

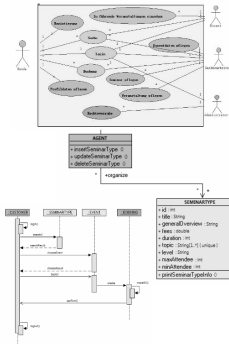
Gruppe 2: TOPCASED

Markus Schnalke
Veysel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu

22. Januar 2008

Markus Schnalke
Veyssel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu

Modellgetriebene Softwareentwicklung auf Basis von TOPCASED am Beispiel eines Seminarverwaltungssystems



```
public abstract class PERSONImpl extends EObjectImpl implements PERSON {

    protected static final String USERNAME_DEFAULT = null;
    protected String username = USERNAME_DEFAULT;

    public String getUsername() {
        return username;
    }

    public void setUsername(String newUsername) {
        String oldUsername = username;
        username = newUsername;
        if (eNotificationRequired())
            eNotify(new ENotificationImpl(this, Notification.SET, ClassID
    )

    public void login() {
        // TODO: implement this method
        // Ensure that you remove $generated or mark it $generated NOT
        throw new UnsupportedOperationException();
    }

    // abstract class PERSONImpl extends EObjectImpl implements PERSON {

    protected static final String USERNAME_DEFAULT = null;
    protected String username = USERNAME_DEFAULT;

    public String getUsername() {
        return username;
    }

    public void setUsername(String newUsername) {
        String oldUsername = username;
        username = newUsername;
        if (eNotificationRequired())
            eNotify(new ENotificationImpl(this, Notification.SET, ClassID
    )

    public void login() {
        // TODO: implement this method
        // Ensure that you remove $generated or mark it $generated NOT
        throw new UnsupportedOperationException();
    }
}
```

Entwicklungsprozess

Markus Schnalke

Anforderungen

Veysel Imamoglu

TOPCASED & Codegenerierung

Karl Oppermann

Modellierung & Templates

Rui Gu

Software Architektur

Dimitar Dimitrov

Demo

Christoph Galler

Risikomanagement

Nathalie Hrycej

Lessons Learned

Wei Zhu

Markus Schnalke
Veyssel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu

Entwicklungs-
Prozess

Anpassungen

Entwicklungsprozess

Markus Schnalke

Rational Unified Process

- ▶ weil wir Theoriewissen aus *Sote1* anwenden wollen
- ▶ muss angepasst werden
- ▶ das war gar nicht so leicht

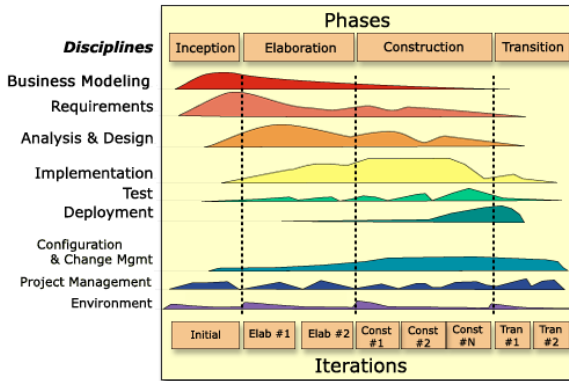
zur Erinnerung

Gruppe 2:
TOPCASED

Markus Schnalke
Veyssel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu

Entwicklungs-
Prozess

Anpassungen



Anpassungen

Gruppe 2:
TOPCASED

Markus Schnalke
Veyssel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu

Entwicklungs-
Prozess

Anpassungen

Zeitliche Dimension

- ▶ 3 Zyklen
- ▶ Phasen so gut es geht
- ▶ keine Iterationen

Inhaltliche Dimension

- ▶ stark abgespeckt
- ▶ jeder darf überall editieren

Anpassungen

Gruppe 2:
TOPCASED

Markus Schnalke
Veyssel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu

Entwicklungs-
Prozess

Anpassungen

Zeitliche Dimension

- ▶ 3 Zyklen
- ▶ Phasen so gut es geht
- ▶ keine Iterationen

Inhaltliche Dimension

- ▶ stark abgespeckt
- ▶ jeder darf überall editieren

Markus Schnalke
Veysel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu

Anforderungen

funktional

nicht funktional

Anforderungen

Veysel Imamoglu

Anforderungsanalyse

- ▶ Funktionale Anforderungen (FR)
- ▶ Nicht Funktionale Anforderungen (NFR)

