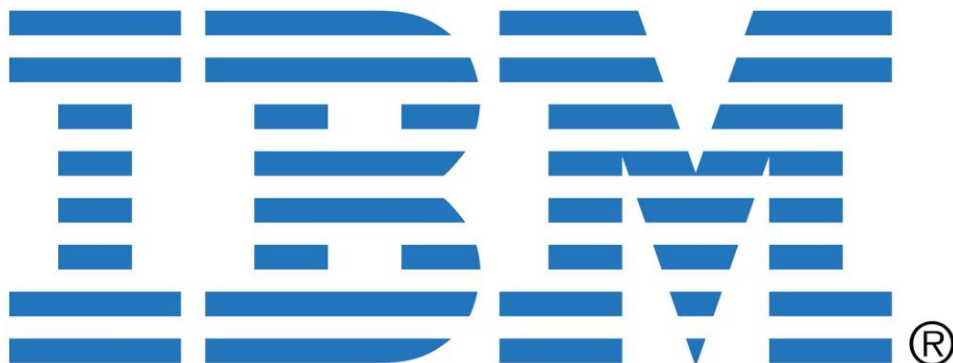


Pflichtenheft WebEst

*Portierung des IBM Software Estimation Tool
als interaktive Webanwendung*



Ansprechpartner:

Andre Munzinger (a.munzinger@stud.tu-darmstadt.de)
Florian Friedrichs (friedrichsflorian@googlemail.com)
Dr. Holger Kremmin (holger.kremmin@de.ibm.com)

Stand: Freitag, 10. Dezember 2010

Inhaltsverzeichnis

1	Einleitung.....	1
2	Hintergrund.....	2
3	Ist-Zustandsanalyse	3
3.1	Darstellung SLIM Estimator	4
3.2	Darstellung der Excel Sheets	5
4	Zielbestimmung	7
4.1	Allgemeine Anforderungen	7
4.1.1	Technische Architektur	7
4.1.2	Anwendungsarchitektur	7
4.1.3	Datenzugriff	7
4.1.4	Strikte Service Orientierung	8
4.2	Muss-Kriterien.....	8
4.2.1	Benutzerauthentifizierung.....	8
4.2.2	Templates.....	9
4.2.3	Solutions.....	9
4.2.4	Constraints.....	10
4.2.5	Reporting.....	10
4.2.6	Dokumentation/Hilfe	11
4.2.7	Logging	11
4.2.8	Property-Files	11
4.2.9	PI-History	11
4.2.10	Size Calculator.....	11
4.2.11	WebServices.....	11
4.3	Wunsch-Kriterien.....	11
4.3.1	Erweiterbarkeit des Reporting	11
4.3.2	Exportierung von Templates als XML-Datei unterstützen	13
4.3.3	Log Level Rotation	13
4.4	Abgrenzungskriterien.....	13
4.4.1	Sicherheitsbestimmungen für Web Services	13
4.4.2	Internationalisierung	13

4.4.3	Benutzerverwaltung	13
4.4.4	Template Verwaltung	13
4.4.5	Import/Export von WebEst in SLIM	13
4.4.6	Versionierung	13
4.4.7	Milestone Calculator	13
4.4.8	Such-/Filtermöglichkeit	13
4.4.9	Defect Categories	14
5	Produkteinsatz	15
5.1	Anwendungsbereiche	15
5.2	Zielgruppen	15
5.3	Betriebsbedingungen	15
6	Technische Produktumgebung	16
6.1	Software	16
6.1.1	Server	16
6.1.2	Client	16
6.2	Hardware	16
7	Produktfunktionen	16
7.1	Übersicht der Prozesse	16
7.2	Detaillierte Use-Cases	18
8	Produktdaten	24
8.1	Beschreibung der langfristig gespeicherten Daten aus Nutzersicht	24
9	Leistungen	28
9.1	Zeitbezogen	28
9.2	Umfangsbezogen	28
10	Qualitätsanforderungen	29
11	Benutzungsoberfläche	31
11.1	Allgemeine Anforderungen	31
11.2	Aufteilung der Arbeitsfläche	31
12	Testen	33
13	Entwicklungsumgebung	33
13.1	Software	33
13.2	Hardware	34
13.3	Orgware	34

13.4	Entwicklungsschnittstellen.....	34
14	Ergänzungen.....	35
14.1	Projektablauf	35
14.2	Lizensierung	35
15	Anhang.....	36
15.1	A1.....	36

Abbildungsverzeichnis

Abbildung 1: Grafische Darstellung des Schätzungsprozess	3
Abbildung 2: Screenshot aus Excel-Datei zur Berechnung der Use-Case Granularität	5
Abbildung 3: Grobkonzept der technischen Architektur	7
Abbildung 4: Darstellung der einzelnen Benutzerrollen.....	8
Abbildung 5: Screenshot aus QSM SLIM-Estimator Management-Report.....	10
Abbildung 6: Screenshot aus SLIM-Estimator Reportvergleich (Projektlaufzeit)	12
Abbildung 7: Screenshot aus SLIM-Estimator Reportvergleich (Projektaufwand)	12
Abbildung 8: Prozessübersicht bzw. Use-Case Übersicht	17
Abbildung 9: Mögliche Darstellung des „Welcome“-Screens.....	31
Abbildung 10: Mögliche Darstellung des Estimationprozesses	32
Abbildung 11: Grobkonzept des Projektablaufs.....	35

Tabellenverzeichnis

Tabelle 1: Detaillierte Darstellung aller angebotenen Funktionen und deren Zugangsberechtigungen.....	9
Tabelle 2: Use-Case für Benutzer Login	18
Tabelle 3: Use-Case für Erstellung eines neuen Estimates	18
Tabelle 4: Use-Case zur Veränderung eines Estimates	19
Tabelle 5: Use-Case zum Vervielfältigen von einem bestehenden Estimate	19
Tabelle 6: Use-Case zum Löschen eines Estimates	20
Tabelle 7: Use-Case zum Anzeigen eines Estimates	20
Tabelle 8: Use-Case zum Erstellen einer Solution	21
Tabelle 9: Use-Case zum Verändern einer Solution	21
Tabelle 10: Use-Case zum Vervielfältigen einer Solution.....	22
Tabelle 11: Use-Case zum Löschen einer Solution	22
Tabelle 12: Use-Case zum Anzeigen einer Solution	23
Tabelle 13: Use-Case zum Anzeigen von einem Report	23
Tabelle 14: Datenbanktabelle für Benutzer	24
Tabelle 15: Datenbanktabelle für Estimates	25
Tabelle 16: Datenbanktabelle für Phasen.....	25
Tabelle 17: Datenbanktabelle für Meilensteine	26
Tabelle 18: Datenbanktabelle für Solutions	26
Tabelle 19: Datenbanktabelle für Use Cases	26
Tabelle 20: Datenbanktabelle für Templates	27
Tabelle 21: Verteilung von Qualitätsanforderungen an die einzelnen Softwareattribute	29

1 Einleitung

Ziel des Projekts ist es, ein bereits beim Kunden IBM Global Business Services im Einsatz befindliches Softwareschätzungstool als Webanwendung zu portieren. Im Folgenden wird die zu erstellende Anwendung WebEst genannt.

In diesem Dokument wird zuerst die aktuelle Ist-Situation analysiert. Im Anschluss werden die für beide Parteien zur Abnahme des Projekts notwendige Verpflichtungen festgehalten.

Zudem werden potentielle Erweiterungs- und Entwicklungsmöglichkeiten im Rahmen der Migration von WebEst in Kapitel 4.4 diskutiert.

2 Hintergrund

Im Rahmen des Wirtschaftsinformatik-Praktikums an der Technischen Universität in Darmstadt soll in Kooperation mit dem Business Partner IBM Global Business Services eine auf Webtechnologien basierte Anwendung zum Erstellen von Softwareschätzungen entwickelt werden. Das Projekt umfasst den gesamten Lebenszyklus eines typischen Softwareprojektes.

Folglich stellt die Migration des zu entwickelnden Tools einen Fortschritt im Bereich Business Intelligence dar.

Die Estimates dienen dem Unternehmen bei der Planung sowie der Kontrolle von Softwareentwicklungsprojekten. Außerdem kann so die Wettbewerbsfähigkeit weiter optimiert werden. Ein Estimate wird nach dem Putnam-Modell errechnet:

$$Productivity\ Index = \frac{Size}{Effort^{\frac{1}{3}} \times Duration^{\frac{4}{3}}}$$

Formel 1: Productivity Index-Berechnung im Putnam-Modell

Hierbei wird Size anhand des UseCasePoint-Sizings in einem Excelprogramm ermittelt. Effort und Duration sind die Rahmenbedingungen denen das zu planende Projekt unterliegt.

Hauptziel ist eine Prognose bezüglich Zeit und Ressourcen abzugeben und daraus die resultierenden Kosten zu ermitteln. Während dieses Prozesses werden Erfahrungswerte generiert die für spätere Schätzungen und Projekte eingesetzt werden können. Das ermöglicht es dem Unternehmen den Softwareerstellungsprozess kritisch im Hinblick auf Wirtschaftlichkeit zu reflektieren sowie den Prozessen der Konkurrenzunternehmen gegenüberzustellen und anschließend die eigenen Prozesse zu optimieren.

Bevor ein Softwareprojekt geplant wird muss zuerst eine Machbarkeitsanalyse stattfinden. Bereits in diesem Schritt kann eine Softwareschätzung erste Fehlerquellen aufzeigen. Später liefert die Schätzung wertvolle Informationen für die Projektplanung sowie das Projektcontrolling.

Die Auslagerung der Software ins Web ermöglicht benutzerfreundlichen Zugriff auf die zentralen Erfahrungswerte. Weiter kann jeder Mitarbeiter überall auf der Welt plattformunabhängig auf die Schätzungssoftware und seine Schätzungen zugreifen.

3 Ist-Zustandsanalyse

Aktuell besteht der Software-Schätzungsprozess aus zwei Hauptkomponenten.

Die erste Komponente ist die kostenpflichtige Anwendung "SLIM-Estimation" vom Drittanbieter Quantitative Software Management Inc. Die Desktopanwendung ist eine Standalone-Lösung die mit Hilfe der in der zweiten Komponente berechneten Parameter das zu planende Softwareprojekt schätzt. Der Benutzer wird mittels eines Assistenten durch die verschiedenen Einstellungsmöglichkeiten geleitet.

Komponente Zwei ist eine eigene, in Excel entwickelte Lösung der IBM Global Business Services. Mit Hilfe von Visual Basic und der normalen Excel-Funktionalitäten werden anhand der Erfahrungen des Unternehmens die für die Schätzung benötigten Parameter errechnet.

Im Folgenden soll der detaillierte Schätzungsprozess genauer dargestellt werden.

