



HTW Chur
Hochschule für Technik und Wirtschaft

Fachhochschule Ostschweiz
University of Applied Sciences

Projektplan

Projekt: *JQuotes*

Version 1.1

TETBörsianer

Fabian Anderegg
David Angleitner
Matthias Kohler
Norman Süsstrunk
Ralph Potztal

Änderungsgeschichte

<u>Datum</u>	<u>Version</u>	<u>Autor</u>	<u>Beschreibung</u>
2008-03-21	1.0	F. Anderegg	Dokument erstellt
2008-03-26	1.1	Ganze Gruppe	Dokument überarbeitet

Inhaltsverzeichnis

1	EINLEITUNG	6
1.1	Zweck	6
1.2	Projektüberblick	6
1.3	Lieferumfang	6
1.4	Abnahmeprozedere	6
1.5	Definitionen, Akronyme und Abkürzungen	6
1.6	Referenzen	7
2	PROJEKTORGANISATION	8
2.1	Vorgehen	8
2.2	Organisationsstruktur	9
2.3	Organisatorische Abgrenzung und Schnittstellen	10
2.4	Projektzuständigkeiten	10
3	MANAGEMENT – PROZESS	11
3.1	Managementziele und Prioritäten	11
3.2	Annahmen und Einschränkungen	11
3.3	Risikomanagement	11
3.4	Berichtswesen	11
3.5	Personalplanung	11
3.6	Dokumentationsplan	12
4	RICHTLINIEN FÜR DIE ENTWICKLUNG	13
4.1	Konfigurationsmanagement	13
4.2	Entwicklungsrichtlinien	14
4.3	Prüfungen (Review und Tests)	14
4.4	Einsatz von Werkzeugen	14
5	ENTWICKLUNGSPLAN	15
5.1	Projektstrukturplan (Arbeitsgliederung)	15

5.2	Abhängigkeiten.....	15
5.3	Ressourcen Anforderungen.....	15
5.4	Kosten- und Ressourcenverteilung	15
5.5	Terminplan	15
A	REFERENZEN	16

Abbildungsverzeichnis

Abbildung 1: V-Modell.....	8
Abbildung 2: Organigramm.....	9

Tabellenverzeichnis

Tabelle 1: Deliverables	6
Tabelle 2: Wichtige Arbeitspakete.....	9
Tabelle 3: Projektzuständigkeiten.....	10

Kapitel 1 – Einleitung

1 Einleitung

1.1 Zweck

Dieser Projektplan ist das zentrale Dokument für das Projekt Management im Projekt *JQuotes*.

1.2 Projektüberblick

Dieses Projekt wird als Studienarbeit an der HTW Chur im 4. Semester durchgeführt. Begleitend dazu dient das Fach Software Engineering, welches die Grundlagen für die Planung liefert.

JQuotes ermöglicht die Darstellung von Börsenkursen. Es soll Privatanlegern erlauben, die eigenen Aktienkurse mit einer kurzen Verzögerung zum aktuellen Kurs abzurufen. Dem Benutzer stehen verschiedene Optionen zur Darstellung der Kurse zur Verfügung. Weiter besteht die Möglichkeit, ein Portfolio zu erstellen und unterschiedliche Kurse miteinander zu vergleichen.

Als Datenlieferant wird in erster Linie eine Schnittstelle von Yahoo verwendet, das Programm soll jedoch so ausgelegt werden, dass Erweiterungen und alternative Datenquellen einfach eingebunden werden können.

1.3 Lieferumfang

Der Projektfortschritt wird durch folgende Lieferungen dokumentiert.

Tabelle 1: Deliverables

Lieferung	Termin	Form
Projektplan	2008-03-30	Papier, Claroline
Anforderungsdokumentation	2008-04-04	Papier, Claroline
Entwurfsdokumentation	2008-04-26	Papier, Claroline
Systemvorschriften	2008-04-26	Papier, Claroline
Lauffähiges Produkt	2008-07-01	CD

1.4 Abnahmeprozedere

Da *JQuotes* keinen wirklichen Auftraggeber hat, sieht das Abnahmeprozedere wie folgt aus:

Sämtliche Deliverables werden termingerecht und formal korrekt bei Herrn R. Bonderer eingereicht. Die fertige Software muss die Systemtests bestanden haben um akzeptiert zu werden. Das gesamte Produkt wird bei der Präsentation vorgestellt.

