



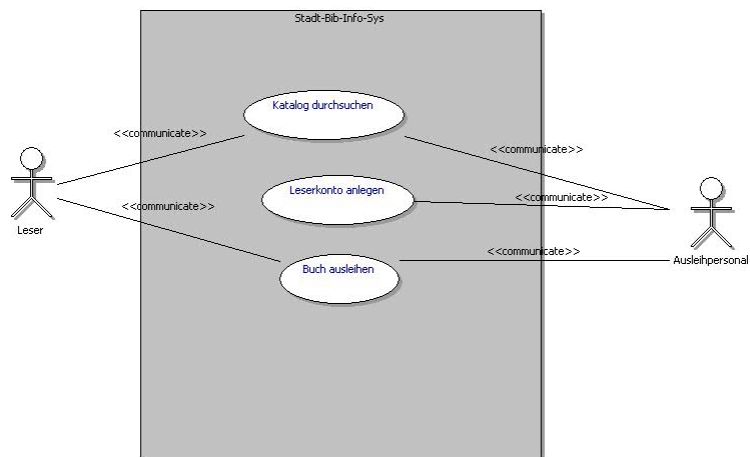
Anforderungsgetriebene SW-Entwicklung in Eclipse

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de

Version: 1.3



2



3

Gliederung

- Use Case Modelle
- **Analyseergebnisse**
- Einführung, Grundlagen
- Klassifizierungen
- Traceability
- CaliberRM Demo
- TogetherEC Demo

4

Analyseergebnisse

- Pflichtenheft (grobe fachliche Sammlung als Entscheider-Überblick)
- Grobspezifikation
 - UseCase-Sammlung (excel für Projektleiter, prios, Aufwände)
 - Aktorensammlung
 - Use Case Diagramm
- Feinspezifikation
 - Funktionale Anforderungen
 - **AWF-Modell**
 - AWF Template je AWF
 - Ablaufdiagramm je AWF
 - **Klassendiagramm mit Attributen**
 - **(Storyboard) - Beschreibung der Benutzeroberfläche**
 - **Optional Fachkonzepte mit hoher Komplexität als Dokument**
 - Honorarverteilungskonzept
 - Nicht-Funktionale Anforderungen
 - **Ergonomie (Layout-Spezifikation)**
 - **Nicht-funktionale Anforderungen (Qualität, ...)**

5

Ergebnistypen

Nr.	Name	Beschreibung	Prio

Liste Use Cases =
Grobspezifikation

Aktorensammlung

Titel	Inhalt
Beschreibung	Für die Analyse, werden an die Entwickler DV und UI, zusammen mit den Entwicklern, die Rollen, die an der Entwicklung teilnehmen, identifiziert. Die Rollen werden die DV-Komponenten und die Auftraggeber durch Rollen, verbunden.

Titel	Inhalt
Beschreibung	Die Rollen mit DV und UI, die die Aufgaben wahrnehmen werden, um sie zu bewerten, insbesondere den Zustand der Realisierung der Aufgabe zu erklären. Die Verantwortlichen für diese Aufgaben werden durch die Bewertung einer Aufgabe an die Rollen von Angewandten

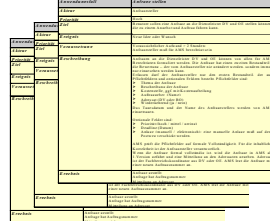


Use Cases Diagram

= Feinspezifikation

Detaillierte Beschreibung

der Use Cases



Aktivitätsdiagramme mit Stereotypen <<userInteraction>> <<showPage>> und <<systemAction>>

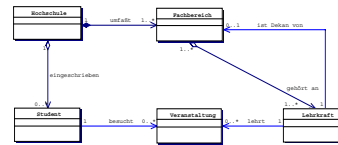
6

Ergebnistypen

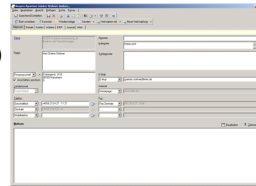
Nr.	Name	Beschreibung	Prio



+ Klassendiagramm



+ Storyboard (Screenshots)