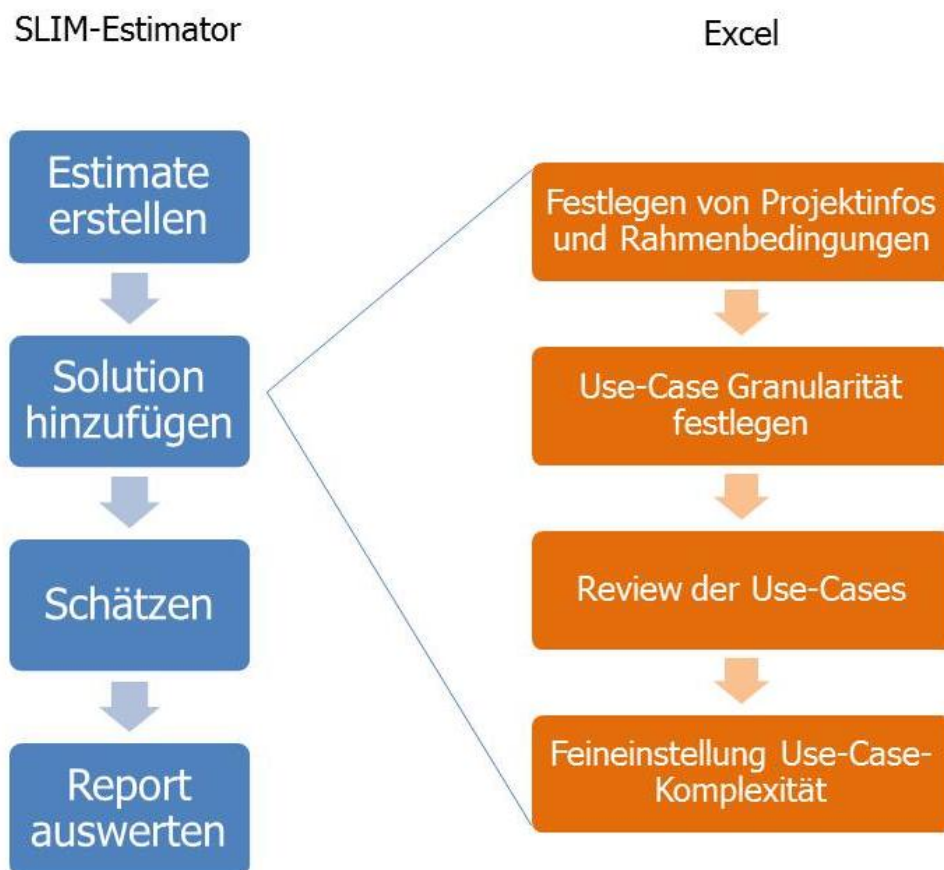


Abbildung 1: Grafische Darstellung des Schätzungsprozess

3.1 Darstellung SLIM Estimator

Schritt 1: "Estimate erstellen"

Im ersten Schritt wird das Projektumfeld erfasst. Der Benutzer kann hier die Daten unter verschiedenen Reitern eintragen. Hierbei werden Daten wie Projektname, Organisation, Abteilung, Name des Estimators sowie das Land festgelegt. Weiter werden der Anwendungstyp, Industriesektor und die Maßeinheiten eingestellt. Der Benutzer hat außerdem die Möglichkeit eine Projektbeschreibung zu hinterlegen.

In den folgenden Reitern werden für den Projekttyp typische Parameter aus den verfügbaren Templates¹ geladen. Sie beinhalten Angaben zur Projektlaufzeit, Defect Categories, Projektphasen und Milestones. Der Benutzer hat jederzeit die Möglichkeit diese Werte anzupassen.

Schritt 2: "Solution hinzufügen"

Im zweiten Schritt wird der Estimation eine Solution hinzugefügt. Jedes Projekt kann mehrere Solutions besitzen. Die Solutions beschreiben verschiedene Lösungsausprägungen des zu planenden Projekts.

Der Benutzer hat außerdem die Möglichkeit die gewünschte Projektlaufzeit einzutragen. Diese wird dann mit der geschätzten Zeit verglichen. Mit der „Calculate“-Schaltfläche wird anhand des Putnam-Modells der notwendige Aufwand berechnet.

Der Size-Calculator berechnet anhand von Komponente 2 den Total UUCP. Im Anschluss daran wird die Schätzung erstellt.

3. Schritt: „Schätzen“

Die Schätzung erfolgt unter der Zuhilfenahme der Putnam Gleichung. Diese verwendet die zuvor ermittelten UUCP als Größe für den erwarteten Projektaufwand in SLOC und berechnet unter der Verwendung von verschiedenen Parametern eine Schätzung in Form von erwarteten Aufwand und Dauer.

4. Schritt: "Report auswerten"

Als Resultat der Schätzung wird ein Report generiert. Er enthält die geschätzten Randbedingungen, der Aufwand und die erwartete Zeit für das Projekt. Eine genauere Darstellung des umzusetzenden Reports findet sich in Kapitel 4.2.5 sowie 4.3.1.

Das Reporting dient als Instrument zur grafischen Illustration der Informationen und stellt folglich eine Grundlage zur Entscheidungsfindung dar.

¹ Eine genauere Erläuterung eines Templates ist unter Kap. 4.2.2 zu finden.

3.2 Darstellung der Excel Sheets

Schritt 1: "Projektinfos und Rahmenbedingungen festlegen"

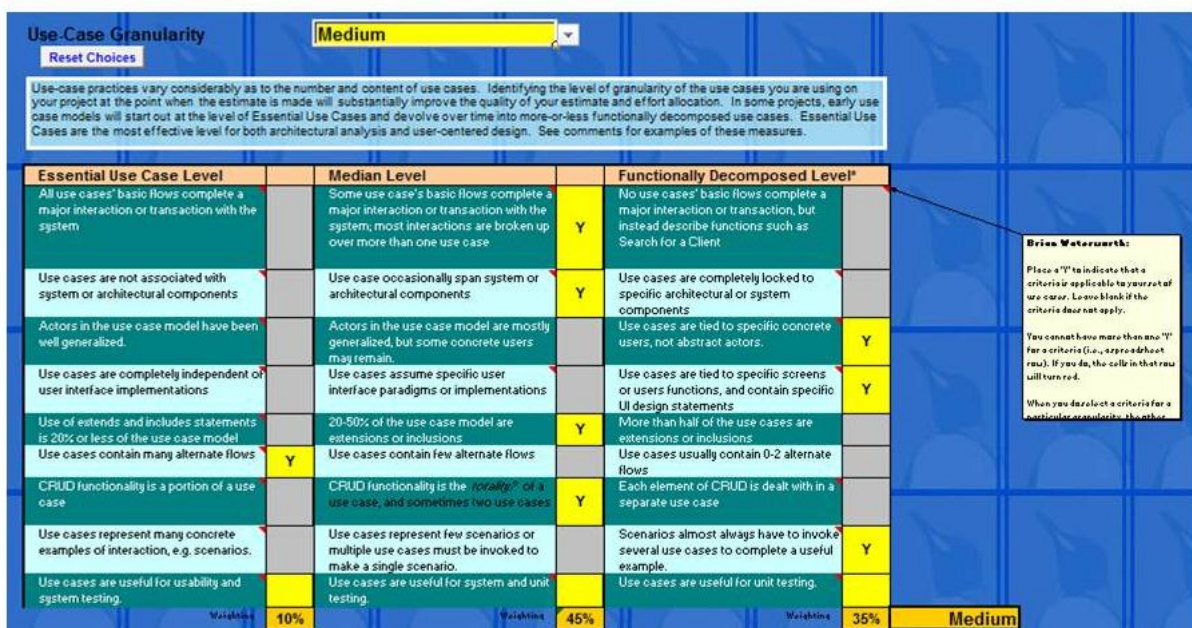
Zunächst kann der Nutzer Projektinformationen zu einem neuen oder bereits bestehenden Projekt hinzufügen oder diese ändern. Darüber hinaus besteht die Möglichkeit durchgeführte Änderungen an einer bereits bestehenden Schätzung in einem "History"-Feld zu speichern, um weiteren Benutzern durchgeführte Änderungen an dem Projekt aufzeigen zu können.

Schritt 2: "Use-Case Granularität festlegen"

Anhand des unten dargestellten Fragebogens wird die Granularität der einzelnen Use-Cases ermittelt. Als Granularität bezeichnet man die Feinheitstufe der Gliederung mit der Funktionen und Aktivitäten in Bezug auf die gesamte Aktion unterteilt sind.

Hinweis: Unterteilung der Granularität in:

- feingranular
- mittelgranular
- und grobgranular statt.



Use-Case Granularity Medium

[Reset Choices](#)

Use-case practices vary considerably as to the number and content of use cases. Identifying the level of granularity of the use cases you are using on your project at the point when the estimate is made will substantially improve the quality of your estimate and effort allocation. In some projects, early use case models will start out at the level of Essential Use Cases and devolve over time into more-or-less functionally decomposed use cases. Essential Use Cases are the most effective level for both architectural analysis and user-centered design. See comments for examples of these measures.

Essential Use Case Level	Median Level	Functionally Decomposed Level ^a
All use cases' basic flows complete a major interaction or transaction with the system	Some use cases' basic flows complete a major interaction or transaction with the system; most interactions are broken up over more than one use case	No use cases' basic flows complete a major interaction or transaction, but instead describe functions such as: Search for a Client
Use cases are not associated with system or architectural components	Use case occasionally span system or architectural components	Use cases are completely locked to specific architectural or system components
Actors in the use case model have been well generalized.	Actors in the use case model are mostly generalized, but some concrete users may remain.	Use cases are tied to specific concrete users, not abstract actors.
Use cases are completely independent of user interface implementations	Use cases assume specific user interface paradigms or implementations	Use cases are tied to specific screens or users functions, and contain specific UI design statements
Use of extends and includes statements is 20% or less of the use case model	20-50% of the use case model are extensions or inclusions	More than half of the use cases are extensions or inclusions
Use cases contain many alternate flows	Use cases contain few alternate flows	Use cases usually contain 0-2 alternate flows
CRUD functionality is a portion of a use case	CRUD functionality is the <i>core</i> ^b of a use case, and sometimes two use cases	Each element of CRUD is dealt with in a separate use case
Use cases represent many concrete examples of interaction, e.g. scenarios.	Use cases represent few scenarios or multiple use cases must be invoked to make a single scenario.	Scenarios almost always have to invoke several use cases to complete a useful example.
Use cases are useful for usability and system testing.	Use cases are useful for system and unit testing.	Use cases are useful for unit testing.
Weighting: 10%	Weighting: 45%	Weighting: 35%
Medium		

Brian Waterworth:
Place a "Y" to indicate that a criterion is applicable to your set of use cases. Leave blank if the criterion does not apply.
You cannot have more than one "Y" for a criterion (i.e., you pick one row). If you do, the cells in that row will turn red.
When you deselect a criterion for a particular scenario, the other

Abbildung 2: Screenshot aus Excel-Datei zur Berechnung der Use-Case Granularität

Schritt 3: "Review der Granularität anhand von Guidelines"

Review der gelisteten Use-Cases anhand im Anhang befindlicher Tabelle. Im Anschluss werden die Use-Cases nach Komplexität, Wiederverwendbarkeit und Ausmaß der Definition sortiert.

Schritt 4: "Feineinstellung der Use-Case-Komplexität"

Auf der letzten Seite der Excel-Anwendung können für jeden Use-Case spezielle Feineinstellungen getroffen werden.

Nachdem alle Einstellungen getätigt wurden, ist der UUCP-Wert auf der Hauptseite der Anwendung zu finden. Dieser wird dann händisch in Komponente 1 übertragen.

4 Zielbestimmung

Im Folgenden werden die Anforderungen an das zu entwickelnde web-basierte Software Estimation Tool dargelegt. Die Umsetzung der Web Applikation aus Microsoft Excel erfolgt inhaltlich identisch. Nach Beschreibung der allgemeinen Anforderungen wird explizit auf die spezifisch gewünschten Anforderungen eingegangen.

4.1 Allgemeine Anforderungen

4.1.1 Technische Architektur

Das Tool wird in der Programmiersprache Java entwickelt. Als Anwendungsserver wird IBM WebSphere Application Server Community Edition 2.1, als Datenbank Apache Derby eingesetzt. Die Entwicklung erfolgt in Eclipse. Die Architektur ist klassisch in die 3 Schichten Model, Logic, Frontend anzulegen.