1.5 Definitionen, Akronyme und Abkürzungen

Aktie

Eine Urkunde, welche dem Inhaber einen Anteil an dem Aktienkapital einer Unternehmung bescheinigt.

Kapitel 1 – Einleitung

Devisenkurs

Ist eine Angabe zu welchem Preis eine bestimmte Währung gehandelt wird. Der Devisenkurs für den Ankauf einer Währung aus der Sicht der Bank ist immer höher als jener für den Verkauf.

Eröffnungskurs

Der erste Kurs des Tages nach Eröffnung des Handels.

Japanische Candlesticks

Eine Möglichkeit die Entwicklung von Kursen darzustellen. Ein Candlestick deckt je nach Auflösung ein gewisses Intervall ab und enthält folgende vier Informationen:

- Anfangskurskurs
- Schlusskurs
- Höchstkurs
- Tiefstkurs

Option

Eine Option beinhaltet für den Inhaber das Recht, nicht aber die Verpflichtung einen bestimmten Basiswert, innerhalb einer bestimmten Frist, zu einem im Voraus festgelegten Preis, zu kaufen (Call- Option) oder zu verkaufen (Put-Option).

Portfolio

Zusammensetzung einer Kapitalanlage, zum Beispiel eine Kapitalanlage mit mehreren Aktien.

Schlusskurs

Der letzte Kurs des Tages.

Swiss Market Index (SMI)

Der SMI ist ein Index. Der SMI fasst die Kursentwicklung der 20 liquidesten und grössten Aktientitel der Schweiz zusammen.

[1], [2]

1.6 Referenzen

siehe Anhang A auf Seite 16 dieses Dokumentes.

Kapitel 2 – Projektorganisation

2 Projektorganisation

2.1 Vorgehen

2.1.1 Vorgehensmodell

Als Vorgehensmodell wurde für das Projekt das V-Modell gewählt. Ausschlaggebend dafür war die einfache Handhabung bei der Softwareentwicklung.

Begonnen wird das Projekt mit der Analyse. Mit den daraus erhaltenen Erkenntnissen werden die beiden Design-Stufen durchlaufen. Parallel zu diesen werden die entsprechenden Testvorschriften geplant. Nach Abschluss dieser Schritte wird mit der Implementation begonnen, dem Erstellen der Software. Danach geht es über zu den Tests. Wird ein gewünschtes Testresultat nicht erreicht, geht man über „verifizieren“ wieder zurück zum jeweiligen Planungspunkt.

Wurden die Unit-, und Systemtests mit Erfolg durchgeführt, steht nur noch der Abnahmetest bevor. Durch die Validierung wird der dokumentierte Beweis erbracht, dass das System die vorher spezifizierten Anforderungen im praktischen Einsatz erfüllt.

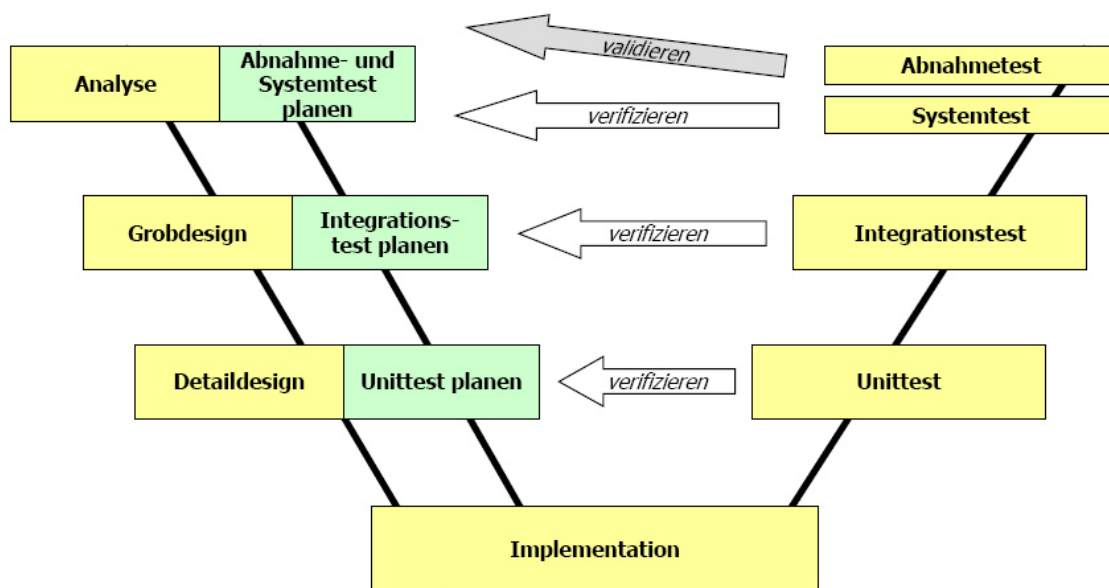


Abbildung 1: V-Modell

[3]