Anforderungen

funktional

nicht funktional

Funktionale Anforderungen (FR)

Gruppe 2:
TOPCASED

Markus Schnalke
Veyssel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu

Funktionale Anforderungen

- ▶ Suchen
- ▶ Buchen
- ▶ Pflegen
- ▶ Stornieren
- ▶ etc.

Use Cases

Beschreibung mit Use Cases

Anforderungen

funktional

nicht funktional

Funktionale Anforderungen (FR)

Gruppe 2:
TOPCASED

Markus Schnalke
Veyssel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu

Funktionale Anforderungen

- ▶ Suchen
- ▶ Buchen
- ▶ Pflegen
- ▶ Stornieren
- ▶ etc.

Use Cases

Beschreibung mit Use Cases

Anforderungen

funktional

nicht funktional

Nicht Funktionale Anforderungen (NFR)

Gruppe 2:
TOPCASED

Markus Schnalke
Veyssel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu

Anforderungen

funktional

nicht funktional

Nicht Funktionale Anforderungen

- ▶ Bedienung mit Webbrowser
- ▶ Schutz gegen unauthorisierten Zugriff

Nicht Funktionale Anforderungen (NFR)

Gruppe 2:
TOPCASED

Markus Schnalke
Veyssel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu

Anforderungen

funktional

nicht funktional

Nicht Funktionale Anforderungen

- ▶ Bedienung mit Webbrowser
- ▶ Schutz gegen unauthorisierten Zugriff

NFR - Implementierungseinschränkung

- ▶ Modellgetriebene Softwareentwicklung
- ▶ Automatisierte Softwareentwicklung
- ▶ Wenig bis kein Handcrafted Code
- ▶ Transformation in eine beliebige Hochsprache
- ▶ Verwendung von TOPCASED

Anforderungen

funktional

nicht funktional

NFR - Implementierungseinschränkung

- ▶ Modellgetriebene Softwareentwicklung
- ▶ Automatisierte Softwareentwicklung
- ▶ Wenig bis kein Handcrafted Code
- ▶ Transformation in eine beliebige Hochsprache
- ▶ Verwendung von TOPCASED

Anforderungen

funktional

nicht funktional

TOPCASED & Codegenerierung

Karl Oppermann

Bedeutung

Steht für Toolkit in Open Source for Critical Applications & Systems Development.

Verwendung

Zur Anforderungsanalyse, Modellierung, Simulation, Implementierung, Test, Validierung und Projektmanagement . . .

- ▶ Luft- und Raumfahrttechnik

Software

Vorgehen

Generierung

Bedeutung

Steht für Toolkit in Open Source for Critical Applications & Systems Development.

Verwendung

Zur Anforderungsanalyse, Modellierung, Simulation, Implementierung, Test, Validierung und Projektmanagement ...

- Luft- und Raumfahrttechnik

Software

Vorgehen

Generierung

Bedeutung

Steht für Toolkit in Open Source for Critical Applications & Systems Development.

Verwendung

Zur Anforderungsanalyse, Modellierung, Simulation, Implementierung, Test, Validierung und Projektmanagement . . .

