

Workshop

Modellgetriebene Software-Entwicklung

★ Effiziente Software-Entwicklung mit Modellen

Für CTOs, Teamleiter und Architekten

Zielgruppe:

- ★ Leiter von Teams, die modellgetriebene Entwicklungsmethoden einführen oder ausweiten wollen.

Aus dem Inhalt:

- ★ Modelle: Abstraktion und Konkretisierung
- ★ Zyklus des Wissensmanagements:
Explication, Formalisierung, Lösung, Veränderung
- ★ Modellgetriebene Entwicklung und V-Modell
- ★ Modellgetriebene Entwicklung und Requirement Engineering
- ★ Modellgetriebene Entwicklung und Spezifikation
- ★ Modellgetriebene Entwicklung und Domänenspezifische Sprachen
- ★ Modellierung und Arbeitsteilung
- ★ Team-Aufstellung:
SE-Team, Stakeholder, Zulieferer, Manager, Kunden
- ★ Aufgabenprofile im Entwicklungsteam:
Projektleiter, Funktionsverantwortlicher, Produkt-Designer,
Spezifikateur, Modellierer, Requirement Engineer, Tester,
Qualitätssicherer, Programmierer, Tool-Crafter, Architekt,
Methodiker
- ★ Co-Evolution von Modell und Modellsprache, Versionsverwaltung
und Migration der Modellierungsumgebung, Qualitätssicherung
von Modell und Modellsprache
- ★ Modell-Standards und -Technologien (z.B. UML, MOF, DSL,
MDS, MDA, EMF, XML)
- ★ Modellierungs-Praxis
Szenarien: HMI-Modelle, Workflow- & Zustandsmodelle
Aspekte: Änderungen, Konfiguration, Software-Familien

Motto:

*Make life more simple: separate
concerns, integrate solutions.*

*Kernpunkte: MDA, MDSD,
Modellieren, Verarbeiten und
Auswerten von Modellen, Umfang
mit Metamodellen und
domänenspezifischen Sprachen,
Code-Generierung,
Automatisierung von Tests*

Gliederung:

- ✦ Überblick über modellgetriebene Methoden:
Was sind Modelle? Wozu sind sie gut?
- ✦ Einbinden von Modellen in Entwicklungsprozesse
- ✦ Stand der Technik: Tools, Methoden, Standards
- ✦ Das Query-View-Transformation Paradigma

- ✦ Modellieren
Aufsetzen eines Modells
Unterscheiden zwischen Modellierung und Metamodellierung
Modelle integrieren
Verifikation und Validierung
Qualitätssicherung

- ✦ Anwenden der Modelle
Praxis der modellgetriebenen Software-Entwicklung:
editieren, visualisieren, transformieren, analysieren, verifizieren,
Code und Steuerungsdaten generieren, testen, Automatisieren
von Tests

- ✦ Wrap Up:
Aktuelle Entwicklungen, Typische Stolpersteine beim Einführen
und Anwenden von Modellen, Bewertung der Methoden und
Techniken aus Teilnehmersicht, Folge-Szenarien

Material auf CD:

- ☑ *Workshopunterlagen*
- ☑ *Checklisten und Roadmaps*
- ☑ *Beispielmodelle für User
Interfaces und Workflows*
- ☑ *Transformationsbeispiele*
- ☑ *Installationsimages freier
MDA Tools*

*Material im Workshop: Eclipse,
EMF, Magic Draw*

*Organisation: 4-8 Teilnehmer,
Schulungsraum mit Internet-
Zugang, Digital-Projektor, Flip
Chart, Schulungsrechner oder
Notebooks der Teilnehmer*

*Anmeldung:
[http://www.eospp.de/leistungen/
workshopanmeldung.html](http://www.eospp.de/leistungen/workshopanmeldung.html)*

Workshop

Entwickeln von User Interfaces

- ★ User Interface Konzepte anhand von Beispielen
- ★ Wie baut man ein User Interface auf?
- ★ Wie setzt man ein UI-Konzept um?
- ★ Wie entwickelt man User Interfaces für konkrete Verwendungssituationen?
- ★ Wie verhält sich der UI-Entwicklungs-Aufwand zu Produktnutzen und -kosten?

Für Informations-Architekten, Software-Architekten, User Interface Entwickler und Requirement-Engineers

Zielgruppe:

- ★ Entwicklerteams von Standardanwendungen.
- ★ Hersteller und Auftraggeber von dialogintensiven Individualsoftwareanwendungen.