UML-Klassendiagramm	UML-Klassendiagramm																								
<table border="1"> <tr><th>Name</th><th>Erklärung</th></tr> <tr><td>Hochschule</td><td>... (text)</td></tr> <tr><td>Fachbereich</td><td>... (text)</td></tr> <tr><td>Student</td><td>... (text)</td></tr> <tr><td>Veranstaltung</td><td>... (text)</td></tr> <tr><td>Lehrkraft</td><td>... (text)</td></tr> </table>	Name	Erklärung	Hochschule	... (text)	Fachbereich	... (text)	Student	... (text)	Veranstaltung	... (text)	Lehrkraft	... (text)	<table border="1"> <tr><th>Name</th><th>Erklärung</th></tr> <tr><td>Hochschule</td><td>... (text)</td></tr> <tr><td>Fachbereich</td><td>... (text)</td></tr> <tr><td>Student</td><td>... (text)</td></tr> <tr><td>Veranstaltung</td><td>... (text)</td></tr> <tr><td>Lehrkraft</td><td>... (text)</td></tr> </table>	Name	Erklärung	Hochschule	... (text)	Fachbereich	... (text)	Student	... (text)	Veranstaltung	... (text)	Lehrkraft	... (text)
Name	Erklärung																								
Hochschule	... (text)																								
Fachbereich	... (text)																								
Student	... (text)																								
Veranstaltung	... (text)																								
Lehrkraft	... (text)																								
Name	Erklärung																								
Hochschule	... (text)																								
Fachbereich	... (text)																								
Student	... (text)																								
Veranstaltung	... (text)																								
Lehrkraft	... (text)																								

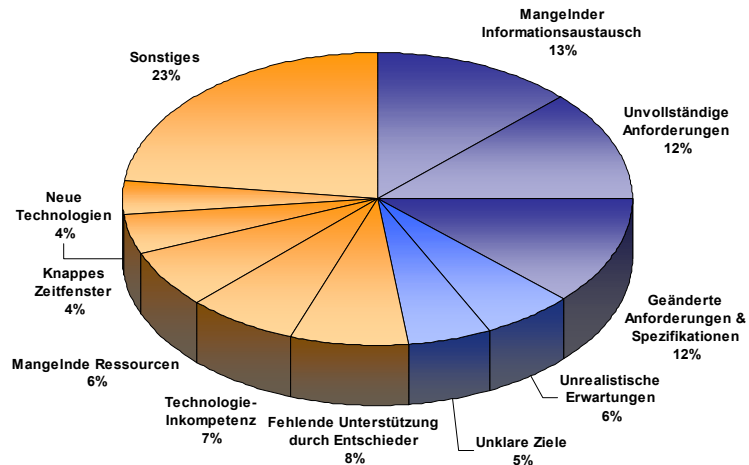
7

Gliederung

- Use Case Modelle
- Analyseergebnisse
- **Einführung, Grundlagen**
- Klassifizierungen
- Traceability
- CaliberRM Demo
- TogetherEC Demo

8

Faktoren zum Scheitern



Standish Group (www.standish.com)

9

Requirements - Definitionen

- Requirement (Anforderung)
 - Eine Aussage über eine Fähigkeit, physikalische Charakteristik oder einen Qualitätsfaktor, die einen Produkt- oder einen Prozeßbedarf beschreibt, für den eine Lösung bereitgestellt werden soll.
- Specification (Spezifikation)
 - Ein Dokument, das ein Produkt oder einen Prozess vollumfänglich mittels Requirements beschreibt und die entsprechenden Prüfbedingungen mitliefert.
- Constraint (Nebenbedingung)
 - Eine Begrenzung oder implizites Requirement, das das Produkt oder den Prozeß begrenzt und nicht durch Aktivitäten änderbar ist

IEEE Std. 1220-1994

10

Vorteile des Requirements Engineering



- Zielorientierung durch Transparenz
- Kostenersparnisse durch Vorverlagern der Fehlerentdeckung
- Validierung wird verbessert durch entstehende Stabilität
- Risikoärmere, weil schlankere Prozesse mit kleinen Releases
- Arbeitsteilung ermöglicht Spezialisierung
- Minimierung von Rollenkonflikten durch Zuständigkeiten

11

Meßbare Requirements



Meßbare Requirements formulieren

- Beispiel:
 - „90% aller Suchresultate sollen schnell angezeigt werden.“
- Indikatoren: minimal, maximal, schnell, leicht, ausreichend, angemessen....
- Besser wäre
 - „90% aller Suchresultate sollen innerhalb 1 Sekunde angezeigt werden.“