- ▶ Luft- und Raumfahrttechnik

Software

Vorgehen

Generierung

Ansatz

MDSD Model Driven Software Development

1. Modellierung
2. Transformation
3. Codegenerierung

- ▶ In TOPCASED enthalten, daher auch *integriertes Werkzeug* des MDSD

Software

Vorgehen

Generierung

Ansatz

MDSD Model Driven Software Development

1. Modellierung
2. Transformation
3. Codegenerierung

- In TOPCASED enthalten, daher auch *integriertes Werkzeug* des MDSD

Software

Vorgehen

Generierung

Ansatz

MDSD Model Driven Software Development

1. Modellierung
2. Transformation
3. Codegenerierung

- ▶ In TOPCASED enthalten, daher auch **integriertes Werkzeug** des MDSD

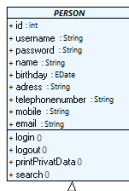
Software

Vorgehen

Generierung

Beispiel

Modell



Code

```
public abstract class PERSONImpl extends EObjectImpl implements PERSON {

    protected static final String USERNAME_EDEFAULT = null;
    protected String username = USERNAME_EDEFAULT;

    public String getUsername() {
        return username;
    }

    public void setUsername(String newUsername) {
        String oldUsername = username;
        username = newUsername;
        if (eNotificationRequired())
            eNotify(new ENotificationImpl(this, Notification.SET, ClassD
    }

    public void login() {
        // TODO: implement this method
        // Ensure that you remove @generated or mark it @generated NOT
        throw new UnsupportedOperationException();
    }
}
```

Gruppe 2:
TOPCASED

Markus Schnalke
Veysel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu

Software

Vorgehen

Generierung

Was?

Ist die auto. Erzeugung von Quellcode in einer best. Programmiersprache

Wie?

Ein Codegenerator übersetzt ein Modell

Generator?

Ein Computerprogramm für die Softwareentwicklung.

- ▶ Input: Modelle in einer Meta-Sprache (wie UML, ..)
- ▶ Output: Eine Programmiersprache (wie Java, ..), also Quellcode

Software

Vorgehen

Generierung

Was?

Ist die auto. Erzeugung von Quellcode in einer best. Programmiersprache

Wie?

Ein **Codegenerator** übersetzt ein Modell

Generator?

Ein Computerprogramm für die Softwareentwicklung.

- ▶ Input: Modelle in einer Meta-Sprache (wie UML, ..)
- ▶ Output: Eine Programmiersprache (wie Java, ..), also Quellcode

Software

Vorgehen

Generierung

Was?

Ist die auto. Erzeugung von Quellcode in einer best. Programmiersprache

Wie?

Ein **Codegenerator** übersetzt ein Modell

Generator?

Ein Computerprogramm für die Softwareentwicklung.

- ▶ Input: Modelle in einer Meta-Sprache (wie UML, ..)
- ▶ Output: Eine Programmiersprache (wie Java, ..), also Quellcode

Software

Vorgehen

Generierung

Markus Schnalke
Veyssel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu

UML Modellierung
und Generierung

Klassendiagramm

Sequenzdiagramm

Aktivitätsdiagramm

Generierungsprozess

Template(xpt)

Workflow(oaw)

Modellierung & Templates

Rui Gu

UML Modellierung und Generierung

Gruppe 2:
TOPCASED

Markus Schnalke
Veyssel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu

UML-Diagramm

Klassendiagramm, Sequenzdiagramm,
Aktivitätsdiagramm

Generierungsprozess

Template(xpt) und Workflow(oaw)

UML Modellierung
und Generierung

Klassendiagramm

Sequenzdiagramm

Aktivitätsdiagramm

Generierungsprozess

Template(xpt)

Workflow(oaw)

UML Modellierung und Generierung

Gruppe 2:
TOPCASED

Markus Schnalke
Veyssel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu

UML-Diagramm

Klassendiagramm, Sequenzdiagramm,
Aktivitätsdiagramm

Generierungsprozess

Template(xpt) und Workflow(oaw)

UML Modellierung
und Generierung

Klassendiagramm

Sequenzdiagramm

Aktivitätsdiagramm

Generierungsprozess

Template(xpt)

Workflow(oaw)

UML Modellierung und Generierung

Gruppe 2:
TOPCASED

Markus Schnalke
Veyssel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu

UML-Diagramm

Klassendiagramm, Sequenzdiagramm,
Aktivitätsdiagramm

Generierungsprozess

Template(xpt) und Workflow(oaw)

UML Modellierung
und Generierung

Klassendiagramm

Sequenzdiagramm

Aktivitätsdiagramm

Generierungsprozess

Template(xpt)

