

Software-Entwicklungswerkzeuge (SEW)

Prof. Dr. Uwe Aßmann
Technische Universität Dresden
Institut für Software- und Multimediatechnik
<http://st.inf.tu-dresden.de>
WS 11/12-0.1, 10/13/11

SEW, © Prof. Uwe Aßmann

1

Ziele

- ▶ Verstehe, wie Software-Werkzeuge funktionieren
 - Architektur
 - Komponenten
 - Sprachen
 - Technikräume, Modellalgebren und Metasprachen
- ▶ Verstehe, wie man sie zusammensteckt (komponiert)
- ▶ Verstehe Software-Entwicklungsumgebungen als zusammengesteckte Werkzeuge
- ▶ Lerne wichtige Sprachen kennen, die Werkzeuge prägen
 - mit denen man einzelne Werkzeuge bauen kann
- ▶ Unterscheide code- und modellzentrierte Werkzeuge
- ▶ Verstehe Werkzeuge für den Software-Entwicklungsprozess

Prof. U. Aßmann, SEW

2

Grobgliederung der SEW-Vorlesung

Teil I: Grundlagen

- ▶ Einführung zu Werkzeugen, Werkzeugmaschinen und Software-Entwicklungsumgebungen (SEU)
- ▶ Basistechniken in Werkzeugen
- ▶ Sprachfamilien in Werkzeugen

Teil II: Architektur von Werkzeugen

- ▶ Architektur von SEU
- ▶ Funktionen und Anforderungen an die Benutzungsschnittstelle
- ▶ Repository
- ▶ Prozessintegration durch Meta- und Vorgehensmodelle

Teil III: Code-zentrierte Werkzeuge

- ▶ Prototyping-Werkzeuge
- ▶ Parser-Generatoren
- ▶ Werkzeuge zur Programmanalyse
- ▶ Werkzeuge zur Transformation
- ▶ Werkzeuge zur Optimierung
- ▶ Werkzeuge zur Programmüberführung (Codegenerierung)

Teil IV: Werkzeuge im Software-Lebenszyklus

- ▶ Werkzeuge zur Anforderungsanalyse
- ▶ Testwerkzeuge
- ▶ Dokumentationswerkzeuge
- ▶ Werkzeuge für Baumanagement
- ▶ Werkzeuge für Wartung und Reengineering
- ▶ Erfahrungen und weitere Entwicklungen

Teil V: Fortgeschrittene Themen

Gliederung Teil I - Grundlagen

Teil I: Grundlagen

- ▶ 10. Taxonomie von Werkzeugen u. Software-Entwicklungsumgebungen (SEU)
 - 1 Aufbau u. prinzipielle Funktion von Software-Entwicklungswerkzeugen
 - 2 Werkzeuggrundtypen - Klassen von CASE-Tools
 - 3 Werkzeug-Landschaft nach Hesse
 - 4 Einführung in Effektkategorien (Blutgruppen)
 - 5 Graph-Logik-Isomorphismus
- ▶ 11. Basistechniken: Metamodellierung und Technologieräume
- ▶ 12. Sprachfamilien in Werkzeugen
 - 1 Begriffsbestimmung
 - 2 DDL: RM, ERD als isomorphes RM, EBNF, CD
 - 3 DQL: Xcerpt, Semmler .QL
 - 4 DCL: OCL, Spider, URML
 - 5 DML: Term- und Graphersetzung
 - 6. DRL: Data Restructuring Languages
 - 7 BSL: DFD in SA, Mashups
- 13 Interpretation and Semantics
- 14 Abstract interpretation of BSL
- 15 Behavioral model checking of automata-based BSL
 - 1 Standard model checking
 - 2 Software model checking
 - 3 Real-time model checking
- Optional: Wiederholungen von ST-1 und ST-2
 - 19.1 Entity-Relationship-Diagramme
 - 19.2 Data Dictionary
 - 19.3 Datenflussdiagramme
 - 19.4 Minispezifikationen (Pseudocode)
 - 19.5 Zustandsgraphen
 - 19.6 Entscheidungstabellen
 - 19.7 Objektorientierte Diagramme

Prof. U. Aßmann, SEW

3

Prof. U. Aßmann, SEW

4

Gliederung Teil II - Architektur von Werkzeugen

Teil II: Architektur von Werkzeugen

- ▶ 20. Architektur von Software-Entwicklungsumgebungen (SEU)
 - 1 Konzepte der Werkzeugintegration
 - 2 Architektur von SEU
 - 3 Das ECMA Referenzmodell
- ▶ 21. Repository
 - 1 Ziele und Aufgaben des Repository
 - 2 Architektur von Repositories
 - 3 Syntax-Mapping und CASE-Austauschformate
 - 4 Frameworks zur Werkzeugintegration (PCTE)
- ▶ 22. Komposition von repository-basierten Werkzeugen
 - 1 Effektkategorien und Blutgruppen
 - 2 Werkzeuge als Objekte
 - 3 Einsatz des Graph-Logik-Isomorphismus
 - 4 Essentielle Zerlegung (EAI)
 - 5 Repository-Integration mit Rollen
- ▶ 23. Komposition von Werkzeugen
 - Datenfluss-basierte Integration
- ▶ **6b Prozessintegration**
 - **Prozessmodelle und ihre Werkzeuge**
 - **SuReal-Umgebung**
 - **Metamodellierung von Prozessen**
- 24 Syntaxmapping mit EMFText
 - Für Syntaxmapping, Technologieraumbrücken und DSL-Entwicklung
- ▶ 25. MetaCase-Werkzeuge
 - Metamodellierung
 - MetaCASE-Umgebungen
 - MOFLON
 - FlowR from Achievo
- ▶ 27 Werkzeuge zur Konstruktion domänenspezifischer Sprachen
 - (EMFText)
 - Reuseware zur Komposition von DSL-Fragmenten
- ▶ **(8 Technologieraumbrücken)**
 - **Ontologyware---Modelware**
 - **OWLText**