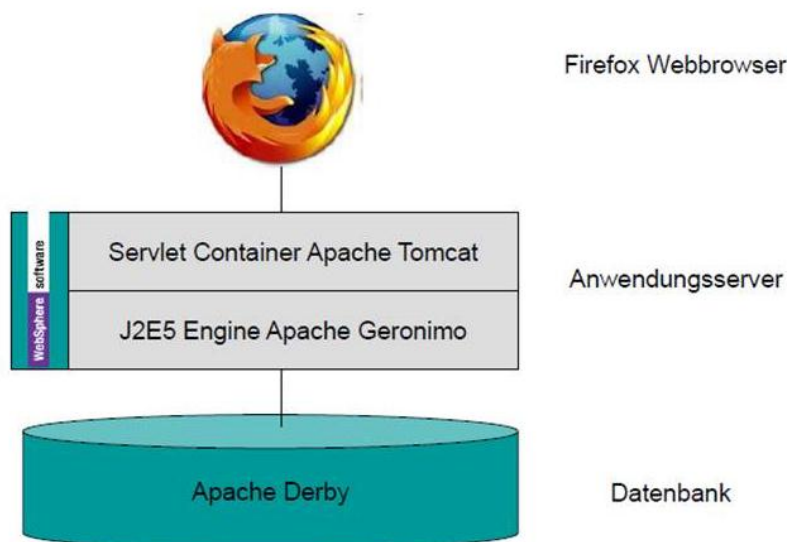


Abbildung 3: Grobkonzept der technischen Architektur

4.1.2 Anwendungsarchitektur

Die Anwendung soll an das MVC Konzept (Model View Controller) angelehnt sein. Dieses Konzept dient als Strukturierungsmuster im Bereich Software Development. Ziel dieses Musters ist ein flexibler Programmentwurf, der eine spätere Modifizierung oder Erweiterung sowie eine einfache Wiederverwendbarkeit ermöglicht. Aus diesem Konzept ergeben sich weitere Implikationen, die nachfolgend dargestellt werden.

4.1.3 Datenzugriff

Generell gilt, dass alle Zugriffe auf die Datenbank gekapselt durch eine Datenzugriffsschicht zu erfolgen haben. Der direkte Zugriff auf Datenbankobjekte durch die Anwendung ist untersagt. Es sind zwei Datenbankschichten zu implementieren.

- Physikalische Datensicht – Objektansicht der Datenbanktabellen.
- Implementierung als EJB 3.0
- Fachliches Datenmodell – rein fachliche Sicht auf die Datenbankobjekte

4.1.4 Strikte Service Orientierung

Alle fachlichen Funktionen sind als Serviceobjekte abzubilden. Es gelten folgende Implementierungsregeln:

- Services werden als Stateless Session Beans gemäß der EJB 3.0 Spezifikation implementiert
- Das Facade Pattern soll sowohl ein Local als auch ein Remote Interface ermöglichen
- Ausgewählte Services werden zusätzlich als Webservices anderen Anwendungen dargeboten
- Fachfunktionen stehen auch AJAX Aufrufen zur Verfügung

4.2 Muss-Kriterien

4.2.1 Benutzerauthentifizierung

Im Hinblick auf das Zugriffsberechtigungssystem sollen drei Rollen definiert werden. In Abbildung 4 ist die detaillierte Gliederung in alle verfügbaren Funktionen und die Zugriffsrechte der verschiedenen Benutzerrollen dargestellt.

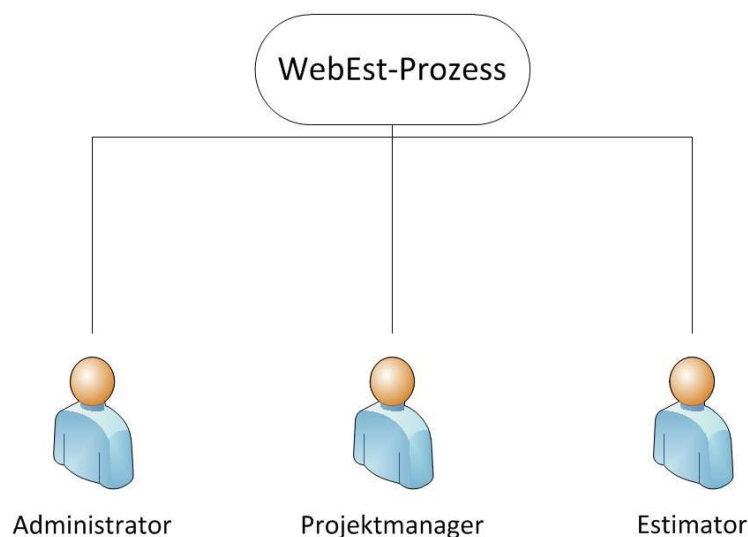


Abbildung 4: Darstellung der einzelnen Benutzerrollen

- **Administrator**

Der Administrator darf alle Estimates sehen und modifizieren. Allgemein darf er alle Aktionen ausführen.

- **Projekt Manager**

Der Projektmanager darf alle Estimates seiner Organisation und Division sehen, jedoch nicht verändern.

- **Estimator**

Der Estimator darf seine eigenen Estimates und die seiner Division sehen, verändern, duplizieren sowie neue anlegen. Er darf jedoch keine Estimates von Anderen sehen.

Der Estimator legt fest zu welcher Organisation und Division das Estimate gehört.

	Admin	PM	Estimator
New Estimate	X		X
Clone Estimate	X		X
Modify Estimate	X		X
Show Estimate	X	X	X
Delete Estimate	X		X
New Solution	X		X
Clone Solution	X		X
Modify Solution	X		X
Delete Solution	X		X
Show Solution	X	X	X
Show Report	X	X	X

Tabelle 1: Detaillierte Darstellung aller angebotenen Funktionen und deren Zugangsberechtigungen

4.2.2 Templates

Templates dienen als vordefinierte Schablonen um gewünschte Default-Parameter eines Estimates zu setzen. Daten für die Templates werden dem Projektteam generiert und direkt in die Datenbank eingefügt.

Eine weitere Ausbaustufe wird in Wunschkriterien erklärt.

Inhalte der Templates sind die Folgenden:

- Days per week
- Hours per day
- Defect Categories (Einbeziehen, Name, Wahrscheinlichkeit)
- Phases (Phasen-Nr., Akronym, Name, Beschreibung)
- Milestones (Einbeziehen, ID, Akronym, Name, Stand der Entwicklung, Phasenzugehörigkeit, Beschreibung)

4.2.3 Solutions

Nach erfolgreicher Anlage eines neuen Estimates sollen diesem Solutions hinzugefügt werden. Diese Solutions dienen dann als Datenbasis für die Erstellung eines Reports.

4.2.4 Constraints

Die Constraints ermöglichen es dem Projekt Rahmenbedingungen zu stellen. Das bedeutet man kann dem Projekt Parameter wie z.B. die maximale Laufzeit sowie die maximale Anzahl an Mitarbeitern die gleichzeitig am Projekt arbeiten können vorgeben kann. Weitere Constraints können in einer der folgenden Ausbaustufen ergänzt werden.

4.2.5 Reporting

Das Reporting dient zur grafischen Visualisierung einer Solution. Jede Solution generiert einen eigenen Report.

Management Report

Diese Art von Report beinhaltet die allgemeinen Felder Eff UUCP, „Gearing“-Faktor, PI sowie folgende Ausprägungen zum gesamten Projekt wie auch den einzelnen Phasen.

- Duration (Months)
- Effort (PDay)
- Average Staff (ppl/month)
- Start Date
- End Date

Weiter sollen in einem frei gestaltbaren Schaubild die Ergebnisse aufgetragen werden. Die Meilensteine des Projekts sollten hier ebenfalls verzeichnet werden. Abbildung 5 stellt eine beispielhafte Umsetzung einer Grafik dar.

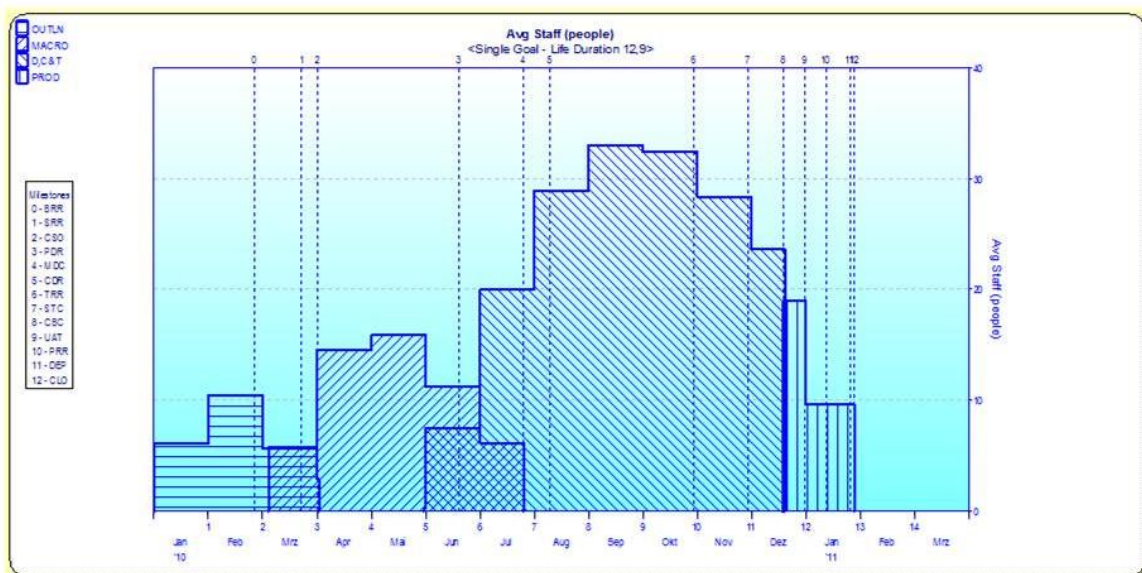


Abbildung 5: Screenshot aus QSM SLIM-Estimator Management-Report

Optionale Erweiterungen sind unter Kapitel 4.3.1 „Erweiterbarkeit des Reporting“ zu finden.

4.2.6 Dokumentation/Hilfe

Die Dokumentation, respektive die Hilfe, werden in Form eines Wikis realisiert. Die Sprache der Dokumentation ist Englisch. Eine Übersetzung kann wenn gewünscht von Kundenseite in Auftrag gegeben werden.

Quellcode-Kommentare werden in Englisch realisiert.

4.2.7 Logging

Die Modifikation, Erstellung sowie Löschung von Estimates soll zukünftig in einer Protokolldatei (logfile) festgehalten werden. Dies ermöglicht Transparenz und Nachvollziehbarkeit. Dabei sollen die verschiedenen Logging-Level wie "Warning", "Error" und "Info" beachtet werden.

4.2.8 Property-Files

Die Auslagerung von Pfaden soll in Property Dateien zu erfolgen. Die genaue Verwendung der Propertyfiles erfolgt im DV-Konzept.

4.2.9 PI-History

Die Productivity-Index-Historie (PI-Historie) stellt eine Entscheidungsgrundlage zur Auswahl des spezifischen PIs einer Solution dar. Eine Liste früherer PIs soll in einer Tabelle nach Projekttypen zusammengefasst werden und deren PI Mittelwert berechnet werden.

4.2.10 Size Calculator

Der Size Calculator soll die Aufwandschätzung eines Projekts in SLOC erlauben. Als Berechnungsgrundlage nutzt der Size Calculator den ermittelten UUCP Wert und einen beliebig wählbaren "Gearing"-Faktor. Aus diesen Werten wird nun die erwartete Projektgröße berechnet.

