	Komponentendiagramm	aktualisiert (~25%)
	Projektbegrenzungen	abgeschlossen
	Funktionale Anforderungen (Anwendungsfälle)	aktualisiert (~25%)
	Beschreibung der Anwendungsfälle (Die 5 wichtigsten als User-Story)	aktualisiert (~25%)
	Anwendungsfalldiagramme (optional)	aktualisiert (~25%)
	AkteurInnenhierarchie	abgeschlossen
	AkteurInnenbeschreibung	abgeschlossen
	Übersicht über alle Anwendungsfälle	aktualisiert (~75%)
	Nichtfunktionale Anforderungen	abgeschlossen
	Lieferkomponenten	abgeschlossen
	Arbeitsstruktur (Horizontale Verantwortlichkeiten) und grober Projektplan auf Meilensteinebene (Work Breakdown Structure oder GANTT Diagramm)	aktualisiert (~25%)
	Iceberglis (erweiterte Feature-Liste)	aktualisiert (~50%)
	Risikoabschätzung	aktualisiert (~25%)
	Informationswesen & Besonderheiten	abgeschlossen

Meilenstein 2: Hello World (IR-1)

Nr.	Artefakt	Status
1	Artefakte des Laufenden Projektmanagement	
	Besprechungsprotokolle	aktualisiert
	Stundenlisten	aktualisiert
	Risikoabschätzung	aktualisiert (~50%)
	IR-1 Statusbericht	abgeschlossen
	bitbucket Issue Tracker	aktualisiert
	Arbeitsstruktur (Horizontale Verantwortlichkeiten)	aktualisiert
	Projektplan	
	WBS oder GANTT auf Arbeitspaketebene	aktualisiert (~50%)
2	Funktionale Anforderungen (Anwendungsfälle)	
	Beschreibung der Anwendungsfälle	aktualisiert (~75%)
	Anwendungsfalldiagramme	abgeschlossen
	AkteurInnenhierarchie	abgeschlossen
	AkteurInnenbeschreibung	abgeschlossen
	Übersicht über alle Anwendungsfälle	abgeschlossen
3	Komponentendiagramm	aktualisiert (~50%)
4	UI-Skizzen	aktualisiert (~50%)
5	„Hello World“ Prototyp mit GUI: erster Integrationstest oder Systemtest	abgeschlossen (Mercurial Tag)
6	Testartefakte	
	Testplan (Testvorgehen)	aktualisiert (~50%)
	Funktionale Testfälle / Akzeptanztests	aktualisiert (~25%)
7	Datenbankbeschreibung	
	ER-Diagramm od. UML Klassendiagramm	aktualisiert (~50%)
	Beschreibung der Attribute (bei ER)	aktualisiert (~25%)

Meilenstein 3: Alpha Version (MR-2)

Nr.	Artefakt	Status
1	Artefakte des Laufenden Projektmanagement	
	Besprechungsprotokolle	aktualisiert
	Stundenlisten	aktualisiert
	Risikoabschätzung	aktualisiert (~75%)
	bitbucket Issue Tracker	aktualisiert
	Arbeitsstruktur (Horizontale Verantwortlichkeiten)	aktualisiert
	Projektplan (WBS od. GANTT)	aktualisiert (~75%)
	MR-2 Statusbericht	abgeschlossen
2	Funktionale Anforderungen (Anwendungsfälle)	abgeschlossen
3	Testartefakte	
	Testplan (Testvorgehen)	abgeschlossen
	Funktionale Testfälle / Akzeptanztests	aktualisiert (~50%)
	Testberichte	
	Akzeptanztests (manueller Bericht)	aktualisiert (~25%)
	Unit & Integrationstests (automatisierte Berichte)	aktualisiert (~75%)
5	Datenbankbeschreibung	abgeschlossen
6	„Alpha Version“: Prototyp mit DAO Tests und 40% der Funktionalen Anforderungen (User Stories) Implementiert	abgeschlossen (Mercurial Tag)
7	UI-Skizzen	abgeschlossen
8	Präsentation zum MR-2	abgeschlossen
9	Komponentendiagramm	abgeschlossen

Meilenstein 4: Beta Version (IR-2)

Nr.	Artefakt	Status
1	Artefakte des Laufenden Projektmanagement	
	Stundenlisten	aktualisiert
	Besprechungsprotokolle	aktualisiert
	IR-2 Statusbericht	abgeschlossen
	Risikoabschätzung	aktualisiert
	bitbucket Issue Tracker	aktualisiert
	Arbeitsstruktur (Horizontale Verantwortlichkeiten)	aktualisiert
	Projektstrukturplan (GANTT od. WBS)	aktualisiert
3	Testartefakte	
	Testplan (Testvorgehen)	abgeschlossen
	Funktionale Testfälle / Akzeptanztests	aktualisiert (~75%)
	Testberichte	
	Akzeptanztests (manueller Bericht)	aktualisiert (~50%)
	Unit & Integrationstests (automatisierte Berichte)	aktualisiert
4	„Beta Version“: 70% der User Stories implementiert mit Service-Schicht Tests	abgeschlossen (Mercurial Tag)

Meilenstein 5: Projektabschluss (MR-3)

Nr.	Artefakt	Status
1	Artefakte des Laufenden Projektmanagement	
	Stundenlisten	abgeschlossen
	Besprechungsprotokolle	abgeschlossen
	Projektendbericht (MR-3 Statusbericht)	abgeschlossen
	bitbucket Issue Tracker	abgeschlossen
	Arbeitsstruktur (Horizontale Verantwortlichkeiten)	abgeschlossen
	Projektstrukturplan (GANTT od. WBS)	abgeschlossen
	Risikoabschätzung	abgeschlossen
2	Testartefakte	
	Testplan (Testvorgehen)	abgeschlossen
	Funktionale Testfälle / Akzeptanztests	abgeschlossen
	Testberichte	
	Akzeptanztests (manueller Bericht)	abgeschlossen
	Unit & Integrationstests (automatisierte Berichte)	abgeschlossen
3	Präsentation zum MR-3	abgeschlossen
4	„General Availability Version“: funktionierendes Produkt	abgeschlossen (Mercurial Tag)