Aus dem Inhalt:

- ★ User Interfaces als Mittel zum Zweck
- ★ UI-Formen sowie Anzeige-und-Bedienkonzepte
- ★ Roadmap für das Entwickeln eines UI
- ★ Interaktion zwischen Anwender und User Interface
- ★ Informationsmodell für ein User Interface
- ★ Usability-Faktoren
- ★ Prozesse: Requirement-, Konzept-, Umsetzungs-, Bereitstellungs-, Release-, Konfigurations-, Änderungs- und Testmanagement
- ★ Methoden: Systemgrenzenbestimmung, UC-Analyse, Workflow-Analyse, Dialogskizze, Dialogentwurf, Bedienmetaphern-Analyse, Auswahl von Kontrollelementen
- ★ Werkzeuge: Spezifikationseditoren, Modellierungs-Tools, Design-Tools, Prototyping-Tools, Test-Tools, Kommunikations-Tools
- ★ Praxisbeispiele: Online-Shop, Kunden- und Vorgangsverwaltung, Routenplaner

Ein User Interface ist das, womit wir ein Werkzeug oder eine Maschine steuern und kontrollieren, z.B. der Cockpit eines Fahrzeugs, Dialogfenster, Widgets, Sprachinterpret, der Griff einer Schaufel, der Henkel einer Tasse, u.v.m.

Gliederung:

Einführung

- ★ Wie man zwischen der „Benutzeroberfläche“ und dem Rest einer Software unterscheidet
- ★ Welchen User Interface Arten man im Alltag und in der Forschung begegnet
- ★ Welche Etappen die Entwicklung eines User Interfaces beinhaltet und wie man mit dem Entwickeln startet
- ★ Typische Stolpersteine und Vorbeugungsmassnahmen
- ★ „Unter der Oberfläche“: Interaktionskontext, Informationsmodelle, Bedienmetaphern, Usability-Faktoren

Anwenden von Methoden und Werkzeugen

- ★ Systemgrenzen-Analyse
- ★ Use-Case Analyse
- ★ Dialog-Fluss-Struktur-Analyse
- ★ Dialogseiten-Entwurf
- ★ Panel-Layout-Entwurf
- ★ Auswahl und Styling von Kontrollelementen
- ★ Prototyping, Kommunikation und Bestätigung
- ★ Variantenhandling, Versionskontrolle
- ★ Anforderungs- und Änderungsmanagement
- ★ Spezifikationsprache, Modellintegration
- ★ Testfälle, Testbereitstellung, Testmanagement, Defect Tracking
- ★ Integration von Ergebnissen aus Usability Tests

Der Workshop ist als Praxisarbeit an konkreten Anwendungsfällen ausgelegt. Neben den Beispielen des Dozenten werden auch die von den Teilnehmern mitgebrachten Dialogseiten und Workflows bearbeitet.

Material auf CD:

- Workshopunterlagen*
- Checklisten und Roadmaps*
- Installationsimages freier Softwarewerkzeuge*

Material im Workshop: MS Office (auf Schulungsrechner), Open Office, Issue Tracker, SVN Server, Tortoise

Organisation: 4-8 Teilnehmer, Schulungsraum mit Internet-Zugang, Digital-Projektor, Flip Chart, Schulungsrechner oder Notebooks der Teilnehmer

Anmeldung:

<http://www.eospp.de/leistungen/workshopanmeldung.html>

Workshop

Usability Engineering

- ★ Usability-Faktoren anhand von Beispielen
- ★ Was nützt Usability und wie misst man sie?
- ★ Wie funktionieren Usability-Analyse und -Engineering?
- ★ Wie kann man Usability anhand von konkreten Fällen entwickeln?
- ★ Wie verhält sich Usability-Aufwand zu Produktnutzen und -kosten?

Für IT-Leiter, Software-Architekten und User Interface Entwickler

Zielgruppe:

- ★ Unternehmen, die Qualitäts-Software entwickeln.
- ★ Unternehmen, die Softwarehäuser mit dem Erstellen von dialogintensiver Individualsoftware beauftragen.

Usability ist das Maß für die Gebrauchstauglichkeit (einer Software).

Aus dem Inhalt:

- ★ Usability als Erfolgsfaktor eines Produkts
- ★ Messkriterien für Usability
- ★ Methoden zum Verbessern der Usability
- ★ Psychologische Faktoren der Usability-Entwicklung
- ★ Übungen anhand von Praxisfällen

*Messkriterien
Entwicklungsmethoden
Praxisfälle*

Gliederung:

Vorbereitungsteil

- ✦ Grundlagen der Usability
- ✦ Unterscheiden von Funktionalität, Form und Verwendungsaspekten in einer Software
- ✦ Objektive und subjektive Usability-Kriterien
- ✦ Usability bewerten: Bewertungsschema, Einzel-Reviews, Peer-Analysen, User-Tests, Interpretieren der Ergebnisse
- ✦ Usability verbessern: Verbesserungspotentiale identifizieren, Stolpersteine beim Betrachten von persönlichen Usability Aspekten, Lösungen erarbeiten, abstimmen und präsentieren.

Hauptteil

- ✦ Werkzeuge und Methoden
 - ✦ Übungen anhand von Praxisfällen
 - ✦ Startseiten von Anwendungen
 - ✦ Navigationsleisten
 - ✦ Dialogseiten
 - ✦ Kontrollelemente und ihre Verwendungsmetaphern
 - ✦ Workflow-Führung
-
- ✦ Der Workshop ist als Praxisarbeit an konkreten Anwendungsfällen ausgelegt. Neben den Beispielen des Dozenten werden auch die von den Teilnehmern mitgebrachten Dialogseiten und Workflows bearbeitet.