4.2.11 WebServices

WebServices dienen der Remoteabfrage von WebEst-Daten. Die erste Ausbaustufe umfasst das Abrufen von Reports sowie das Abfragen von Estimates und Solutions. Als Zeichensatz wird UTF-8 verwendet.

4.3 Wunsch-Kriterien

4.3.1 Erweiterbarkeit des Reporting

Durch die Erweiterbarkeit des Reporting soll es möglich sein, Solutions systematisch untereinander in Bezug auf Aufwand und Dauer zu vergleichen und zu analysieren. Weiter soll ein WBS-Report erstellt werden.

Verschiedene Solutions Vergleichen

Hier soll eine Auflistung der Metadaten des Projekts sowie zwei Schaubilder welche den Aufwand und die Dauer der verglichenen Solutions enthalten sein. Ergänzend zu den Schaubildern sollen die Wertetabellen ebenfalls abgebildet werden. In Abbildung 6 und Abbildung 7 ist eine mögliche Darstellung der Schaubilder abgebildet.

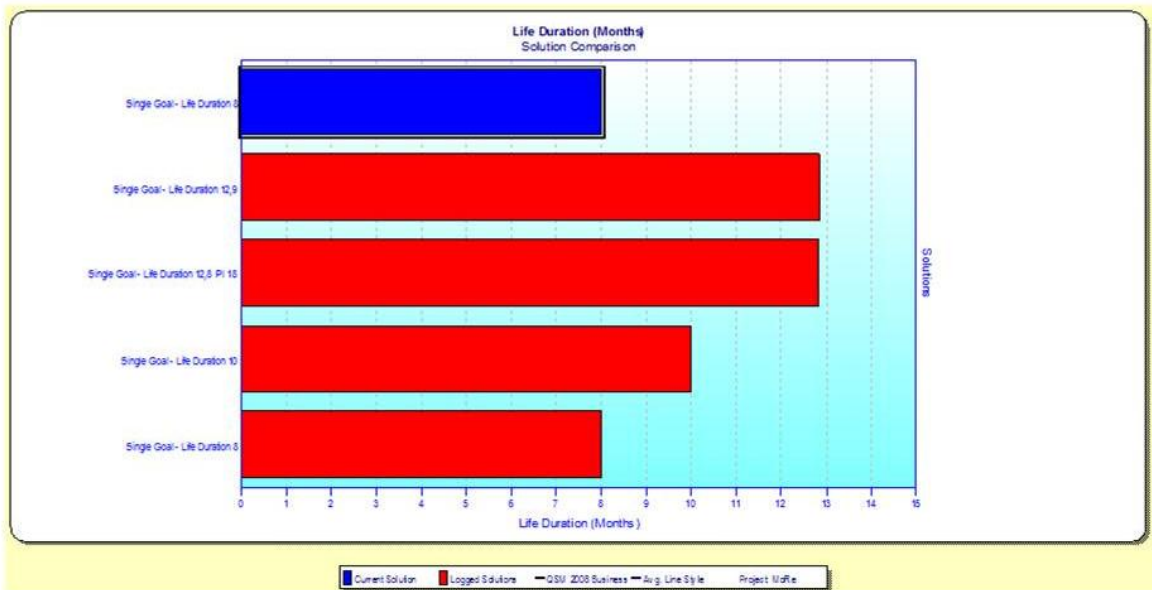


Abbildung 6: Screenshot aus SLIM-Estimator Reportvergleich (Projektlaufzeit)

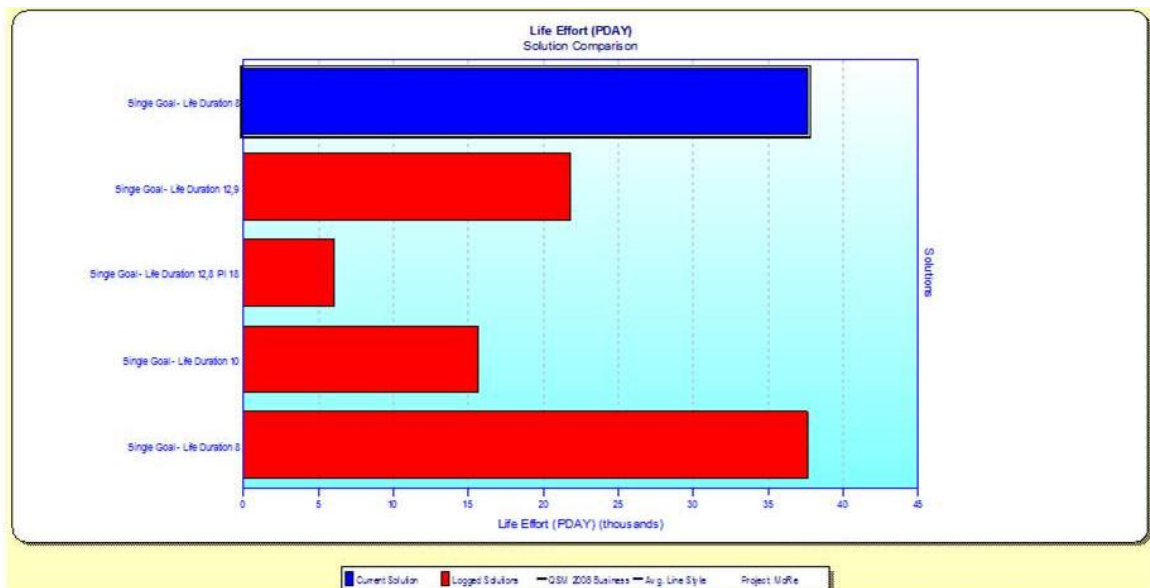


Abbildung 7: Screenshot aus SLIM-Estimator Reportvergleich (Projektaufwand)

Work Breakdown Structure Report

Der WBS-Report enthält eine detaillierte Unterteilung des Projekts in seine Phasen und die dazu gehörenden Arbeitspakete. Weiter werden zu den Datenpaketen der Start- und Endzeitpunkt sowie die Dauer, der Aufwand und die durchschnittlich benötigte Anzahl an Mitarbeitern angegeben. Ergänzend wird der Startzeitpunkt der einzelnen Milestones dargestellt.

Eine Teilweise Umsetzung der Erweiterung des Reportings ist denkbar.

4.3.2 Exportierung von Templates als XML-Datei unterstützen

Es soll möglich sein ein in einer XML-Datei gespeichertes Template in die Webanwendung zu laden. Umgekehrt soll es möglich sein, ein Template als XML-Datei abzuspeichern. Eine teilweise Umsetzung dieser Ausbaustufe ist umsetzbar.

4.3.3 Log Level Rotation

Log Level Rotation ermöglicht die Generierung einer gesamten Log Level Dateistruktur. Die Protokoll Dateien werden nicht in einer einzigen Datei gespeichert, sondern systematisch durch timestamps generiert und individuell verwaltet.

4.4 Abgrenzungskriterien

4.4.1 Sicherheitsbestimmungen für Web Services

Bei der Umsetzung des Projekts steht die Implementierung der Lösung im Vordergrund. Es erfolgt keine Verschlüsselung der Web Services. Die Daten werden unverschlüsselt zur Verfügung gestellt. Es wird nur ein Mindestmaß an Sicherheit gewährleistet.

4.4.2 Internationalisierung

Für die Interaktion mit dem Benutzer wird als Sprache Englisch verwendet. Es erfolgt keine Unterstützung von mehreren Sprachen.

4.4.3 Benutzerverwaltung

Auf administrative Use Cases wie das Befüllen von Wertetabellen (User, Gruppen, Industrien, Bereichen etc.) wird im Rahmen des Projektes verzichtet, diese werden direkt in die Datenbank geladen.

4.4.4 Template Verwaltung

Ein Template Manager erlaubt das Generieren von Templates sowie das nachträgliche Modifizieren von bereits erstellten Templates.

4.4.5 Import/Export von WebEst in SLIM

Eine weitere Erweiterungsmöglichkeit wäre die Import-/ Exportfunktion in das von IBM verwendete SLIM Tool.

4.4.6 Versionierung

Die Versionierung des zu entwickelnden Web Application erlaubt eine fehlerfreie Rückführung auf ältere Versionsstände.

4.4.7 Milestone Calculator

Der Milestone Calculator generiert aus projektbezogenen Eckdaten (Projektdauer, benötigte Ressourcen, zeitliche Rahmenbedingungen etc.) vorgegebene Milestones und positioniert diese im Projektablaufsplan.

4.4.8 Such-/Filtermöglichkeit

Durch die Implementierung einer Such-/Filtermöglichkeit soll es möglich sein, dynamisch nach bereits erstellten Estimates zu suchen. Im Rahmen des aktuellen Projektes wird diese Funktionalität nicht implementiert.

4.4.9 Defect Categories

Eine Einbeziehung der verschiedenen Fehlerkategorien (Low, Medium, High) sollen im Initialprojekt nicht implementiert werden.

5 Produkteinsatz

Das Ziel dieses Projektes ist die Ersetzung und Erweiterung eines Prozesses, der beim Kunden im Bereich der Aufwandsschätzung bereits produktiv genutzt wird.

5.1 Anwendungsbereiche

Die bereits vorhandene Software zur Aufwandsschätzung soll portiert werden, so dass diese als Web-Anwendung nutzbar wird. Die eigentliche Analyse-Software "SLIM Estimator" wird weiterhin verwendet werden.

Der Anwendungsbereich der bisherigen Software erweitert sich um die Möglichkeiten, die eine Web-Anwendung mit sich bringt:

- Plattformunabhängigkeit: Es ist nur ein Webbrowser erforderlich. Um welches Betriebssystem es sich handelt, ist in erster Linie nicht entscheidend.
- Eliminierung möglicher Fehlerquellen: Die durch verschiedene Excel-Versionen sowie bei der Übertragung von den Daten aus den verschiedenen Tools entstehen können.
- Geringere Wartungskosten: Die Logik der Schätzung liegt zentral auf dem Server, anstatt in mehreren Excel-Dateien.

5.2 Zielgruppen

Zielgruppe sind alle bereits produktiv arbeitenden Estimators der IBM Global Business Service, prinzipiell sind aber alle Mitarbeiter potenzielle Anwender. Durch die bereits genannten Vorteile wird es einem größeren Benutzerkreis möglich, einfache Schätzungen vorzunehmen. Die Benutzerführung soll dabei helfen, Bedienungsfehler zu minimieren.

5.3 Betriebsbedingungen

Die Anwendung sollte eine Verfügbarkeit von 99% im Intranet aufweisen. Eine Datensicherung sollte in regelmäßigen Abständen von einer qualifizierten Fachkraft absolviert werden.

6 Technische Produktumgebung

6.1 Software

6.1.1 Server

Das Produkt ist lauffähig auf einem IBM WebSphere Application Server Community Edition (IBM WASCE) in der Version 2.1. WebSphere nutzt einen Apache Geronimo Application Server welcher Tomcat als Servlet Container verwendet. Als Datenbank kommt eine integrierte Apache Derby Instanz zum Einsatz.

Grundlage für den Server ist folgende JVM:

- IBM Java SE 6 (32/64)

Die Serverkomponente des Produkts muss kompatibel zu folgenden Microsoft und Open Source Betriebssystemen sein:

- Windows XP (32 Bit)
- Windows 7 (32/64 Bit)
- Windows Server 2008 R2 (32/64 Bit)
- Debian GNU Linux v5 (32/64 Bit)
- Redhat Linux v5
- Novell SUSE Linux 11

6.1.2 Client

Zur Benutzung der grafischen Oberfläche wird der Browser Mozilla Firefox 3.6 verwendet.

