

Software Engineering

SS 2005

Prof. Dr. Barbara Paech, Jürgen Rückert



Institut für Informatik
Im Neuenheimer Feld 326
69120 Heidelberg
<http://www-swe.informatik.uni-heidelberg.de>
paech@informatik.uni-heidelberg.de

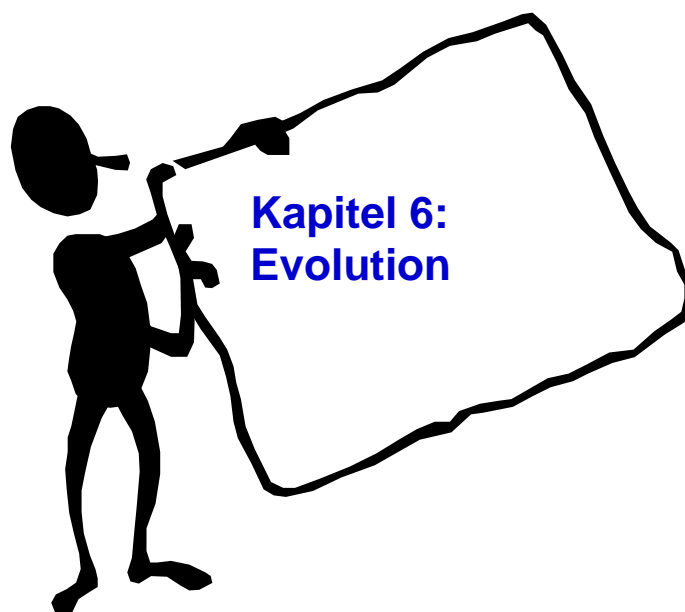


RUPRECHT-KARLS-UNIVERSITÄT HEIDELBERG



6. Evolution

- 6.1. Überblick
- 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung



1.3.1. Aufgabenbereiche des Engineering



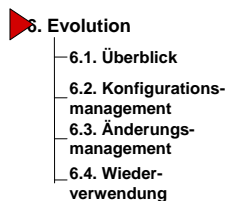
Folie 3

Barbara Paech

Vorlesung - Software Engineering - SS 2005

6. Evolution

- ◆ 6.1. Überblick
- ◆ 6.2. Konfigurationsmanagement
- ◆ 6.3. Änderungsmanagement
- ◆ 6.4. Wiederverwendung



Folie 4

Barbara Paech

Vorlesung - Software Engineering - SS 2005

- ◆ Verwendung von Softwareeinheiten in **neuen Kontexten**
 - Verwendung möglichst ohne Veränderung => **Wiederverwendung**
 - Verwendung mit Änderungen => **Weiterentwicklung**
 - Erfordert Verständnis der existierenden Software => **Re-Engineering**
 - Erfordert Zusammensetzung existierender und neuer Komponenten => **Konfigurationsmanagement**
 - Erfordert systematischen Prozess des Einbringen und Umsetzen von Änderungswünschen => **Änderungsmanagement**

6. Evolution

- ▶ 6.1. Überblick
- 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung

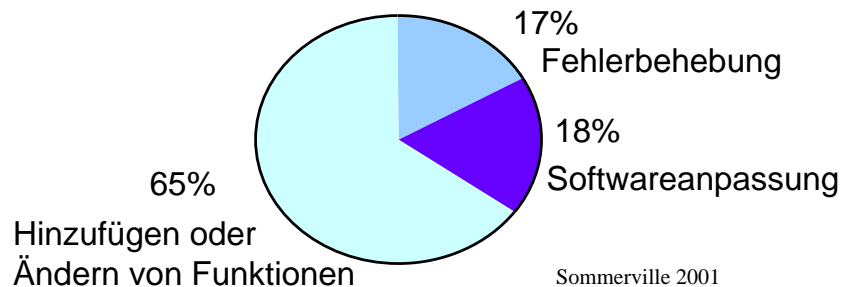
Software-Systeme müssen geändert werden

- Während der Entwicklung (z.B. durch instabile Anforderungen)
- Nach Auslieferung (z.B. durch neue Benutzerwünsche)