Kapitel 2 – Projektorganisation

2.1.2 Wichtige Arbeitspakete

Tabelle 2: Wichtige Arbeitspakete

Bezeichnung des Resultats	Datum der Fertigstellung	Unter Änderungs-Kontrolle	An Kunden auszuliefern	Verantwortlicher
Projektplan	2008-03-27	Ja	Ja	Projektleiter
Projektplan überarbeitet	2008-05-08	Ja	Ja	Projektleiter
Anforderungsdokumentation	2008-04.03	Ja	Ja	Leiter Qualitätsmanagement
Anforderungsdokumentation überarbeitet	2008-05.01	Ja	Ja	Leiter Qualitätsmanagement
Entwurfsdokumentation	2008-04-23	Ja	Ja	Leiter Core Developer
Entwurfsdokumentation überarbeitet	2008-05-22	Ja	Ja	Leiter Core Developer
Systemvorschriften	2008-04-23	Ja	Ja	Leiter User Interface Design
Systemvorschriften überarbeitet	2008-05-22	Ja	Ja	Leiter User Interface Design
Produkt - Abnahmetest	2008-05-29	Nein	Ja	Leiter Testing
Präsentation	2008-05-22	Nein	Nein	Leiter Qualitätsmanagement
Präsentation Produkt	2008-06-02	Nein	Nein	Projektleiter

2.2 Organisationsstruktur

Die Organisation ist wie folgt strukturiert:

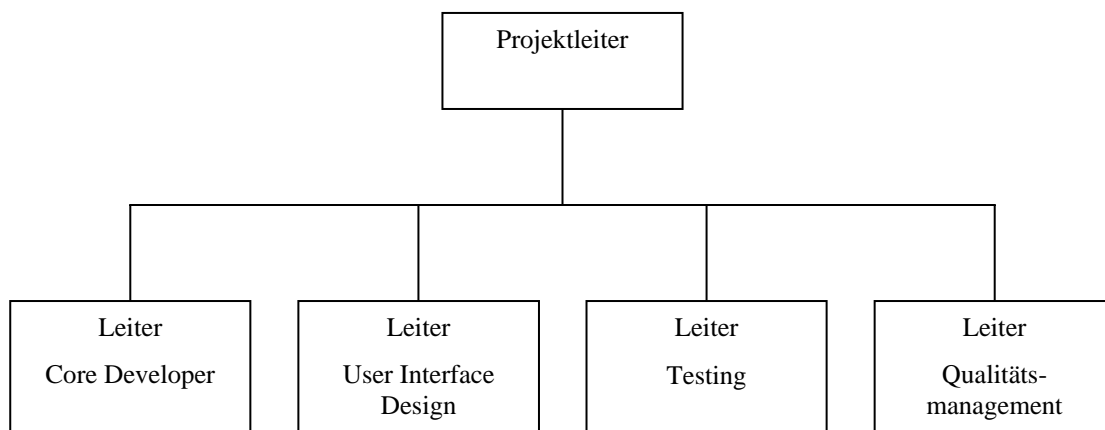


Abbildung 2: Organigramm

Kapitel 2 – Projektorganisation

2.3 Organisatorische Abgrenzung und Schnittstellen

Schnittstelle nach aussen und Kunde ist Herr Prof. Dipl. El.- Ing. ETH R. Bonderer.

2.4 Projektzuständigkeiten

:

Tabelle 3: Projektzuständigkeiten

Rolle	Name	Zuständigkeiten
Projektleiter	Fabian Anderegg	Projektplan, Konfigurationsmanagementplan, Planung
Leiter Core Developer	Norman Süsstrunk	Entwicklung backend, Dokumentation
Leiter User Interface Design	David Angleitner	GUI, SourceForge, Dokumentation
Leiter Testing	Ralph Potztal	Testen Applikation, Qualitätssicherung
Leiter Qualitätsmanagement	Matthias Kohler	Anforderungsdokumentation, Börsenübersicht, Projektplan

Kapitel 3 – Management – Prozess

3 Management – Prozess

3.1 Managementziele und Prioritäten

Erste Priorität haben die Einhaltung der Zeitvorgaben und ein lauffähiges Programm.