6.2 Hardware

Die Hardwarevoraussetzungen entsprechen denen des IBM WASCE. Es wird folgendes vorausgesetzt:

- Mindestens 200MB freier Festplattenspeicher
- Mindestens 512MB Arbeitsspeicher Kapazität

7 Produktfunktionen

In diesem Kapitel werden die zu implementierenden Funktionen aus Benutzersicht gezeigt und erläutert.

7.1 Übersicht der Prozesse

Zur Übersicht der detaillierten Use-Cases ist hier Prozess in seinen groben Zügen als Ereignisgesteuerte Prozesskette (EPK) modelliert:

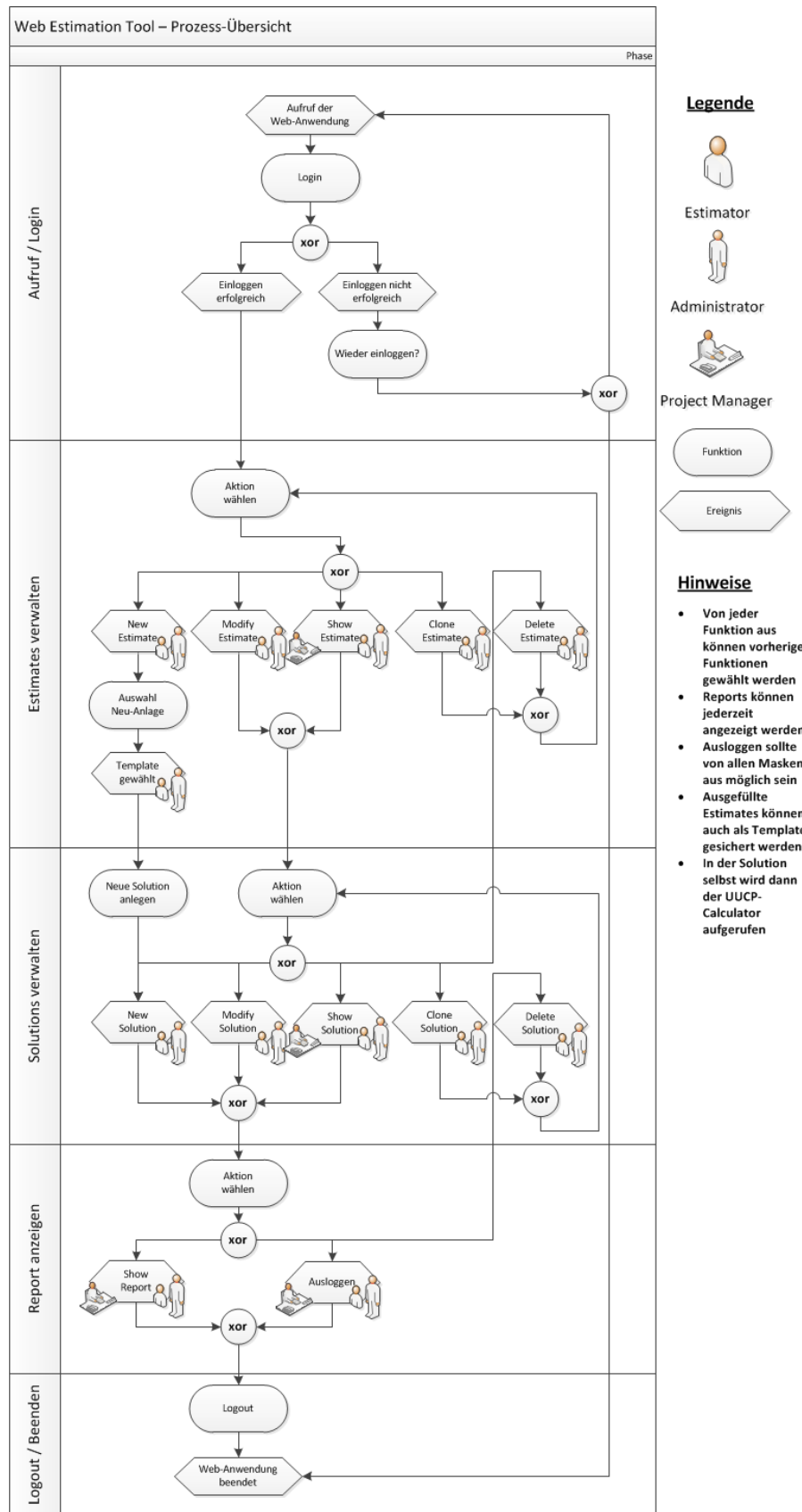


Abbildung 8: Prozessübersicht bzw. Use-Case Übersicht

7.2 Detaillierte Use-Cases

Anwendungsfall	Login
Kurzbeschreibung	Formular mit Eingabemöglichkeiten für Benutzername und Passwort sowie Button für Login
Rollen	Estimator, Projektmanager und Administrator
Minimale Garantien	Anzeige der Startseite
Erfolgsgarantien	Nach erfolgreichem Login wird dem User eine Liste der letzten Estimates und Solutions, die seiner Rolle zugeordnet sind, angezeigt
Hauptakteur	Benutzer
Untergeordnete Akteure	Keine
Übergeordnete Use Cases	Keine
Untergeordnete Use Cases	New Estimate, Modify Estimate, Clone Estimate, Delete Estimate, Show Estimate
Vorbedingungen	Seite muss geöffnet sein
Haupterfolgs-szenario	<ol style="list-style-type: none"> 1. Benutzer gibt Benutzername und Passwort ein 2. Drücken von "Login" 3. Dem Benutzer wird vom System seine Rolle zugeordnet 4. Startseite wird geöffnet
Verschiedenes	Der Benutzer kann sich jederzeit im angemeldeten Zustand wieder abmelden. Bei Falsch-Eingabe des Passworts oder Benutzernamens wird wieder die Login-Seite aufgerufen

Tabelle 2: Use-Case für Benutzer Login

Anwendungsfall	New Estimate
Kurzbeschreibung	Eine neue Estimation wird erstellt
Rollen	Estimator und Administrator
Minimale Garantien	Im Mainframe wird der Templateauswahlscreen angezeigt
Erfolgsgarantien	Eine neue Estimation wird angelegt und kann ausgefüllt werden
Hauptakteur	Estimator
Untergeordnete Akteure	Administrator
Übergeordnete Use Cases	<ul style="list-style-type: none"> • Login
Untergeordnete Use Cases	Neue Solution anlegen
Vorbedingungen	Erfolgreiches Login
Haupterfolgs-szenario	<ol style="list-style-type: none"> 1. Template-Auswahl wird angezeigt 2. Es wird eine leere Estimation oder je nach Template gefüllt Estimation angezeigt 3. In mehreren Tabs können die Werte eingegeben werden
Verschiedenes	Hier werden ein Kommentarlog und ein Kommentarfeld angezeigt, der Benutzer kann Kommentare hinzufügen, editieren oder löschen.

Tabelle 3: Use-Case für Erstellung eines neuen Estimates

Anwendungsfall	Modify Estimate
Kurzbeschreibung	Eine bereits angelegte Estimation wird editiert
Rollen	Estimator
Minimale Garantien	Der erste Tab des zu bearbeitenden Estimates wird angezeigt
Erfolgsgarantien	Ein angelegtes Estimate wird angezeigt und die verschiedenen Tabs können editiert werden
Hauptakteur	Estimator
Untergeordnete Akteure	Administrator
Übergeordnete Use Cases	<ul style="list-style-type: none"> • Login
Untergeordnete Use Cases	<ul style="list-style-type: none"> • Neue Solution anlegen • Delete Estimation
Vorbedingungen	<ul style="list-style-type: none"> • Erfolgreiches Login • Es muss mindestens eine Estimation existieren
Haupterfolgs-szenario	<ol style="list-style-type: none"> 1. Die gewählte Estimate wird angezeigt 2. In mehreren Tabs können Werte eingegeben werden
Verschiedenes	Rechts werden ein Kommentarlog und ein Kommentarfeld angezeigt, hier kann der Benutzer Kommentare hinzufügen, editieren oder löschen.

Tabelle 4: Use-Case zur Veränderung eines Estimates

Anwendungsfall	Clone Estimate
Kurzbeschreibung	Eine bereits bestehende Estimation wird 1:1 kopiert.
Rollen	Estimator und Administrator
Minimale Garantien	Es werden alle der Rolle zugeordneten Estimates zum Klonen angezeigt
Erfolgsgarantien	Eine Estimation wird 1:1 kopiert und unter neuem Namen gesichert
Hauptakteur	Estimator
Untergeordnete Akteure	Administrator
Übergeordnete Use Cases	<ul style="list-style-type: none"> • Login
Untergeordnete Use Cases	<ul style="list-style-type: none"> • Neue Solution anlegen • Modify Estimate
Vorbedingungen	<ul style="list-style-type: none"> • Erfolgreiches Login • Es muss mindestens eine Estimation existieren
Haupterfolgs-szenario	<ol style="list-style-type: none"> 1. Es wird eine Auswahl an bestehenden Estimates angezeigt 2. Nach Auswahl der Estimation wird das Objekt kopiert 3. Die geklonte Estimation öffnet sich im Editier-Modus und es kann ein neuer Name vergeben werden

Tabelle 5: Use-Case zum Vervielfältigen von einem bestehenden Estimate

Anwendungsfall	Delete Estimate
Kurzbeschreibung	Eine bereits bestehende Estimation wird gelöscht
Rollen	Estimator und Administrator
Minimale Garantien	Eine Liste der Estimates wird angezeigt
Erfolgsgarantien	Eine gewählte Estimate wird gelöscht
Hauptakteur	Estimator
Untergeordnete Akteure	Administrator
Übergeordnete Use Cases	<ul style="list-style-type: none"> • Modify Estimation
Untergeordnete Use Cases	Keine
Vorbedingungen	<ul style="list-style-type: none"> • Erfolgreiches Login • Es muss mindestens eine Estimation existieren
Haupterfolgs-szenario	<ol style="list-style-type: none"> 1. Eine Liste der Estimates wird angezeigt 2. Man kann eine der Estimates auswählen 3. Diese Estimate wird gelöscht

Tabelle 6: Use-Case zum Löschen eines Estimates

Anwendungsfall	Show Estimate
Kurzbeschreibung	Eine bereits bestehende Estimation wird angezeigt
Rollen	Estimator, Projektmanager und Administrator
Minimale Garantien	Eine Liste der Estimates wird angezeigt
Erfolgsgarantien	Das erste Tab einer gewählten Estimate wird angezeigt
Hauptakteur	Projektmanager
Untergeordnete Akteure	Administrator und Estimator
Übergeordnete Use Cases	<ul style="list-style-type: none"> • Login
Untergeordnete Use Cases	Keine
Vorbedingungen	<ul style="list-style-type: none"> • Erfolgreiches Login • Es muss mindestens eine Estimation existieren
Haupterfolgs-szenario	<ol style="list-style-type: none"> 1. Eine Liste der Estimates wird angezeigt 2. Man kann eine der Estimates auswählen 3. Das erste Tab des gewählten Estimates wird angezeigt 4. Es können keine Werte geändert werden

Tabelle 7: Use-Case zum Anzeigen eines Estimates

Anwendungsfall	New Solution
Kurzbeschreibung	Aus eine Estimation wird eine Solution angelegt.
Rollen	Estimator und Administrator
Minimale Garantien	Im Mainframe wird der erste Tab der Solution angezeigt
Erfolgsgarantien	Eine neue Solution wird erstellt
Hauptakteur	Estimator
Untergeordnete Akteure	Administrator
Übergeordnete Use Cases	<ul style="list-style-type: none"> • Neue Estimation anlegen • Estimation modifizieren
Untergeordnete Use Cases	<ul style="list-style-type: none"> • Report erstellen • Delete Solution
Vorbedingungen	<ul style="list-style-type: none"> • Erfolgreiches Login • Anlage/Änderung einer Estimation
Haupterfolgs-szenario	<ol style="list-style-type: none"> 1. Aus der Estimation wird eine neue Solution erstellt 2. Der erste Tab der Solution wird angezeigt 3. Die Werte in den Tabs der Solution können geändert werden
Verschiedenes	Rechts wird ein Kommentarlog und ein Kommentarfeld angezeigt, hier kann der Benutzer Kommentare hinzufügen, editieren oder löschen.

