Zurich University of Applied Sciences

School of Engineering

Bachelor of Science (BSc) in Informatik

Modul Advanced Software Engineering 1 (ASE1)

Software Architektur

Werkzeuge

Institut für Angewandte Informationstechnologie (InIT)

Walter Eich (eicw) / Matthias Bachmann (bacn)

https://www.zhaw.ch/de/engineering/institute-zentren/init/

Agenda

Zurich University of Applied Sciences

School of Engineering

- 6.1. Lernziele
- 6.2. Allgemeine Hinweise
- 6.3. Werkzeuge zum Anforderungsmanagement
- 6.4. Werkzeuge zur Modellierung
- 6.5. Werkzeuge zur Generierung
- 6.6. Werkzeuge zur statischen Codeanalyse
- 6.7. Werkzeuge zur dynamischen Analyse
- 6.8. Werkzeuge zum Build-Management
- 6.9. Werkzeuge zum Konfigurations- und Versionsmanagement
- 6.10. Werkzeuge zum Codemanagement
- 6.11. Werkzeuge zum Test
- 6.12. Werkzeuge zur Dokumentation

Lernziele

- LZ 5–1: Wichtige Werkzeugkategorien benennen und einordnen
- LZ 5–2: Werkzeuge bedarfsgerecht auswählen

Allgemeine Hinweise

Zurich University of Applied Sciences
School of Engineering

- Für die Prüfung zum CPSA-F müssen keine konkreten Vertreter typischer Werkzeuge oder Produkte genannt werden.
- Es geht für die Prüfung primär um Werkzeugkategorien und Entscheidungskriterien
- Werkzeuge sollen ihre jeweiligen Aufgaben
  - funktional umfänglich und zuverlässig erledigen,
  - geringe Auswirkungen an anderen Stellen besitzen.
- Wichtig bei Werkzeugen
  - Kosten
  - Lizenzbedingungen.

Kategorien

- Werkzeuge zum Anforderungsmanagement
- Werkzeuge zur Modellierung
- Werkzeuge zur Generierung
- Werkzeuge zur statischen Codeanalyse
- Werkzeuge zur dynamischen Analyse
- Werkzeuge zum Build-Management
- Werkzeuge zum Konfigurations- und Versionsmanagement
- Werkzeuge zum Codemanagement
- Werkzeuge zum Test
- Werkzeuge zur Dokumentation

Werkzeuge zum Anforderungsmanagement

- Anforderungsmanagement wird über den gesamten Lebenszyklus der Systementwicklung durchgeführt
- Unterstützung Workflow:
  - Anforderungen erheben
  - Dokumentieren
  - abstimmen und kommunizieren
  - verwalten

Werkzeuge zum Anforderungsmanagement Anforderungen und Entscheidungskriterien

- Unterstützung bei der Erfassung und Analyse der Anforderungen
- Möglichkeit zur textuellen und grafischen Darstellung von Anforderungen
- Verwaltung der Anforderungen
- Reduktion von Redundanzen
- Unterstützung der Rückverfolgbarkeit zwischen Anforderungen und Architektur
- Teamfähigkeit
- Versions- und Konfigurationsmanagement
- Parallele Bearbeitung (durch mehrere Benutzer)

Werkzeuge zur Modellierung

Zurich University of Applied Sciences
zh aw School of Engineering

- Modellierungswerkzeuge können fachliche und technische Modelle von Software sowie Anforderungs- und Problemdomänen darstellen.
- Sie helfen dabei, meist grafische Abstraktionen der Realität zu erstellen und zu pflegen.
- Anforderungen und Entscheidungskriterien
  - Unterstützung standardisierter Modellierungsmethoden, beispielsweise UML, SysML, Entity-Relationship-Modelle, BPMN, StateCharts oder andere
  - Unterstützung freier (informeller) Modelle
  - Unterstützung unterschiedlicher Sichten/Modelltypen (Diagrammarten)
  - Unterstützung von statischer und dynamischer Modellierung
  - Unterstützung expliziter Metamodelle
  - Möglichkeiten zur manuellen oder programmatischen Modifikation der Metamodelle
  - Domänenspezifische Sprachen, auch grafischer Art

Werkzeuge zur Generierung

- Generierungswerkzeuge können auf Basis abstrakter Beschreibungen beliebige Artefakte generieren.
- Beispielsweise können sie auf Basis von
  - UML-Klassenmodellen die zugehörigen Klassen- und Methodenrümpfe in unterschiedlichen Programmiersprachen generieren.
  - Datenmodellen die zugehörigen SQL- oder DDL-Statements für bestimmte Datenbanksysteme generieren, Testdaten oder auch Datenzugriffsbausteine.
  - Formalen Grammatiken die zugehörigen Lexer und Parser generieren.
  - XML-Schemata (xsd's) passende Klassen oder generieren,
  - Quellcode zugehörige Dokumentation erzeugen.
- Anforderungen und Entscheidungskriterien
  - Unabhängigkeit von der Zielplattform bzw. des Formats der generierten Artefakte
  - Unabhängigkeit von Metamodellen der Eingabedaten/-artefakte
  - Flexibilität des Transformationsprozesse

