

Software Reuse

1. Einführung

Peter Sturm
Universität Trier

(c) 2004 AG SYSOFT - UNIVERSITY OF TRIER

Organisatorisches

- Hauptstudiumsvorlesung
 - Ideale Ergänzung im Anschluß an Systemsoftware I und II
- 2+2 SWS
 - Vorlesung Mittwochs, 14.00 bis 15.30 Uhr?
 - 3-4 Übungsblätter (Programmieraufgaben)
- Leistungsnachweis
 - “Erfolgreiche” Bearbeitung der Übungsblätter
- Folienkopien
 - Im WWW
 - Vorhandene Kopien sind je nach Thema noch aktuell



(c) 2004 AG Sysoft - University of Trier

Inhalte

- Themenkomplex etwas erweitert
- Einführung
 - Allgemein "Software Reuse"
- Komponentensysteme
 - Konzepte
 - Beispiele: Corba, Java Beans und EJB, .NET, (COM+)?
 - Web Services
- Frameworks
 - Konzepte
 - ggf. Beispiele (MFC-Problematik)
- Reuse von Entwurfsprozessen und -wissen



© 2004 AG Sysoft - University of Trier

Softwareentwicklung

- Wer hat schon eigene Software entwickelt?
 - Umfang?
 - Projektgröße in Anzahl Entwickler?
- Wer hat schon fremde Software (wieder)verwendet?
 - "Not invented here syndrom"?
 - Umfang?
- Wie wurde Software (wieder)verwendet?
 - Code Scavenging?
 - Bibliothek?
 - Komponenten?
 - Frameworks?
 - Ganze Applikationen verklebt?



© 2004 AG Sysoft - University of Trier

Reuse

- Ausgangspunkt
 - Entwicklungsprojekte sind keine einmaligen Ereignisse
- ... von Code
 - Frameworks
 - Komponenten
 - Produktfamilien
- ... von Erfahrungen
 - Gemeinsame Sprache: UML
 - Entwurfsmuster guter Softwarestrukturen
 - Erweiterte Softwareprozesse (z.B. CMM)



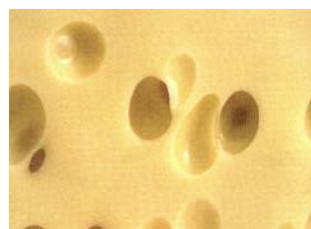
© 2004 AG Sysoft - University of Trier

Frameworks und Komponenten



Komponenten

- Hohe Qualität der Einzelkomponente erreichbar
- Mehraufwand amortisiert sich durch Reuse
- Hoher Verbreitungsgrad (COM, EJB, .NET, CORBA)



Framework

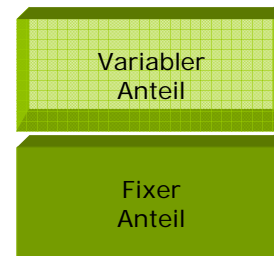
- Von Experten geschrieben \Rightarrow Hohe Qualität
- Je spezifischer die Anwendungsdomäne, desto besser
- Mehraufwand amortisiert sich durch Reuse



© 2004 AG Sysoft - University of Trier

Produktfamilien

- Abfolge von ähnlichen Produktentwicklungen
- Isolierung der variablen Produktanteile
- Einzelprojekt-übergreifender Softwareprozeß
- Ergänzend zu Frameworks und Komponenten

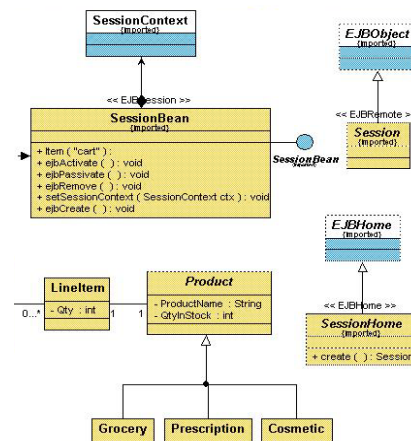


© 2004 AG Sysoft - University of Trier

Gemeinsame Sprache

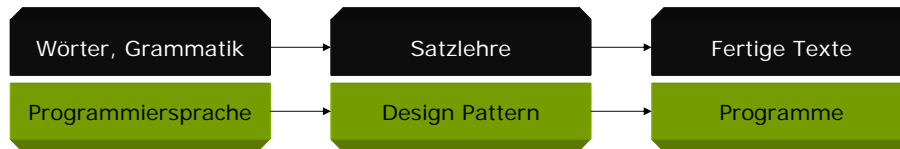


- Graphische Notation mit wohldefinierter Semantik
- Vielfältige Diagramme
 - Statisch Strukturbezogen
 - Dynamisch Ablaufbezogen
 - Hoher Einarbeitungs-aufwand
- Phasenspezifische Diagramme



© 2004 AG Sysoft - University of Trier

Design Pattern



- Direkte Umsetzung in Form einer Code-Bibliothek nicht möglich
 - Struktureigenschaften der Anwendung
 - Verteilt sich über mehrere Codestellen
- Einheitliche Bezeichnung und Beschreibung
- Standardwerke, z.B. Gamma et al.



© 2004 AG Syssoft - University of Trier

Process Improvement

- Reflektion durch alle Beteiligten
- Verbesserungen über mehrere Projekte hinweg
 - Anpassung bzw. Änderung der Prozesse
 - Optimieren und Ergänzen von Metriken
 - “Erfahrungsdatenbanken” wünschenswert
- Beispiele
 - Capability Maturity Model (CMM), Spice, Six Sigma, ...
- Es gibt auch kritische Stimmen



© 2004 AG Syssoft - University of Trier

Werkzeugunterstützung

- Tools sind unersetzlich geworden
 - Dramatische Steigerung der Effizienz bei hoher Qualität
 - Moderne Werkzeuge unterstützen höhere Ebenen (z.B. Design Pattern)
 - Refactoring
 - Hoher Einarbeitungsaufwand notwendig
- Wiederverwendung von Erfahrung
- “A fool with a tool is still a fool”



© 2004 AG Sysoft - University of Trier