Workflow(oaw)

Klassendiagramm

Gruppe 2:
TOPCASED

Markus Schnalke
Veysel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu

UML Modellierung
und Generierung

Klassendiagramm

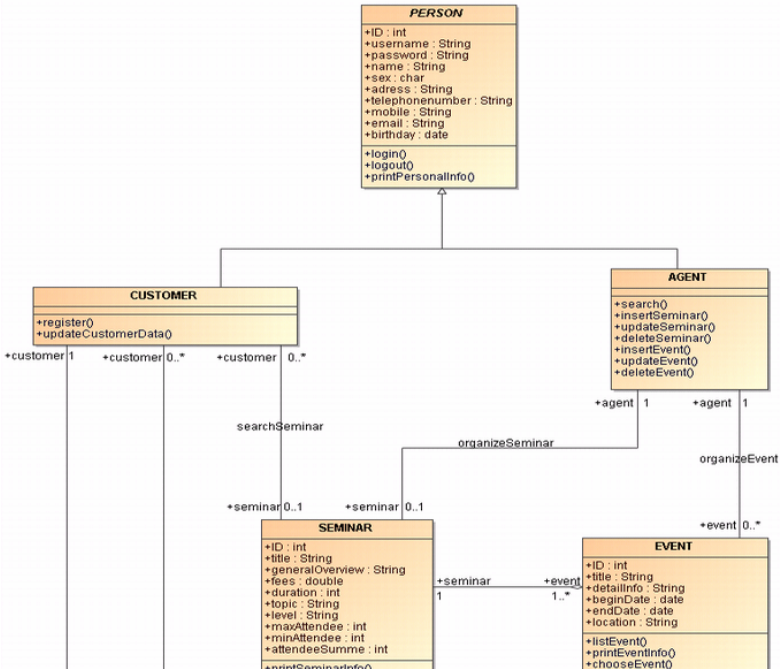
Sequenzdiagramm

Aktivitätsdiagramm

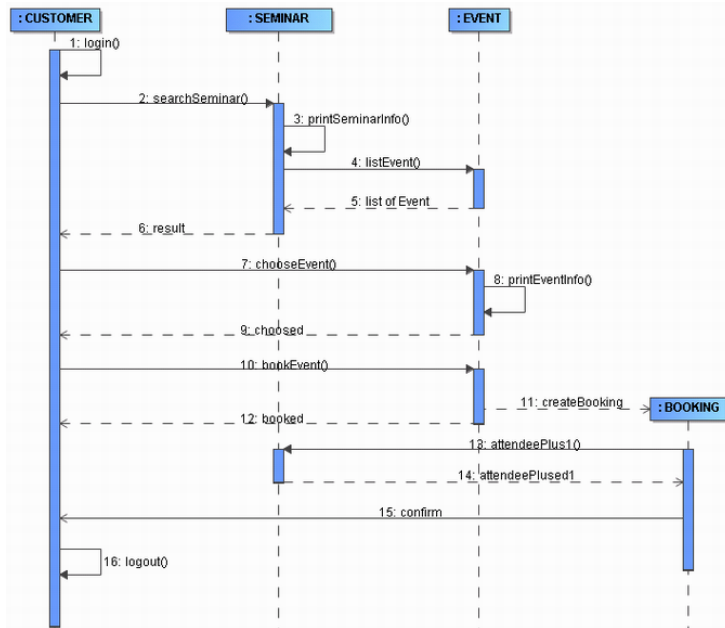
Generierungsprozess

Template(xpt)

Workflow(oaw)