Tabelle 8: Use-Case zum Erstellen einer Solution

Anwendungsfall	Modify Solution
Kurzbeschreibung	Eine bereits angelegte Solution wird editiert
Rollen	Estimator
Minimale Garantien	Der erste Tab des zu bearbeitenden Solution wird angezeigt
Erfolgsgarantien	Ein angelegtes Solution wird angezeigt und die verschiedenen Tabs können editiert werden
Hauptakteur	Estimator
Untergeordnete Akteure	Administrator
Übergeordnete Use Cases	<ul style="list-style-type: none"> • Neue Estimation anlegen • Estimation modifizieren
Untergeordnete Use Cases	<ul style="list-style-type: none"> • Report erstellen • Delete Solution
Vorbedingungen	<ul style="list-style-type: none"> • Erfolgreiches Login • Es muss mindestens eine Estimation existieren • Es muss mindestens eine Solution existieren
Haupterfolgs-szenario	<ol style="list-style-type: none"> 1. Die gewählte Solution wird angezeigt 2. Der erste Tab der Solution wird angezeigt 3. Die Werte in den Tabs der Solution können geändert werden
Verschiedenes	Rechts werden ein Kommentarlog und ein Kommentarfeld angezeigt, hier kann der Benutzer Kommentare hinzufügen, editieren oder löschen.

Tabelle 9: Use-Case zum Verändern einer Solution

Anwendungsfall	Clone Solution
Kurzbeschreibung	Eine bereits bestehende Solution wird 1:1 kopiert.
Rollen	Estimator und Administrator
Minimale Garantien	Es werden alle der Rolle zugeordneten Solutions zum Klonen angezeigt
Erfolgsgarantien	Eine Solution wird 1:1 kopiert und unter neuem Namen gesichert
Hauptakteur	Estimator
Untergeordnete Akteure	Administrator
Übergeordnete Use Cases	<ul style="list-style-type: none"> • Modify Estimation
Untergeordnete Use Cases	<ul style="list-style-type: none"> • Report erstellen • Modify Solution
Vorbedingungen	<ul style="list-style-type: none"> • Erfolgreiches Login • Es muss mindestens eine Estimation existieren • Es muss mindestens eine Solution existieren
Haupterfolgs-szenario	<ol style="list-style-type: none"> 1. Es wird eine Auswahl an bestehenden Solutions angezeigt 2. Nach Auswahl der Solution wird das Objekt kopiert 3. Die geklonte Solution öffnet sich im Editier-Modus und es kann ein neuer Name vergeben werden

Tabelle 10: Use-Case zum Vervielfältigen einer Solution

Anwendungsfall	Delete Solution
Kurzbeschreibung	Eine bereits bestehende Solution wird gelöscht
Rollen	Estimator und Administrator
Minimale Garantien	Eine Liste der Solutions wird angezeigt
Erfolgsgarantien	Eine gewählte Solution wird gelöscht
Hauptakteur	Estimator
Untergeordnete Akteure	Administrator
Übergeordnete Use Cases	<ul style="list-style-type: none"> • Modify Solution
Untergeordnete Use Cases	Keine
Vorbedingungen	<ul style="list-style-type: none"> • Erfolgreiches Login • Es muss mindestens eine Estimation existieren • Es muss mindestens eine Solution existieren
Haupterfolgs-szenario	<ol style="list-style-type: none"> 1. Eine Liste der Solutions wird angezeigt 2. Man kann eine der Solution auswählen 3. Diese Solution wird gelöscht

Tabelle 11: Use-Case zum Löschen einer Solution

Anwendungsfall	Show Solution
Kurzbeschreibung	Eine bereits bestehende Solution wird angezeigt
Rollen	Estimator, Projektmanager und Administrator
Minimale Garantien	Eine Liste der Solutions wird angezeigt
Erfolgsgarantien	Das erste Tab einer gewählten Solution wird angezeigt
Hauptakteur	Projektmanager
Untergeordnete Akteure	Administrator und Estimator
Übergeordnete Use Cases	Show Estimation
Untergeordnete Use Cases	Keine
Vorbedingungen	<ul style="list-style-type: none"> • Erfolgreiches Login • Es muss mindestens eine Estimation existieren • Es muss mindestens eine Solution existieren
Haupterfolgs-szenario	<ol style="list-style-type: none"> 1. Eine Liste der Solutions wird angezeigt 2. Man kann eine der Solution auswählen 3. Das erste Tab der gewählten Solution wird angezeigt 4. Es können keine Werte geändert werden

Tabelle 12: Use-Case zum Anzeigen einer Solution

Anwendungsfall	Show Report
Kurzbeschreibung	Aus einer Solution wird ein Report erstellt
Rollen	Estimator, Projektmanager und Administrator
Minimale Garantien	Der Report Screen erscheint
Erfolgsgarantien	Der Report Screen erscheint, es können verschiedene Ansichten der Charts gewählt werden
Hauptakteur	Projektmanager
Untergeordnete Akteure	Administrator und Estimator
Übergeordnete Use Cases	Anlage oder Modifikation einer Solution
Untergeordnete Use Cases	Keine
Vorbedingungen	<ul style="list-style-type: none"> • Erfolgreiches Login • Es muss mindestens eine Estimation existieren • Es muss mindestens eine Solution existieren
Haupterfolgs-szenario	<ol style="list-style-type: none"> 1. Aus der Solution heraus kann der Report aufgerufen werden 2. Es wird sofort ein neuer Report zu der Solution erstellt und angezeigt 3. Im Report können verschiedene Ansichten der Charts ausgewählt werden

Tabelle 13: Use-Case zum Anzeigen von einem Report

8 Produktdaten

8.1 Beschreibung der langfristig gespeicherten Daten aus Nutzersicht

- **Benutzer:** Jeder Anwender der Software ist ein Benutzer dessen Stammdaten gespeichert werden. Jedem Benutzer wird eine Rolle (Administrator, Estimator oder Project Manager) zugeteilt, welche mit bestimmten Berechtigungen verbunden ist.
- **Estimate:** Zu jedem Estimate werden die zugehörigen Projektdaten gespeichert.
- **Phase:** Ein Projekt ist in verschiedene Phasen gegliedert.
- **Meilenstein:** Zu einem Projekt werden verschiedene zu erreichende Meilensteine definiert.
- **Solution:** Zu jedem Estimate können verschiedene Solutions angelegt werden, die eine bestimmte Schätzung darstellen.
- **Use-Case:** Definiert die Use-Cases als Basis der Schätzung.

Benutzer	Benutzername*	50 Zeichen a-z 0-9 @ _ - .
	Passwort*	50 Zeichen
	Vorname*	50 Zeichen
	Nachname*	50 Zeichen
	Rolle	1
	Organisation/Division	1

Tabelle 14: Datenbanktabelle für Benutzer

* Pflichtfeld

Estimate	Name*	50 Zeichen
	Beschreibung	4000 Zeichen
	Estimator*	1 (ist Benutzer)
	Organisation/Division*	1
	Projektname*	50 Zeichen
	Projektbeschreibung	4000 Zeichen
	Tage pro Woche*	Zahl 0-7
	Stunden pro Tag*	Zahl 0 - 24
	Land*	1
	Währung*	1
	Aufwandseinheit (Effort Unit)*	1
	Mean Time To Defect-Zeiteinheit (MTTD Time Unit)*	1
	Industrie-Sektor*	1
	Art der Anwendung (Application Type)*	1
	Phasen*	4
	Meilensteine	>= 0
	Template	0-1
	Solutions*	>= 1
	Defect Categories	5

Tabelle 15: Datenbanktabelle für Estimates

Phase	Nummer*	Zahl 1-4
	Abkürzung*	10 Zeichen
	Name*	50 Zeichen
	Beschreibung	4000 Zeichen

Tabelle 16: Datenbanktabelle für Phasen

 * Pflichtfeld

Meilenstein	ID*	Zahl <= 65000
	Abkürzung*	10 Zeichen
	Name*	50 Zeichen
	Prozent*	Zahl 0-100
	Phase*	1
	Beschreibung	4000 Zeichen

Tabelle 17: Datenbanktabelle für Meilensteine

Solution	Name*	50 Zeichen
	Startdatum*	Datum
	Expected PI*	Zahl 0-100
	Kommentar	4000 Zeichen
	Constraints	>= 0
	Use Cases (gruppiert)	>= 0
	Granularitätsfragen*	9
	Staffing Shapes pro Phase	1 pro Phase

Tabelle 18: Datenbanktabelle für Solutions

Use Case	Name*	255 Zeichen
	Multiplikator*	Kommazahl
	Granularität	0-1
	In Scope*	Wahrheitswert
	Sicherheit (certainty)*	1
	Annahmen	4000 Zeichen
	Komplexität*	1

Tabelle 19: Datenbanktabelle für Use Cases

 * Pflichtfeld

Template	Name*	50 Zeichen
	Beschreibung	4000 Zeichen
	Tage pro Woche*	Zahl 0-7
	Stunden pro Tag*	Zahl 0-24
	Phasen*	4
	Meilensteine	≥ 0
	Defect Categories	5

Tabelle 20: Datenbanktabelle für Templates

* Pflichtfeld

9 Leistungen

In diesem Teil sind verschiedene Regelungen und Rahmenbedingungen zu Umsetzung zu dem Projekt enthalten.

9.1 Zeitbezogen

Durch die Vorgaben des IT-Praktikums ist der zeitliche Rahmen der Projektdurchführung genau abgesteckt. Anfang Dezember 2010 wird die Planung abgeschlossen und mit der Implementierung der Lösung begonnen. Der endgültige Termin zur Fertigstellung des Produktes ist der 1. März 2011.

Der Testzeitraum und mit der Übergabe der Software sollte ungefähr einen Monat betragen. Dies wäre ab Anfang Februar der Fall.

Innerhalb dieses Zeitraumes stehen auch einem Studenten nur begrenzt Ressourcen zur Verfügung. Mit der Absolvierung des IT-Praktikums erhalten die Teilnehmer 10 Credit Points, welche umgerechnet ca. 200-300 Stunden Aufwand widerspiegeln. Für das gesamte Projekt ergeben sich für 9 Teammitglieder damit zeitliche Ressourcen von in etwa 1800-2700 Stunden. Der Aufwand jedes Studenten wird erfasst und je nach Aufgabe ausgewertet.

9.2 Umfangsbezogen

Wie bereits zuvor erwähnt, sind durch das IT-Praktikum schon einzelne Vorgaben und Rahmenbedingungen vorhanden, die folgende Ergebnisse umfassen:

- Dokumente
 - Projektplan
 - Pflichtenheft (Fachkonzept)
 - DV-Konzept
 - Dokumentation
 - Abschlussbericht
 - Testprotokolle
- Funktionsfähige Software
- Zwischenpräsentation
- Abschlusspräsentation

10 Qualitätsanforderungen

Die Software wird dazu entwickelt um bereits bestehende Prozesse zu beschleunigen bzw. zu erleichtern.