12

Kein Schwafeln

- Beispiel:
 - „Falls die genannten Eingaben von den aufgezählten Subsystemen in der richtigen Reihenfolge eintreffen, so daß das System die unterschiedlichen Quellen unterscheiden kann, soll das Ausgabesignal mit den entsprechenden Rahmenbedingungen aus Sektion 4.2.3 übereinstimmen, um den gewünschten Eingabezustand zu signalisieren.“
- Besser wäre
 - „Das Ausgabesignal soll mit Sektion 4.2.3 konform sein.“
 - „Das Ausgabesignal soll den gewünschten Eingabezustand anzeigen.“

13

Keine Vorgaben ans Design

- Beispiel:
 - „Das Sytem soll einen Auftrag in eine Rechnung überführen, indem es die Schnittstelle *createInvoice()* an der Session Facade aufruft und die Felder *aufNum04* und *totalBetr* übergibt.“
- Indikatoren: Nennung von Feldern, Komponenten, Objekten...
- Besser wäre
 - „Das System soll einen Auftrag in eine Rechnung überführen.“
 - „Die Rechnungspositionen enthalten automatisch die Werte der Auftragspositionen“.

14

Gliederung

- Use Case Modelle
- Analyseergebnisse
- Einführung, Grundlagen
- **Klassifizierungen**
- Traceability
- CaliberRM Demo
- TogetherEC Demo

15

Typen

- Der **Typ** von Requirements
 - primär
 - **ist Vertragsbestandteil, kommt von außerhalb des Teams**
 - abgeleitet
 - **kein Vertragsbestandteil, aber (zunächst) notwendige Konsequenz**
 - **Umformulierungen von Kunden-Requirements sind keine abgeleiteten Requirements.**
- hilft bei Fragen der Änderbarkeit

IEEE Std. 1220-1994

16

Kategoriensystem für Requirements

- Typ
- Anwendungsbereich
- Erfüllungsgrad
- Priorität
- Wichtigkeit
- Autor
- Interessensvertreter
- Eigner
- Quelle
- Kontext
- Thema
- Gültigkeitsbereich
- Überprüfungsmethode
- Überprüfungsstatus - stufe
- Status
- Reife
- Risiko
- Kosten
- Release

17

Formalisierungsgrad

- Der **Formalisierungsgrad** von Requirements
 - vorformal / textuell
 - **Jeder Motor ist nach Produktion Bestandteil eines Autos. Ein Auto hat immer einen Motor**
 - semiformal / grafisch Welchen würden Sie bevorzugen?

```
classDiagram
    class Auto
    class Motor
    Auto "1" -- "1" Motor : has a
```
 - formal / mathematisch logisch
 - $\forall m:\text{Motor} \exists! a:\text{Auto} (\text{teil_von}(m,a))$
 - $\forall a:\text{Auto} \exists! m:\text{Motor} (\text{teil_von}(m,a) \wedge \forall m_1:\text{Motor} (\text{teil_von}(m_1,a) \Rightarrow m_1=m))$

18

Gliederung

- Use Case Modelle
- Analyseergebnisse
- Einführung, Grundlagen
- Klassifizierungen
- **Traceability**
- CaliberRM Demo
- TogetherEC Demo

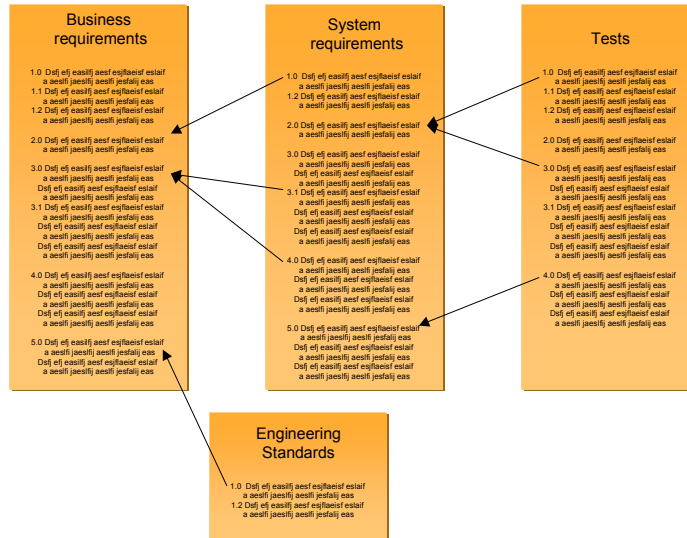
19

Fragestellungen

- Wie wurde dieses Requirement implementiert?
- Warum entwickle ich, was ich entwickle?
- Wieviel muß ich ändern, wenn ich ein Requirement ändere?
- Wurden alle Requirements umgesetzt?
- Bestehen für alles was entwickelt wurde Requirements?