Sequenzdiagramm



Gruppe 2:
TOPCASED

Markus Schnalke
Veyssel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu

UML Modellierung
und Generierung

Klassendiagramm

Sequenzdiagramm

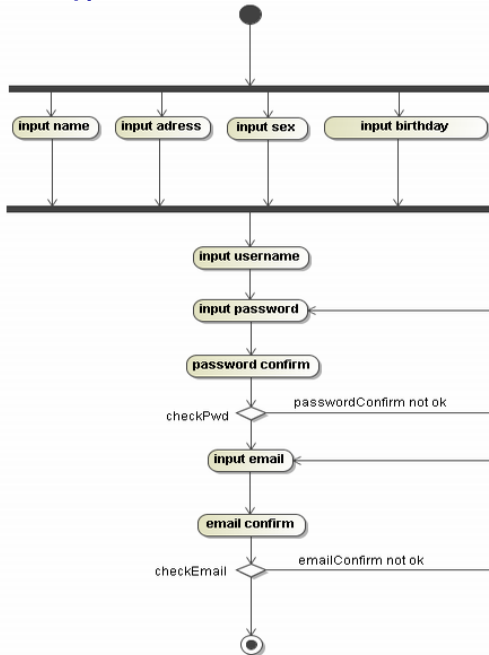
Aktivitätsdiagramm

Generierungsprozess

Template(xpt)

Workflow(oaw)

Aktivitätsdiagramm



Gruppe 2:
TOPCASED

Markus Schnalke
Veysel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu

UML Modellierung
und Generierung

Klassendiagramm

Sequenzdiagramm

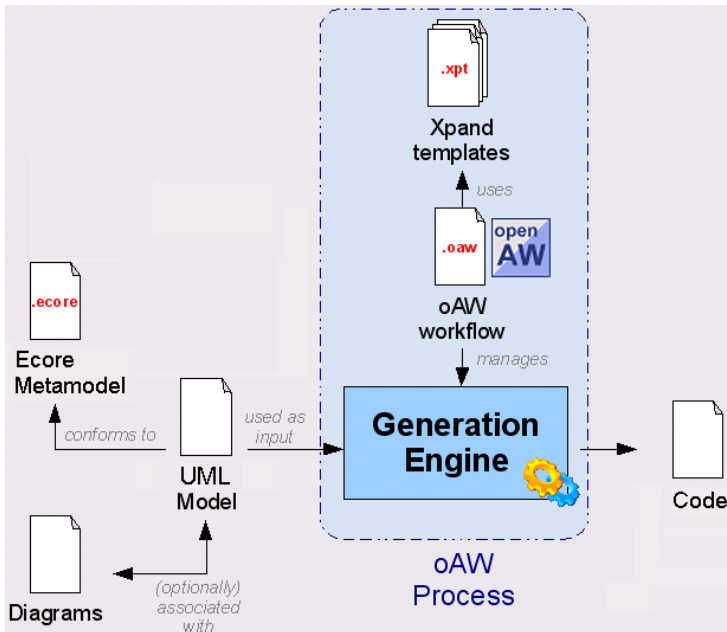
Aktivitätsdiagramm

Generierungsprozess

Template(xpt)

Workflow(oaw)

Generierungsprozess



Gruppe 2:
TOPCASED

Markus Schnalke
Veyssel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu

UML Modellierung
und Generierung

Klassendiagramm

Sequenzdiagramm

Aktivitätsdiagramm

Generierungsprozess

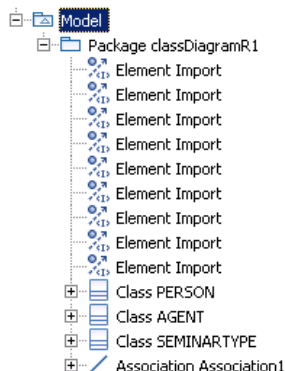
Template(xpt)

Workflow(oaw)

Template(xpt)

```
«DEFINE Root FOR uml::Model»  
  «EXPAND PackageRoot FOREACH ownedElement»  
«ENDDDEFINE»  
  
«DEFINE PackageRoot FOR uml::Package»  
  «EXPAND ClassRoot FOREACH ownedType»  
«ENDDDEFINE»  
  
«DEFINE ClassRoot FOR uml::Class»  
  «FILE name+ ".java"»  
    public class «name» {}  
  «ENDFILE»  
«ENDDDEFINE»  
  
«DEFINE Root FOR uml::Element»«ENDDDEFINE»  
«DEFINE PackageRoot FOR uml::Element»«ENDDDEFINE»  
«DEFINE ClassRoot FOR uml::Element»«ENDDDEFINE»
```

template root.xpt



UML Modelbaum

Gruppe 2:
TOPCASED

Markus Schnalke
Veyssel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu

UML Modellierung
und Generierung

Klassendiagramm

Sequenzdiagramm

Aktivitätsdiagramm

Generierungsprozess

Template(xpt)