Besonders bei diesem Projekt ist es wichtig, ein Produkt einzuführen, welches schnell und unkompliziert genutzt werden kann. Ansonsten würde die Nutzung des Produkts unnötig Zeit verschwenden und somit für das Unternehmen Kosten verursachen, welche - betriebswirtschaftlich gesehen - vermieden werden sollten. So bald das Produkt mehr Kosten als Nutzen verursacht, hat sich die Wettbewerbssituation des Kunden verschlechtert.

Aufgrund dessen wurde die Gewichtung bei den Qualitätsanforderungen wie folgt verteilt.

	Sehr wichtig	Wichtig	Weniger wichtig	Unwichtig
Robustheit	X			
Zuverlässigkeit	X			
Korrektheit	X			
Benutzerfreundlichkeit		X		
Effizienz		X		
Portierbarkeit			X	
Kompatibilität		X		
Wartbarkeit		X		

Tabelle 21: Verteilung von Qualitätsanforderungen an die einzelnen Softwareattribute

Robustheit

Hat eine sehr hohe Priorität, da die Anwendung Ergebnisse zurückgibt, auf die sich der Benutzer verlassen muss. Der Benutzer sollte jederzeit wissen, wo er sich im System befindet, und beurteilen können, was seine Aktionen verändern. Korrekturen sollten leicht möglich sein und auf Fehler sollte vom System hingewiesen werden.

Zuverlässigkeit

Da das Tool für den täglichen Einsatz gedacht ist, sollte es zuverlässig funktionieren.

Korrektheit

Der Benutzer muss sich auf die Ergebnisse verlassen können, deshalb hohe Priorität.

Benutzerfreundlichkeit

Auch Benutzer, die keine professionellen Anwender von Estimation Tools (z.B. SLIM) sind, sollten sich schnell zurechtfinden und in einem angemessenen Zeitraum einarbeiten können.

Effizienz

Das Tool hat die Aufgabe, schneller als komplizierte, umfassende Estimation Tools zu arbeiten. In Bezug auf die Berechnungen ist die Effizienz jedoch zu vernachlässigen, da keine komplizierten, zeitaufwändigen Berechnungen durchgeführt werden.

Portierbarkeit

Portierbarkeit ist weniger wichtig, da es sich um eine rein IBM-interne Anwendung handelt. Dies widerspricht nicht der Bereitstellung von Schnittstellen, die eine vielfältige, flexible Nutzung möglich machen. Implementierung in Java erleichtert die Portierbarkeit zusätzlich.

Kompatibilität

Es werden gängige Programmiersprachen / Datenbanken / Frameworks / Datentypen etc. verwendet.

Wartbarkeit

Eine nachhaltige Wartbarkeit des Produkts wird durch Standards (siehe auch Kompatibilität), standardisierte Code- und Kommentierungsrichtlinien und Code-Reviews erreicht.

11 Benutzungsoberfläche

Im Folgenden werden die Anforderungen an die Benutzeroberfläche (GUI) beschrieben. Die Implementierung erfolgt mit Hilfe von JSF, AJAX und CSS. Die Benutzeroberfläche sollte den folgenden Anforderungen genügen.

11.1 Allgemeine Anforderungen

- Das Layout muss per Cascading Style Sheets (CSS) konfigurierbar sein.
- Die Benutzeroberfläche folgt Standards für Internet-Seiten, sprich den W3C-Standards, und sollte diesen genügen.
- Die Implementierung folgt dem Model-View-Controller-Pattern.
- Die Sprache der Benutzeroberfläche ist Englisch.
- Es wird AJAX verwendet, um eine interaktive Benutzeroberfläche zu bieten.
- Die GUI folgt den Prinzipien der Usability, also den Prinzipien der Erlernbarkeit, Flexibilität und Robustheit.
- Sie orientiert sich an schon dem Benutzer bekannten Darstellungsweisen (Benutzung von Tabs, Navigationsmenüs).
- Die GUI sollte so weit wie möglich selbsterklärend bzw. schnell erlernbar sein.
- Die Oberfläche ist auf Benutzung mit Maus und Tastatur ausgelegt.
- Die Darstellung der GUI muss für Mozilla Firefox optimiert sein.
- Als Zeichensatz wird standardmäßig UTF-8 verwendet.

11.2 Aufteilung der Arbeitsfläche

Die Aufteilung der Arbeitsoberfläche soll mit folgenden, beispielhaften Screenshots, illustriert werden.

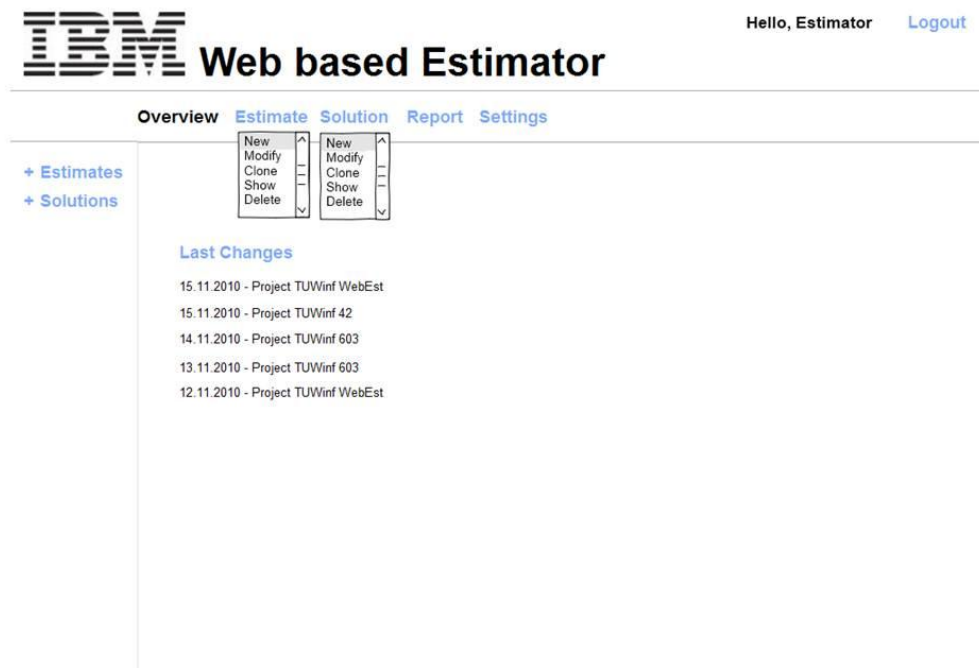


Abbildung 9: Mögliche Darstellung des „Welcome“-Screens

Die GUI ist in drei große Teilbereiche eingeteilt. Die Hauptnavigation befindet sich im Header. Die Navigation der linken Seitenleiste passt sich dynamisch dem jeweiligen Inhalt des Mainframes an. Im Mainframe werden die Informationen angezeigt und ein Großteil der Interaktion findet statt. Die Menüpunkte der Hauptnavigation bieten eine Dropdown-Funktionalität an. Wenn der Benutzer z.B. auf den Menüpunkt "Estimate" klickt, erscheint ein Dropdown Menü mit den Möglichkeiten, ein neues Estimate anzulegen, ein Estimate zu modifizieren, anzuzeigen oder zu löschen.

Die linke Navigationsleiste zeigt ausklappbare Menüs an, die dazu dienen, bestimmte Estimates und Solutions schneller zu erreichen.

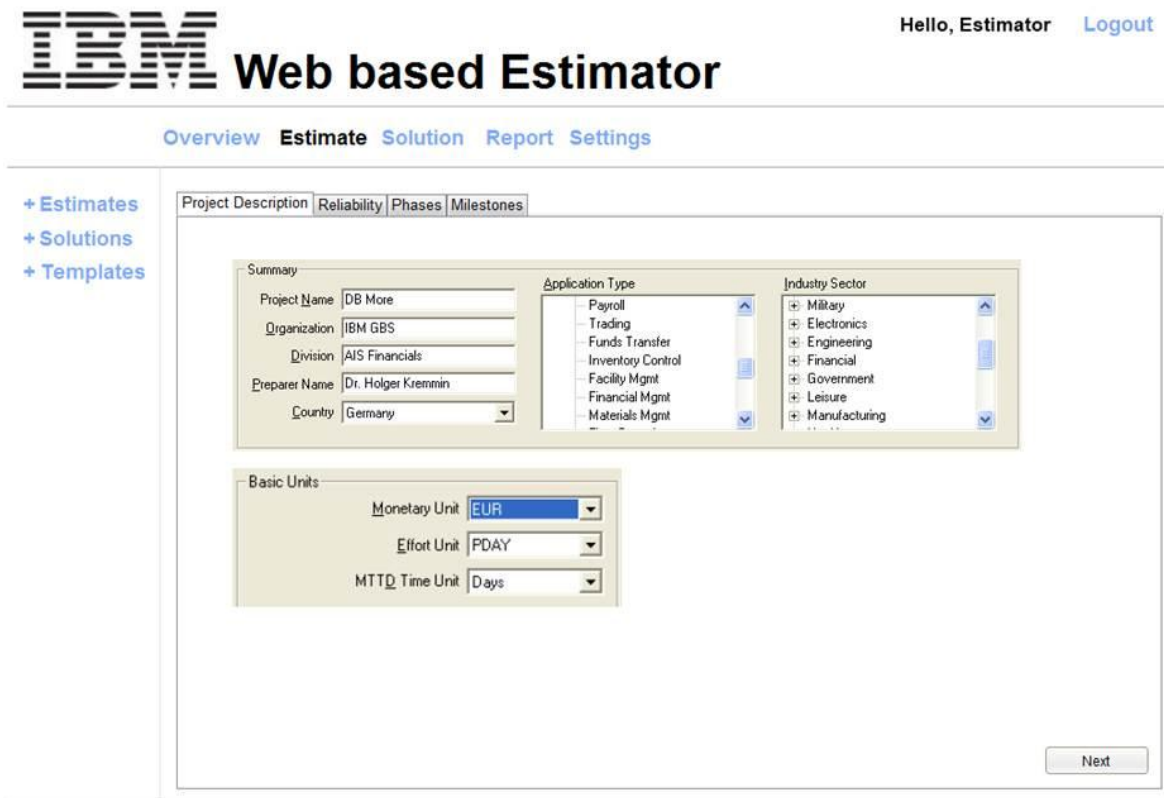


Abbildung 10: Mögliche Darstellung des Estimationprozesses

Der Hauptteil der Interaktion findet im Mainframe statt. Zusammengehörende Informationen, wie z.B. die Komponenten, die zu "Estimate" gehören, werden in Tabs angezeigt.

Innerhalb der Tabs gibt es Formularfelder zur Eingabe von Informationen, Buttons, um sich weiter oder zurück zu bewegen oder auch eine Auswahl zu bestätigen oder zu verwerfen sowie andere bekannte GUI Elemente wie z.B. Radio Buttons, Dropdown Menüs etc. Desweiteren wird in "Estimates" und "Solutions" jeweils rechts ein Kommentarlog angezeigt, in dem neue Kommentare angelegt, geändert oder gelöscht werden können. Darüber hinaus bietet die GUI eine dynamische "on-the-fly" Anzeige von Charts (z.B. Stab- und Liniendiagramme) bei der Erstellung von Reports.

12 Testen

Im Folgenden wird auf die Ausgestaltung der Testfälle eingegangen. Das Testen beinhaltet funktionale, als auch nicht-funktionale Tests.

Die funktionalen Testfälle dienen zur Überprüfung des funktionalen Ablaufes, das heißt sie beschreiben Testszenarien aus der Sicht des Anwenders bzw. Benutzers. Ein funktionaler Test überprüft beispielsweise die definierte Anforderung, respektive die gewünschte Funktionalität, eines GUI-Elementes (z.B. Button).