20

Beispiel



21

Tools wofür?

- Begrifflichkeit:
 - Horizontale Werkzeuge:
 - für bestimmte Phasen oder -übergänge des Prozesse
 - Analyse-, Design-, Programmier-, Testwerkzeuge...
 - Vertikale Werkzeuge:
 - unterstützen mehrere Phasen oder den gesamten Prozess
 - Projektmanagement-, Konfigurations-, QS-Werkzeuge...
- Requirements Engineering Werkzeuge sind
 - horizontal und vertikal
 - daher hoher Integrationsbedarf

22

Gliederung

- Use Case Modelle
- Analyseergebnisse
- Einführung, Grundlagen
- Klassifizierungen
- Traceability
- **CaliberRM Demo**
- TogetherEC Demo

23

CaliberRM Demo

The screenshot displays the CaliberRM application window titled 'CaliberRM [admin] - Katalog durchsuchen'. The interface includes a menu bar (File, Edit, View, Insert, Format, Requirement, Tools, Reports, Help) and a toolbar. On the left, a project tree shows a hierarchy: 'Ist-Objekt-bibliothek' > '1. Projektauftrag (WHI)' > '2a. Funktionale Anforderungen (VHA)' > '2a.1 Katalog durchsuchen'. The main workspace is divided into several panes. The 'Responsibilities' pane is active, showing the following details for the use case 'Katalog durchsuchen':

- Alttexte:** Leser, Ausleihpersonal
- Ziel:** Der Benutzer kann in Bestandskatalog nach den vorhandenen Büchern der Bibliothek suchen.
- Ereignis:** Buchwunsch
- Voraussetzung:** - zugangsberechtigt zu Bestandskatalog vorhanden
- Ergebnis:** Der Benutzer besitzt die Information über die Verfügbarkeit seines Buchwunsches.

Below the main workspace, a text area contains the following description: 'Leser und Ausleihpersonal haben Zugriff auf den Katalog. Dieser zeigt den Bücherbestand mit bibliographischen Informationen über die Bücher sowie ihren Ausleihzustand (ausgeliehen bis... reserviert ja/nein)'. The bottom status bar shows 'RMX'.

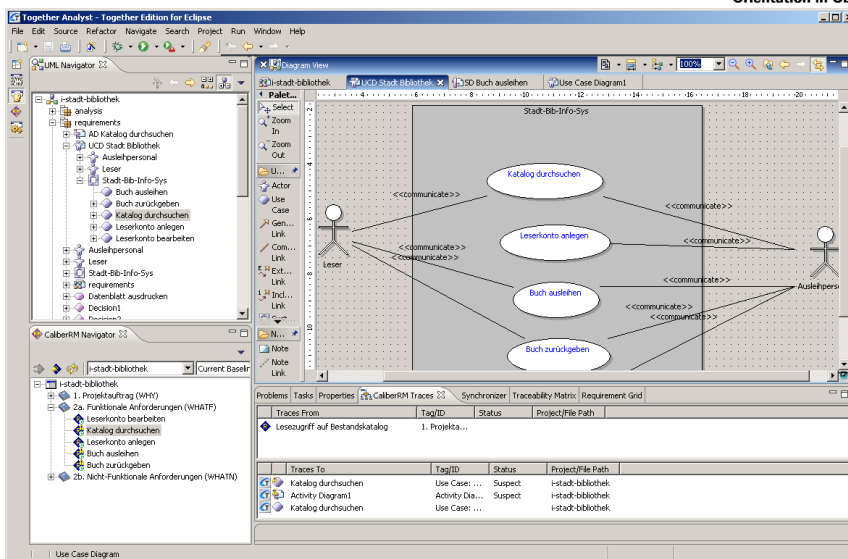
24

Gliederung

- Use Case Modelle
- Analyseergebnisse
- Einführung, Grundlagen
- Klassifizierungen
- Traceability
- CaliberRM Demo
- **TogetherEC Demo**

25

TogetherEC Demo



26



Fragen ?

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de

Version: 1.3



**Vielen Dank für Ihre
Aufmerksamkeit !**

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de

Version: 1.3