Workflow(oaw)

Workflow(oaw)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<workflow>
<bean class="oaw.uml2.Setup" standardUML2Setup="true"/>
<component class="oaw.emf.XmiReader">
<modelFile value="classDiagramR1.uml"/>
<outputSlot value="model"/>
</component>
```

```
<component id="dirCleaner"
class="oaw.workflow.common.DirectoryCleaner"
directories="src-gen"/>
```

```
<component id="generator" class="oaw.xpand2.Generator"
skipOnError="true">
```

```
<metaModel class="oaw.type.emf.EmfMetaModel">
<metaModelPackage value="org.eclipse.emf.ecore.EcorePackage"/>
</metaModel>
<metaModel class="oaw.uml2.UML2MetaModel"/>
```

```
<expand value="templates::Root::Root FOR model"/>
<outlet path="src-gen"/>
<postprocessor class="oaw.xpand2.output.JavaBeautifier"/>
</outlet>
</component> </workflow>
```

Markus Schnalke
Veyssel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu

Software
Architecture

Logical View

Data View

Software Architektur

Dimitar Dimitrov

Software Architecture

Gruppe 2:
TOPCASED

Markus Schnalke
Veyssel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu

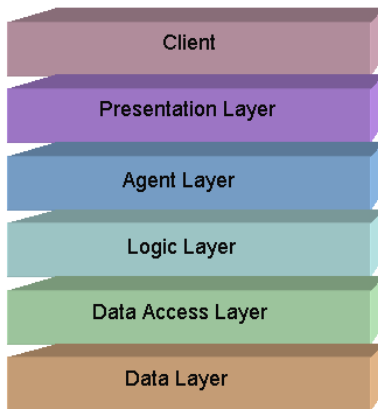
1. Architectural Representation
2. Architectural Goals and Constraints
3. Logical view
4. Implementation view
5. Data view

Software
Architecture

Logical View

Data View

Logical View



Gruppe 2:
TOPCASED

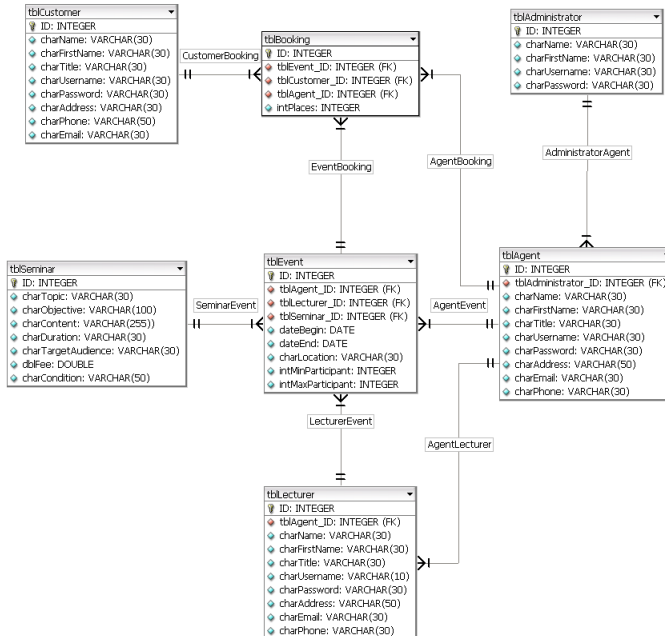
Markus Schnalke
Veysel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu

Software
Architecture

Logical View

Data View

Data View



Gruppe 2:
TOPCASED

Markus Schnalke
Veysel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu

Software
Architecture
Logical View
Data View

Markus Schnalke
Veyssel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu

Demonstration

Demo

Christoph Galler

Demonstration unseres Prototypen

- ▶ kein generierter Code
- ▶ funktional aber passend
- ▶ ist Zielvorgabe für Codegenerierung
- ▶ technisch Probleme finden

Demonstration

Markus Schnalke
Veyssel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu

Risikomanagement

Nathalie Hrycej

Risikomanagement

Rolle des
Risikomanagements
Warum ist
Risikomanagement wichtig?

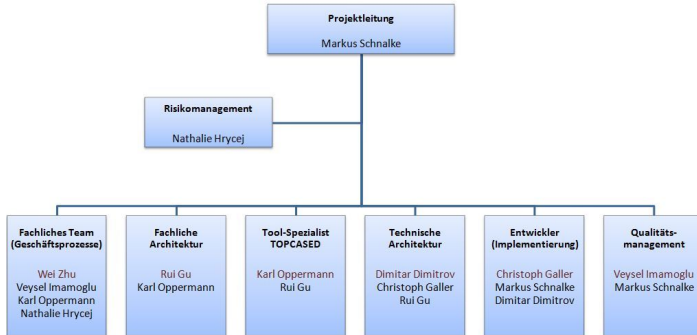
Risiken

Technologie
Zeit
Kommunikation

Rolle des Risikomanagements

Gruppe 2:
TOPCASED

Markus Schnalke
Veysel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu



Risikomanagement

Rolle des
Risikomanagements

Warum ist
Risikomanagement wichtig?

Risiken

Technologie

Zeit

Kommunikation

Warum ist Risikomanagement gerade für uns wichtig?

Gruppe 2:
TOPCASED

Markus Schnalke