Folgende funktionale Testfälle sind umzusetzen:

- Login
- New Estimate
- Modify Estimate
- Clone Estimate
- Delete Estimate
- Show Estimate
- New Solution
- Modify Solution
- Clone Solution
- Show Solution
- Delete Solution
- Show Report

Nicht funktionale Tests hingegen sind Tests, die eine bestimmte Eigenschaft des Systems überprüfen sollen, z.B. Robustheit oder Benutzerfreundlichkeit (die genauen Ausprägungen sind unter Kapitel 10 beschrieben) einer Anwendung.

Des Weiteren hat eine Dokumentation der Testfälle zu erfolgen. Die Dokumentation dient als Qualitätssicherung um die Erfüllung der Anforderungen zu gewährleisten.

13 Entwicklungsumgebung

13.1 Software

Durch die Anforderungen des Projektes ist die zu benutzende Software zu einem großen Teil bereits vorgegeben:

- Programmierung: "Eclipse JavaEE" mit den Erweiterungen
 - Subclipse
 - WTP Server Adapters (für WASCE v2.1)
 - JavaDocs
 - JUnit
- Browser: Firefox 3.6

- Server:
 - WebSphere Application Server Community Edition v2.1
 - Subversion
- Bugtracker: Mantis
- Testen:
 - Java: JUnit
 - Webservices: soapUI
 - Oberflächentests: Selenium
 - Web-Load-Tests: soapUI
- Weitere Komponenten:
 - IBM Java SDK v1.6

Als Entwicklungsumgebung wird Microsoft Windows eingesetzt.

13.2 Hardware

Jeder Entwickler wird die Implementierung des Projektes auf seinem ihm zur Verfügung stehenden Computer vornehmen, die Hardware fällt dabei sehr unterschiedlich aus. Um z.B. Probleme mit Zeichensätzen, unterschiedlichen Versionen etc. zu vermeiden, erstellt eine Arbeitsgruppe Anleitungen zur standardisierten Installation der Komponenten.

Fertige Versionsstände werden auf einem vom Fachbereich für Rechts- und Wirtschaftswissenschaften bereitgestellten Server mit Subversion geladen. Eine Produktiv-Umgebung existiert nicht. Auf dem Server wird lediglich der Code mit dem dazugehörigen Bugtracking verwaltet.

13.3 Orgware

Um die Projektleitung und die Implementierungsarbeit zu vereinfachen, werden folgende Dokumente gemeinsam erweitert:

- Projektplan (Microsoft Project 2010)
- ERM- und UML-Diagramm (Microsoft Visio 2010)
- Dokumentation (Microsoft Word)

13.4 Entwicklungsschnittstellen

Die komplette Anwendung wird in einem Enterprise Application Archive (EAR) auf dem Application Server installiert. Alle funktional und technisch bedingten Teilprojekte werden in diesem Archiv zusammengeführt. Somit stehen alle zum Betrieb der Anwendung erforderlichen Komponenten, die nicht Teil des Application Servers sind, mit der Installation des Archivs zur Verfügung.

14 Ergänzungen

14.1 Projektablauf

In Abbildung 11 ist der geplante Projektablauf erläutert. Ein genauerer Projektplan liegt der Projektleitung vor. Eventuelle Pufferzeiten finden in dieser stark vereinfachten Darstellung keine Berücksichtigung. Der Termin der Abschlusspräsentation wurde noch nicht von Seiten des Fachgebiets Information Systems der TU Darmstadt bestätigt.

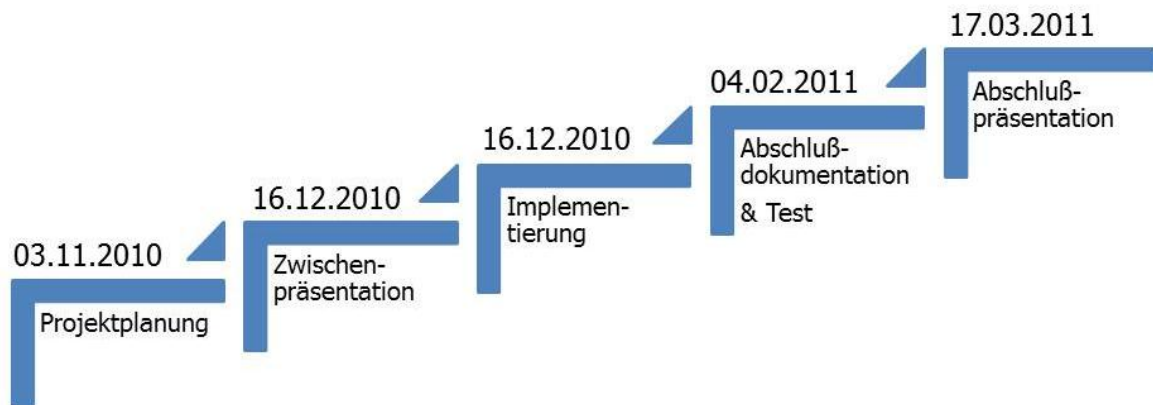


Abbildung 11: Grobkonzept des Projektablaufs

14.2 Lizenzierung

Das Produkt ist der Apache-Lizenz 2.0 unterstellt und wird somit als OpenSource Software entwickelt. Alle verwendeten Komponenten müssen mindestens den Anforderungen dieser Lizenz entsprechen. Eine Kopie der Lizenz findet sich unter folgender URL.
<http://www.apache.org/licenses/LICENSE-2.0>.

15 Anhang

15.1 A1

In unten stehender Tabelle werden die Zuordnungskriterien für Granularitätslevel aus der vom Kunden zur Verfügung gestellten Excel-Tabelle dargestellt.

Complexity	Decomposed	Median	Essential
High	<p>Business function is not self-contained - Implementation of the function will depend on frameworks which may be more complex than the business functions the framework supports</p> <p>There are ten or fewer business rules</p> <ul style="list-style-type: none"> • Less than half with interdependencies • No more than a third have exceptions defined • Less than three rules require external validity checks <p>Algorithms are complex but well documented and consistent</p> <p>Two or fewer keys used to search data</p> <p>No interaction with historical data</p> <p>One simple security check during a transaction</p> <p>Not more than one real time (synchronous) external accesses</p> <p>Six or fewer major business objects accessed during a transaction - No more than three major business objects updated and no more than three major business objects accessed</p>	<p>Business function is a fundamental business construct</p> <p>Thirty or fewer business rules have been identified</p> <ul style="list-style-type: none"> • More than half of the rules have exceptions, some complex • Less than ten rules exhibit interdependencies • Less than five rules require external validity checks (e.g., data table lookup) <p>Complex algorithms and calculations exist in support of this business function</p> <p>Five or fewer keys used to search data</p> <p>Need to access and maintain historical data</p> <p>Five or fewer security checks during a transaction</p> <p>Not more than two real time (synchronous) external accesses</p> <p>Ten or fewer major business objects accessed during a transaction - No more than five major business objects updated and no more than five major business objects</p>	<p>Business function is a fundamental business construct</p> <p>Not more than fifty business rules have been identified</p> <ul style="list-style-type: none"> • More than 70% of the rules have exceptions, many complex • No more than fifteen rules exhibit interdependencies • Not more than ten rules require external validity checks (e.g., data table lookup) <p>Complex algorithms and calculations exist in support of this business function</p> <p>No more than five keys used to search data</p> <p>Need to access and maintain historical data</p> <p>No more than ten security checks during a transaction</p> <p>No more than five real time (synchronous) external accesses</p> <p>No more than twenty major business objects accessed during a transaction - no more than ten major business objects updated and</p>

		accessed	no more than five major business objects accessed
Medium	<p>Maximum of five business rules</p> <ul style="list-style-type: none"> • Less than two rules have interdependencies • No more than one simple exception condition <p>Algorithms are simple, well documented and easily understood by the developer without user involvement. There is only one algorithm per rule on average. No more than one key is used to search data</p> <p>No interaction with historical data</p> <p>No security checks during a transaction</p> <p>No real time (synchronous) external access, unless functional decompose and then not more than one (synchronous) external access</p> <p>Two or less major business objects accessed during a transaction - No more than one major business objects updated and no more than one major business objects accessed</p>	<p>Business function is not self-contained - Implementation of the function will depend on frameworks which may be more complex than the business functions the framework supports</p> <p>There are ten or fewer business rules</p> <ul style="list-style-type: none"> • Less than half with interdependencies • No more than a third have exceptions defined • Less than three rules require external validity checks <p>Algorithms are complex but well documented and consistent</p> <p>Two or fewer keys used to search data</p> <p>No interaction with historical data</p> <p>One simple security check during a transaction</p> <p>Not more than one real time (synchronous) external accesses</p> <p>Six or fewer major business objects accessed during a transaction - No more than three major business objects updated and no more than three major business objects accessed</p>	<p>Business function is a fundamental business construct</p> <p>Thirty or fewer business rules have been identified</p> <ul style="list-style-type: none"> • More than half of the rules have exceptions, some complex • Less than ten rules exhibit interdependencies • Less than five rules require external validity checks (e.g., data table lookup) <p>Complex algorithms and calculations exist in support of this business function</p> <p>Five or fewer keys used to search data</p> <p>Need to access and maintain historical data</p> <p>Five or fewer security checks during a transaction</p> <p>Not more than two real time (synchronous) external accesses</p> <p>Ten or fewer major business objects accessed during a transaction - No more than five major business objects updated and no more than five major business objects accessed</p>

Low	<p>Possible read-only requirement</p> <p>Maximum of one business rule with</p> <ul style="list-style-type: none"> • No interdependencies and no exception conditions • One simple edit check only <p>No security checks; one business object involved in transaction and only one transaction maximum</p>	<p>Maximum of five business rules</p> <ul style="list-style-type: none"> • Less than two rules have interdependencies • No more than one simple exception condition <p>Algorithms are simple, well documented and easily understood by the developer without user involvement. There is only one algorithm per rule on average. No more than one key is used to search data</p> <p>No interaction with historical data</p> <p>No security checks during a transaction</p> <p>No real time (synchronous) external access, unless functional decompose and then not more than one (synchronous) external access</p> <p>Two or less major business objects accessed during a transaction - No more than one major business objects updated and no more than one major business objects accessed</p>	<p>Business function is not self-contained - Implementation of the function will depend on frameworks which may be more complex than the business functions the framework supports</p> <p>There are ten or fewer business rules</p> <ul style="list-style-type: none"> • Less than half with interdependencies • No more than a third have exceptions defined • Less than three rules require external validity checks <p>Algorithms are complex but well documented and consistent</p> <p>Two or fewer keys used to search data</p> <p>No interaction with historical data</p> <p>One simple security check during a transaction</p> <p>Not more than one real time (synchronous) external accesses</p> <p>Six or fewer major business objects accessed during a transaction - No more than three major business objects updated and no more than three major business objects accessed</p>
Very low	<p>Read-only</p> <p>No business rules</p> <p>No security checks; no transactions</p>	<p>Possible read-only requirement</p> <p>Maximum of one business rule with</p>	<p>Maximum of five business rules</p> <ul style="list-style-type: none"> • Less than two rules have

	One business object involved	<ul style="list-style-type: none"> • No interdependencies and no exception conditions • One simple edit check only No security checks; one business object involved in transaction and only one transaction maximum	interdependencies <ul style="list-style-type: none"> • No more than one simple exception condition Algorithms are simple, well documented and easily understood by the developer without user involvement. There is only one algorithm per rule on average. No more than one key is used to search data No interaction with historical data No security checks during a transaction No real time (synchronous) external access, unless functional decompose and then not more than one (synchronous) external access Two or less major business objects accessed during a transaction - No more than one major business objects updated and no more than one major business objects accessed
--	------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------