Das Projektmanagement stellt sicher, dass die Ressourcen richtig geplant, eingesetzt und überprüft werden. Dadurch soll sichergestellt werden, dass die Ziele erreicht werden können.

Es ist uns aber auch ein Anliegen, dass der Aufwand für Planung und Management in einem angemessenen Verhältnis zur Projektgrösse bleibt und das Programm vor lauter Planung nicht mehr erstellt wird.

3.2 Annahmen und Einschränkungen

Es stehen fünf Studenten mit je 60 Stunden zur Verfügung, es besteht eine zeitliche Einschränkung von insgesamt 300 Arbeitstunden. Die vorgegebenen Deadlines für die Deliverables sind zwingend einzuhalten.

Finanzielle Mittel stehen keine zur Verfügung.

3.3 Risikomanagement

Die Daten für JQuotes werden über eine Schnittstelle von Yahoo bezogen. Würde diese Schnittstelle eingestellt werden, müssten die Daten von einem anderen Lieferanten bezogen werden. Deshalb wird die ganze Software so erstellt, dass ein Austausch des Datenlieferanten mit sehr geringem Aufwand vollzogen werden kann.

3.4 Berichtswesen

Das Ziel ist es, wöchentlich eine Teamsitzung durchzuführen. Der Projektleiter leitet die Sitzung. Am Schluss der Sitzung wird Ort und Zeit des nächsten Treffens vereinbart.

Der Projektleiter informiert sich bei allen Teammitgliedern, wie weit sie mit ihren Teilaufgaben sind und legt zusammen mit dem Team fest, wie weiter vorgegangen wird.

Zu jeder Sitzung wird ein kurzes Protokoll verfasst. Dieses enthält in erster Linie die folgenden zwei Informationen:

1. Standortbestimmung (wie weit sind wir mit den einzelnen Teilaufgaben).
2. Aufträge an die einzelnen Teammitglieder.

Das Protokoll wird innerhalb von 24 Stunden an alle Teammitglieder per E-mail verschickt. Somit wird sichergestellt, dass alle Teammitglieder jederzeit über den Fortschritt des Projekts informiert sind und jeder kann im Protokoll nachlesen, was er bis wann zu erledigen hat.

3.5 Personalplanung

Jeder der fünf Studenten im Team ist für einen Teilbereich verantwortlich. Mit einer guten Personalplanung soll erreicht werden, dass jedes Teammitglied immer optimal ausgelastet ist und keine Leerläufe entstehen.

Fabian Anderegg ist Projektleiter, Norman Süssstrunk und David Angleitner sind für das Programmieren der Software zuständig, Ralph Potzital schreibt die Sitzungsprotokolle und ist in einer späteren Phase für das Testen der Software zuständig und Matthias Kohler definiert die Anforderungen.

Kapitel 1 –

Jeder dokumentiert den Teil des Projektes in seinem Verantwortungsbereich. Dokumente die nicht eindeutig zugeordnet werden können werden gesondert zugeteilt.

3.6 Dokumentationsplan

Folgende Dokumente werden erstellt:

- Gantt-Diagramm
- Deliverables (gemäss Vorgabe)
- Sitzungsprotokolle
- Benutzerhandbuch oder gegebenenfalls Handhabungsvideo

Kapitel 4 – Richtlinien für die Entwicklung

4 Richtlinien für die Entwicklung

4.1 Konfigurationsmanagement

In den Verfahren des Konfigurationsmanagement wird festgelegt, wie vorgeschlagene Systemänderungen festgehalten und bearbeitet werden.

Das Konfigurationsmanagement beinhaltet folgende 4 Bereiche:

4.1.1 Changemanagement

Während der Entwicklung sind Änderungen am Code an der Tagesordnung. Vorwiegend treten Fehlerkorrekturen ("Bugfixes") bei entdeckten Softwarefehlern sowie neue oder geänderte Anforderungen an die Software auf. Der Prozess des Changemanagement setzt jeweils bei einem neuen Release der Software ein. Der Code wird getestet und allfällig auftretende Änderungen werden an das Changemanagement übergeben. Die erste Stufe im Ablauf ist das Ausfüllen eines Änderungsantrags, in welchem die gewünschten Änderungen beschrieben werden. Der Antrag kann Vorschläge für die Änderungen sowie Priorität der Änderung beinhalten. Der Antrag wird überprüft und kann dann angenommen oder abgelehnt werden. Wurde eine Reihe von Änderungen genehmigt, wird die Software an die Entwickler übergeben und der Zyklus beginnt von neuem. Als zentrales Tool für diesen Vorgang wird Bugzilla eingesetzt. Es ist jedoch noch abzuklären, ob Bugzilla als Bestandteil von Sourceforge eingesetzt werden kann.