Veyssel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu

Risikomanagement

Rolle des
Risikomanagements

Warum ist
Risikomanagement wichtig?

Risiken

Technologie

Zeit

Kommunikation

- ▶ Unerfahrenheit mit Projektumsetzung
- ▶ Einsatz unbekannter Technologien
- ▶ Arbeitsweise nach dem RUP

Risiko: Neue Technologien

Gruppe 2:
TOPCASED

Markus Schnalke
Veyssel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu

Problem

- ▶ Einsatz neuer und unbekannter Technologien

Auswirkung

- ▶ Lange Einarbeitungszeit

Lösung

- ▶ Bildung von Spezialistenteams unter Berücksichtigung der vorhandenen Kenntnisse

Risikomanagement

Rolle des
Risikomanagements

Warum ist
Risikomanagement wichtig?

Risiken

Technologie

Zeit

Kommunikation

Risiko: Zeit

Gruppe 2:
TOPCASED

Markus Schnalke
Veyssel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu

Problem

- ▶ Umfangreiches Projekt in kurzer Zeit

Auswirkung

- ▶ Gefährdung der Projektzielerreichung

Lösung

- ▶ Realistische Zielsetzung

Risikomanagement

Rolle des
Risikomanagements

Warum ist
Risikomanagement wichtig?

Risiken

Technologie

Zeit

Kommunikation

Problem

- ▶ Fehlende Kommunikationsstruktur

Auswirkung

- ▶ Schlechter Informationsfluss
- ▶ Mangelnde Koordination

Lösung

- ▶ Einrichtung eines Mailverteilers
- ▶ Regelmäßige Team- und Statusmeetings

Risikomanagement

Rolle des
Risikomanagements
Warum ist
Risikomanagement wichtig?

Risiken

Technologie
Zeit
Kommunikation

Markus Schnalke
Veyssel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu

Lessons Learned

Wei Zhu

Was ist Lessons
Learned

Was haben wir
gelernt oder
gefunden

Was haben wir
gelernt oder
gefunden

Was waren die
grössten Probleme

Was sind unsere
Erfolge

Was würden wir
beim nächsten Mal
anders machen

Was ist Lessons Learned

Gruppe 2:
TOPCASED

Markus Schnalke
Veyssel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu

Definition

Darstellung durch Erfahrung erworbenes Wissen oder Verständnis

Zweck

- ▶ Lernen aus den Erfahrungen anderer
- ▶ Selbstreflexion der Teammitglieder

Was ist Lessons Learned

Was haben wir gelernt oder gefunden

Was haben wir gelernt oder gefunden

Was waren die grössten Probleme

Was sind unsere Erfolge

Was würden wir beim nächsten Mal anders machen

Was ist Lessons Learned

Gruppe 2:
TOPCASED

Markus Schnalke
Veyssel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu

Definition

Darstellung durch Erfahrung erworbenes Wissen oder Verständnis

Zweck

- ▶ Lernen aus den Erfahrungen anderer
- ▶ Selbstreflexion der Teammitglieder

Was ist Lessons Learned

Was haben wir gelernt oder gefunden