Material auf CD:

- Workshopunterlagen*
- Checklisten und Roadmaps*
- Installationsimages freier Office Bausteine*

Material im Workshop: MS Office (auf Schulungsrechner), Open Office, Issue Tracker, SVN Server, Tortoise, Thunderbird Mail, IMAP/POP3 Server

Organisation: 4-8 Teilnehmer, Schulungsraum mit Internet-Zugang, Digital-Projektor, Flip Chart, Schulungsrechner oder Notebooks der Teilnehmer

Anmeldung:

<http://www.eospp.de/leistungen/workshopanmeldung.html>

Workshop

Projekte organisieren mit CMMI

- ★ Methoden und Werkzeuge zur Organisation von Informationen, Dokumenten und Daten
- ★ Wie steuere ich effektiv mein Projekt-Team?
- ★ Wie Sorge ich dafür, dass Anforderungen, Fehlermeldungen, Aufgaben und Änderungswünsche gesammelt und koordiniert abgearbeitet werden?
- ★ Wie behalte ich den Überblick über Dateien, Quellcode, Dokumente, Versionen, Berichte und Aufgaben?
- ★ Wie organisiere ich Veröffentlichungen und Versionen?

Für Projektleiter, Architekten und Chefentwickler

Zielgruppe:

- ★ Projektteams, Projektleiter, Architekten, Qualitätsmanager, Konfigurationsmanager

Aus dem Inhalt:

- ★ Rahmenbedingungen der Produktentwicklung: Es wird etwas entwickelt, das im definierten Zeitrahmen im festgelegten Budget und Qualität/Features an einen internen oder externen Kunden geliefert werden soll.
- ★ Übertragen der CMMI Prozesse in die Projektpraxis
- ★ Management von Änderungen, Aufgaben und Fehlern mit einem Issue-Tracker (am Beispiel von Roundup-Tracker)
- ★ Verteilen, Versionieren, Änderungskontrolle und Sichern der Projektdokumente mit einem Versionskontrollsystem (am Beispiel von Subversion mit Tortoise Shell)
- ★ Fortschreiben und Präsentieren des Projektfortschritts mit einem Blog-Werkzeug (am Beispiel eines Projekt-Wikis)
- ★ Release-Management: Hauptlinie und Branches
- ★ Vorgehensweise: V-Modell, CMMI oder Muddling-Through?
- ★ Verteiltes Projektteam: Die Teammitglieder sitzen an verschiedenen Orten.

Motto: Projektküche. Das Rezept für das zu erstellende Produkt ist klar. Zutaten und Leute auch. Aber was für Küchengeräte nimmt man und wie setzt man sie effektiv ein?

Kernpunkte: Ablagen, Versionen, Aufgaben, Tipps und Tricks für die Arbeitsorganisation, Arbeits- und Veröffentlichungsstrukturen

Gliederung:

- ★ Analyse der Produktstruktur und der Kommunikationswege: Was liefert das Projekt an wen? Was wird dem Projekt von wem zugeliefert. Entwickeln einer Ablagestruktur für Arbeits- und Veröffentlichungsstände des Projekts.
- ★ Aufsetzen einer Projekt-Struktur im Dateibaum; Verknüpfung der Work-Area mit dem Projektrepository
- ★ Verwalten von Dokumenten, Versionen und Ablagen
- ★ Issues: Management von Anforderungen, Aufgaben, Änderungen und Defects

- ★ Arbeiten mit Groupware-Werkzeugen (am Beispiel von Outlook)
- ★ Kommunikation der Projekthinhalte und des Projektfortschritts
- ★ Aufsetzen eines Projekt-Wikis
- ★ Releases: koordinierte Bereitstellung von Ergebnissen

Durchführung:

- ★ Anhand eines Übungsprojekts wird der Einsatz der Werkzeuge Subversion, Wiki und Roundup in verschiedenen Projekt-Szenarien durchgespielt.
- ★ Die Teilnehmer erlernen gleichzeitig das Benutzen der genannten Werkzeuge, sowie das notwendige Prozess-KnowHow, um ihre IT-Projekte erfolgreich durchzuführen.
- ★ Die Schulung orientiert sich an den Arbeitsweisen der CMMI, bietet aber statt prinzipieller Forderungen konkrete, und sofort einsetzbare Lösungen in Form von Methoden und Werkzeugen, um ein IT-Projekt zu steuern.

Material:

- ☑ *Workshopunterlagen*
- ☑ *Checklisten und Roadmaps*
- ☑ *Zugang zu einem Server mit den vorinstallierten Tools*
- ☑ *Die Tools sind allesamt Open-Source, Lizenzkosten fallen nicht an.*

Organisation: 4-8 Teilnehmer, Schulungsraum mit Internet-Zugang, Digital-Projektor, Flip Chart, Schulungsrechner oder Notebooks der Teilnehmer

Anmeldung:

<http://www.eospp.de/leistungen/workshopanmeldung.html>