Prof. U. Aßmann, SEW

5

Gliederung Teil III

Teil III: Code-zentrierte Werkzeuge

- ▶ 31. Prototyping-Werkzeuge
 - 31.1 Arten und Werkzeugklassen für das Prototyping
 - **31.2 Skriptsprache: Pike, Groovy, adaptable typing**
 - **31.3 Refactoring-Werkzeuge**
- ▶ 32. *Technologieraum Grammarware*
 - Moderne Parsergeneratoren
 - Optimierergeneratoren
 - Backend-Generatoren
 - 32.1 ANTLR
- ▶ 33 Werkzeuge zur Programmanalyse
 - 33.1 Visualisierung
 - VCG, AiSee, LogLin und CroCoCosmos
 - 33.2 WCETA mit AiT
 - Interprocedural Analysis with PAG
- ▶ **34 Werkzeuge zur Termersetzung**
 - **Stratego**
- ▶ 35 Werkzeuge zur Modell-Transformation
 - 34.1 Graphersetzung
 - 34.3 Fujaba
 - 34.2 Triple-Graph-Grammars (Tornado)
 - Control-Flow Analysis with EARS
 - Interprocedural Analysis with IDFS
- ▶ 38 Werkzeuge für die MDA
- ▶ 37. Werkzeuge zur Programmüberführung
 - 36.1 Codeumwandlung
 - 36.2 Codeergänzung und -selektion
 - 36.3 Codemodifikation und -rückführung

Prof. U. Aßmann, SEW

6

Gliederung Teil IV

Teil IV: Werkzeuge im Software-Lebenszyklus

- ▶ 40. Werkzeuge zur Anforderungsanalyse
 - 1 Techniken des Requirement Engineering
 - 2 Lastenheft und Pflichtenheft
 - 3 Werkzeuggestützte Anforderungsanalyse
 - 4 Prüf-, Dokumentations- und Verwaltungswerkzeuge
 - **5 Geschäftsregelanalyse (RuleXpress)**
- 41. Testwerkzeuge
 - 1 Aufgaben und Arten von Testwerkzeugen
 - 2 Automatisierungsgrad von Testwerkzeugen
 - 3 Werkzeuge ausgewählter Testumgebungen
 - 4 Debugger
 - 5 Kommerzielles Beispiel: Werkzeugsuite TestBench
- ▶ 43. Dokumentationswerkzeuge
 - 43.1 Aufgaben der Dokumentationswerkzeuge
 - 43.2 Beispiele einiger typischer CASE-Dokumentationswerkzeuge
- ▶ 44 Werkzeuge zum Konfigurationsmanagement
- ▶ 45. Baumanagement
 - 45.1 Werkzeuge
 - 45.2 Das Baumanagementsystem von GNU
 - 45.3 Fehlermanagement
- ▶ 46. Werkzeuge für Wartung und Reengineering
 - 46.1 Aufgaben von Wartung und Reengineering
 - 46.2 Vorgehen des Reengineering
 - 46.3 Werkzeuge für das Reengineering
- ▶ Teil V: Fortgeschrittene Themen
 - 50. Hybride MDS
 - 51. Modellmanagement

Prof. U. Aßmann, SEW

7

Literatur zu SEW

- [1 B93] Balzert, H. (Hrg.) u.a.: CASE - Systeme und Werkzeuge; BI-Wissenschaftsverlag Mannheim, 5. vollst. überarb. Auflage 1993
- [2 BAL, BOO] Balzert, H. : Lehrbuch der Software-Technik (Bd. I): Software-Entwicklung; Spektrum Akademischer Verlag (3. Auflage) Heidelberg 2009
- [3 BAL-II B98] Balzert, H. : Lehrbuch der Software-Technik (Bd. II): Software-Management. Spektrum Akademischer Verlag Heidelberg 1998
- [4 HMF] Hesse, W., Merbeth, G., Fröhlich, R.: Software-Entwicklung / Vorgehensmodelle. Projektführung, Produktverwaltung; Oldenbourg Verlag München 1992
- [5 SN92] Schönthaler, F., Nemeth, T.: Software-Entwicklungswerkzeuge: Methodische Grundlagen; B.G. Teubner Verlag Stuttgart 1992
- [6 ES89] Engels, G., Schäfer, W.: Programmentwicklungsumgebungen - Konzepte und Realisierung; B.G. Teubner Verlag Stuttgart 1989
- [7 ZK04] Zeller, A., Krinke, J.: Open-Source-Programmierwerkzeuge (2. Auflage) dpunkt.verlag Heidelberg 2004
- [8 F91] Fisher, A.S.: CASE - Using Software Development Tools (Second Edition); John Wiley & Sons 1991
- [9 B92] Bauknecht, K. (Hrg.): Informatik-Anwendungsentwicklung - Praxiserfahrungen mit CASE; B.G. Teubner Verlag Stuttgart 1992
- [Züll] Züllighoven, Heinz. Object-Oriented Construction Handbook; dpunkt.verlag 2005
- [Raasch] Raasch. Systementwicklung mit strukturierten Methoden. Hanser. 1993

Prof. U. Aßmann, SEW

8



► http://en.wikipedia.org/wiki/Computer-aided_software_engineering