**Verhältnis
Entwicklung <-> Wartung**
20-40 % 60-80 %

Jalote 1997

Verschiedene Arten von Änderungen



6. Evolution

- ▶ 6.1. Überblick
- 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung

6.1. Umgang mit Änderungen

- ◆ **Traditionelles Vorgehen:**
 - Versuche Änderungen vorherzusehen und **Software** so auszulegen, dass Änderungen leicht durchzuführen sind (z.B. sehr generisch, starke Modularisierung)
- ◆ **Agiles Vorgehen:**
 - Änderungen lassen sich schwer vorhersehen, Aufwand für „Vorhersage“ ist zu hoch. Deshalb Prozess so auslegen, dass Änderungen leicht durchzuführen sind (z.B. Regressionstest, Re-Factoring)
- ◆ **Synthese: Risikomanagement**
 - Schätze **effizient** Kosten und Gefahr **schwerwiegender Änderungen** ab; bereite **Software** auf diese Änderungen vor; ermögliche agilen **Änderungsprozess** für kleinere Änderungen

6. Evolution

- ▶ 6.1. Überblick
- 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung

6.2. Konfigurationsmanagement

- ◆ 6.2.1. Begriffe
- ◆ 6.2.2. Aufgaben des Konfigurationsmanagements
- ◆ 6.2.3. Verwaltung der Komponenten
- ◆ 6.2.4. Zusammensetzung
- ◆ 6.2.5. Verteiltes Arbeiten

6. Evolution

- ▶ 6.1. Überblick
- 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung

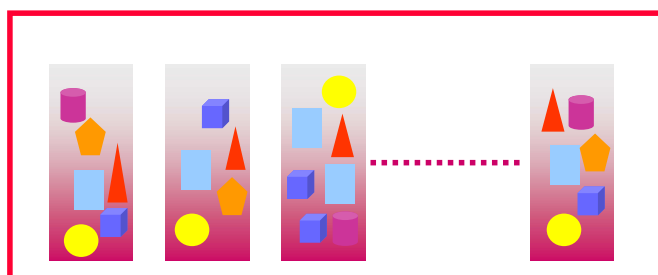
6.2.1. Begriffe des Konfigurationsmanagements

- ◆ Entwicklung einer Komponente über die Zeit => **Version**
- ◆ Zusammensetzung aus verschiedenen **Komponenten** => **Konfiguration**
- ◆ Auslieferung eines Produktes (einer Konfiguration) in verschiedenen Ausbaustufen an einen Kunden => **Release**
- ◆ Auslieferung **verschiedener Produkte** (mit verschiedenen Features) an **verschiedene Kunden** => **Varianten, Produktlinien**

6. Evolution

- 6.1. Überblick
- ▶ 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung

6.2.1. Begriffsübersicht

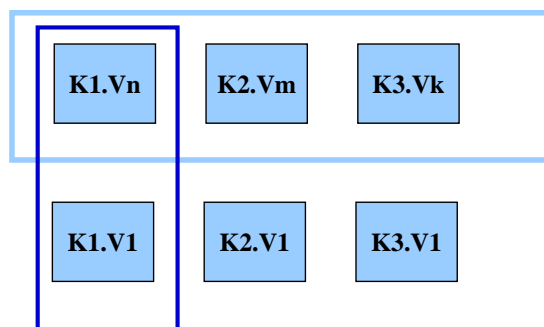


**Produkt-
Linie,
Releases**



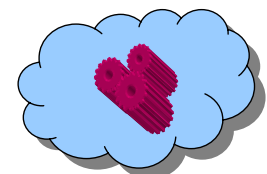
6. Evolution

- 6.1. Überblick
- ▶ 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung



**Konfigura-
tion**

Versionen



6.2.1. Version und Konfiguration

- ◆ Version ist die Ausprägung einer Komponente zu einem bestimmten Zeitpunkt
- ◆ Konfiguration ist eine konsistente Zusammenstellung von mehreren Softwarekomponenten
- ◆ Eine Version kann zu mehreren Konfigurationen gehören
- ◆ Zu einer Konfiguration kann von einer Komponente (höchstens) eine Version gehören
- ◆ **Baseline:** speziell ausgezeichnete Konfiguration, als Ausgangspunkt für Weiterentwicklung