4.1.2 Versions- und Releasemanagement

Beim Versionsmanagement handelt es sich um den Prozess der Bezeichnungen und Verfolgung verschiedener Softwareversionen und -Releases. Das Versionsmanagement stellt in erster Linie sicher, dass, wenn nötig, verschiedene Versionen eines Systems wiederhergestellt werden können und nicht versehentlich geändert werden. Als Versionsverwaltungssystem für diesen Prozess wird CVS eingesetzt.

Releases sind Versionen, welche an den Kunden ausgeliefert werden. Unterschiedliche Versionen können intern erzeugt werden, um beispielsweise verschieden Varianten zu testen. Diese Versionen bleiben intern und müssen für den Kunden nicht sichtbar sein.

4.1.3 Buildmanagement (Systemerstellung)

Das Buildmanagement regelt das Kompilieren und die Konfiguration der Softwarekomponenten. Es überprüft zuerst, ob alle nötigen Werkzeuge und Komponenten (Compiler, benötigte Libraries etc.) installiert sind. Erst dann wird der Quellcode übersetzt und lauffähige Komponenten werden erzeugt. Für diesen Prozess wird das CASE-Tool Ant eingesetzt, welches den gesamten beschriebenen Prozess automatisiert.

Kapitel 1 –

4.2 Entwicklungsrichtlinien**4.2.1 Coding-Standards**

Es werden die Coding-Standards verwendet, wie sie von Sun empfohlen werden. Es wird auf das externe PDF-File CodeConventions_Java.pdf verwiesen. [4]

4.2.2 Dokumentations Standards

Für die Dokumentation des Quellcodes wird die Javadoc-Schreibweise verwendet. Mittels dem im Java SDK integrierten Tool javadoc wird anschliessend eine HTML-Dokumentation des Quellcodes erzeugt.

4.3 Prüfungen (Review und Tests)

Die erstellten Programmkomponenten werden automatisiert mittels JUnit sowie durch Systemtestpersonen selbst getestet.

4.4 Einsatz von Werkzeugen

Als Programmiersprache wird das Java SDK 1.6 eingesetzt. Als IDE kommt Eclipse 3.3 zum Einsatz. Die jeweiligen Tools für das Konfigurationsmanagement sind in den jeweiligen Abschnitten erwähnt und alle vollständig in Eclipse integriert.

Für das Debugging kommt einerseits der in Eclipse integrierte Debugger zum Einsatz. Ergänzend dazu wird noch das Framework log4j eingesetzt. log4j ist ein Framework zum Loggen von Anwendungsmeldungen in Java und ermöglicht eine sehr detaillierte Ausgabesteuerung.

5 Entwicklungsplan

5.1 Projektstrukturplan (Arbeitsgliederung)

Die Struktur des Projektes kann aus dem Terminplan herausgelesen werden.

5.2 Abhängigkeiten

Die Abhängigkeit der verschiedenen Arbeitsschritte ist aus dem Terminplan ersichtlich.

5.3 Ressourcen Anforderungen

Benötigte Hardware:

Die vom Studiengang vorgeschriebenen Notebooks sind die einzigen hardwaremässigen Ressourcen.

Benötigte Software:

Siehe 4.4 Einsatz von Werkzeugen und 4.12 Version- und Releasemanagement

5.4 Kosten- und Ressourcenverteilung

Dies ist eine Studienarbeit der HTW Chur und ist deshalb nicht mit Kosten behaftet.

5.5 Terminplan

Die Terminplanung kann aus folgendem Dokument entnommen werden: Plaunug_V1.0.mpp

Anhang A – Referenzen

A Referenzen

- [1] <http://www.derivate.n-tv.de/showpage.asp?pageid=109>
- [2] <http://www.varengold.de/glossar.ssi>
- [3] Vorlesung Vorgehensmodelle in der Softwareentwicklung, Reto Bonderer
- [4] <http://java.sun.com/docs/codeconv/html/CodeConvTOC.doc.html>