Werkzeuge zur statischen Codeanalyse

- Bei der statischen Analyse wird der Quelltext einer Reihe formaler Prüfungen unterzogen, um die Anwendung nach Auffälligkeiten und Fehlern zu durchsuchen.
- Dies kann manuell sowie werkzeuggestützt erfolgen.
- Prüfung der Umsetzung der Architekturvorgaben
- Anforderungen der Werkzeuge
  - automatisierbar, in Build-Prozess integrierbar
  - Reporting: Aufbereitung der Ergebnisse in verschiedenen Formaten (HTML, RSS, etc.) inklusive Visualisierung
  - flexible Analysekriterien und Metriken
  - Unterstützung unterschiedlicher Programmiersprachen

Werkzeuge zur dynamischen Analyse

- Ziele sind unter anderem
  - Geschwindigkeitsmessung
  - Zeitmessung bestimmter Systemteile in Relation zu anderen Systemteilen
  - Messung der Speichernutzung
  - statistische Auswertung (Nutzungsmetriken)
- Anforderungen und Entscheidungskriterien
  - möglichst geringe Auswirkung auf das Laufzeitverhalten, den Speicher- oder CPU-Bedarf
  - verständliche, zielgruppengerechte Darstellung der Ergebnisse
  - Eignung auch für verteilte Systeme

Werkzeuge zum Build-Management

Zurich University of Applied Sciences
School of Engineering

- Aufgaben der Werkzeuge
  - Management der Übersetzungs- und Transformationsaufgaben (Compile,
  - Link, Deploy)
  - Management von Continuous Integration
  - Abhängigkeitsmanagement
  - Ausführung und Reporting automatisierter Tests
  - Prüfung auf Einhaltung struktureller Vorgaben und Programmierkonventionen
- Anforderungen und Entscheidungskriterien
  - Build-Prozess definierbar
  - minimale Auflösung transitiver Abhängigkeiten,
  - Integration mit Werkzeugen zur Versions- und Konfigurationsverwaltung,
  - Codeanalyse, Ausführung automatisierter Tests und deren Reporting
  - Anbindung an Continuous-Integration-Werkzeuge oder -Prozesse

Werkzeuge zum Konfigurations- und Versionsmanagement

- Aufgaben der Werkzeuge
  - Zuordnung und Selektion von Konfigurationselementen zu einer Konfiguration
  - Inventarisierung
  - Rekonstruktion einer Konfiguration
- Anforderungen und Entscheidungskriterien
  - Skalierbarkeit auf große Entwicklungsteams
  - Behandlung beliebiger Varianten (Branches, Versionen)
  - Zuverlässigkeit und Robustheit
  - Integration mit anderen Werkzeugen (z. B. Versionsverwaltung, Issue- und Anforderungsmanagement, Build-Werkzeuge, Codemanagement)

Werkzeuge zum Codemanagement

- Zu dieser Kategorie von Werkzeugen zählen:
  - syntaxbezogene Editoren
  - Refactoring-Werkzeuge: Umformung von Quellcode unter Beibehaltung der funktionalen Eigenschaften
  - Debugger und integrierte Entwicklungsumgebungen
- Anforderungen und Entscheidungskriterien
  - Integration in Entwicklungsumgebung
  - einfache, ausführbare Beschreibung von Tests
  - Reporting von Testergebnissen
  - Sammlung von Testergebnissen über mehrere Testläufe hinweg zur Erkennung von Trends

Werkzeuge zum Test

- Testwerkzeuge
  - Unit-Tests, z. B. xUnit-Derivate
  - Laufzeittests, etwa Last-/Performancetests, Stresstests, Robustheitstests
  - Penetrationstests, Angriffsszenarien
  - Verwaltung von Testfällen, Testdaten, Testergebnissen
- Anforderungen und Entscheidungskriterien
  - Integration in Entwicklungsumgebung
  - einfache, ausführbare Beschreibung von Tests
  - Reporting von Testergebnissen
  - Sammlung von Testergebnissen über mehrere Testläufe hinweg zur Erkennung von Trends

Werkzeuge zur Dokumentation

- Werkzeuge
  - Klassische Textverarbeitung (»Office-Produkte«) diverser Hersteller
  - Markup-basierte Ansätze (DocBook, DITA, SGML, MarkDown, XHTML),
  - Wikis (beispielsweise die vielen Open-Source-Wikis wie TWiki, Mediawiki), oder das kommerzielle Confluence (Firma Atlassian)
  - Viele Modellierungswerkzeuge können Dokumentation aus ihrem Datenbestand/Repository generieren.
- Anforderungen
  - Integration mit Versions- und Konfigurationsverwaltung
  - Konformität der Ergebnisdokumente mit Unternehmens- oder Organisationsstandards, etwa Corporate Layout oder Corporate Design
  - Möglichkeit zur Erzeugung Stakeholder-spezifischer Dokumentation
  - einfache Synchronisierung der Dokumentation mit Releases oder Versionen der Software und Integration mit Bug- oder Issue-Tracking-Systemen

Fragen