Was haben wir gelernt oder gefunden

Was waren die grössten Probleme

Was sind unsere Erfolge

Was würden wir beim nächsten Mal anders machen

Was haben wir gelernt oder gefunden

- ▶ Teamfähigkeit
- ▶ Eigene Stärke und Schwäche
- ▶ Softwareprojekt entwerfen, realisieren und dokumentieren
- ▶ Entwurfserfahrung sammeln
- ▶ Unsicherheiten frühzeitig erkennen und korrigieren

Was ist Lessons
Learned

Was haben wir
gelernt oder
gefunden

Was haben wir
gelernt oder
gefunden

Was waren die
größten Probleme

Was sind unsere
Erfolge

Was würden wir
beim nächsten Mal
anders machen

Was haben wir gelernt oder gefunden

- ▶ Nicht alles generieren
- ▶ TOPCASED nur die Generierung des Klassenrumpfes
- ▶ Zusätzliche Templates
- ▶ Unterstützt den UML2-Editor, jedoch keine Abhängigkeit zwischen einzelnen Diagramme
- ▶ Zusätzlich openArchitecture-Ware (oAW)

Was ist Lessons Learned

Was haben wir gelernt oder gefunden

Was haben wir gelernt oder gefunden

Was waren die grössten Probleme

Was sind unsere Erfolge

Was würden wir beim nächsten Mal anders machen

Was waren die größten Probleme

- ▶ Mangelndes technisches Know-How
- ▶ Mangelnde Information
- ▶ Unreife Technologie bezüglich TOPCASED
- ▶ Termine zu eng

Was ist Lessons
Learned

Was haben wir
gelernt oder
gefunden

Was haben wir
gelernt oder
gefunden

Was waren die
größten Probleme

Was sind unsere
Erfolge

Was würden wir
beim nächsten Mal
anders machen

Was sind unsere Erfolge

- ▶ Vollständige funktionale Anforderungen mit UML
- ▶ Ausführliche Protokolle
- ▶ Projektplanung mit Releaseterminen
- ▶ Template-Dateien funktionieren teilweise
- ▶ Das Datenbank-Modell ist anhand des Dokuments Usecase vollständig erstellt
- ▶ Rechtzeitige Identifikation und realistische Bewertung von Risiken
- ▶ Die Tutorials wurden gleichzeitig mit neuen Erkenntnissen von den Mitgliedern erstellt

Was ist Lessons Learned

Was haben wir gelernt oder gefunden

Was haben wir gelernt oder gefunden

Was waren die grössten Probleme

Was sind unsere Erfolge

Was würden wir beim nächsten Mal anders machen

Was würden wir beim nächsten Mal anders machen

- ▶ Klare Abgrenzung der Aufgaben und Kompetenzen
- ▶ Genügend Kommunikation
- ▶ Überwachung und Steuerung des Projektes
- ▶ Aufwand richtig einschätzen

Was ist Lessons
Learned

Was haben wir
gelernt oder
gefunden

Was haben wir
gelernt oder
gefunden

Was waren die
grössten Probleme

Was sind unsere
Erfolge

Was würden wir
beim nächsten Mal
anders machen

Markus Schnalke
Veyssel Imamoglu
Karl Oppermann
Rui Gu
Dimitar Dimitrov
Christoph Galler
Nathalie Hrycej
Wei Zhu

Danke für Ihre Aufmerksamkeit