6. Evolution

- 6.1. Überblick
- ▶ 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung

6.2.1. Release

- ◆ **Release:** an Kunden ausgelieferte Konfiguration
- ◆ Ein **Release** enthält typischerweise
 - Konfigurationsdateien (Auswahl der Konfiguration)
 - Daten, die für einen erfolgreichen Betrieb nötig sind
 - Installationsprogramm
 - Systemhandbuch
 - Produktwerbung
- ◆ **Neues Release** notwendig, wenn
 - Ernsthafte Fehler behoben wurde (bei kleinen Fehlern Patches)
 - Wichtige neue Funktionalität implementiert wurde
 - Um Konkurrenzprodukt zu überholen
 - Um Marketingplanung einzuhalten
 - Um Änderungswünsche von Kunden aufzunehmen

6. Evolution

- 6.1. Überblick
- ▶ 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung

6.2.2. Aufgaben der Konfigurationsverwaltung

- ◆ Verwaltung von Versionen und Konfigurationen (erleichtern von Ablage und Auffinden)
- ◆ Zusammensetzung der Komponenten
- ◆ Verteiltes Arbeiten

6. Evolution

- 6.1. Überblick
- 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung

6.2.3. Verwaltung

- ◆ Elemente **und ihre Änderungen** zugreifbar machen
- ◆ Statusverwaltung

6. Evolution

- 6.1. Überblick
- 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung

6.2.3. Zugriff ermöglichen

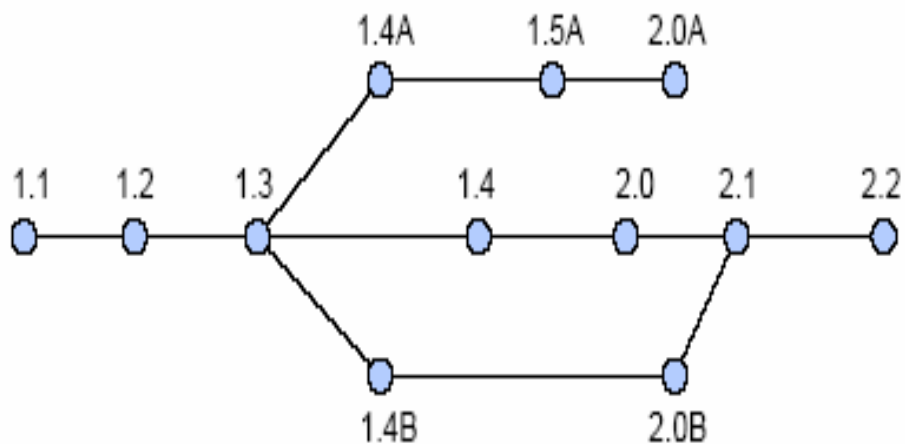
- ◆ **Eindeutige Bezeichnung**
 - Linear: Doc1, Doc2, Doc3
 - Hierarchisch: LSB.BDS.ENT.Doc1 (Projekt LSB, Komponente BDS, Phase Entwurf)
- ◆ **Dokumentation der Änderung**
 - Aufzeichnung der Änderungsgeschichte
 - um Speicher zu sparen: Speicherung des Deltas zur wiederholten Herstellung
- ◆ **Registrierung aller Komponenten**
 - Liste
 - Datenbank

6. Evolution

- 6.1. Überblick
- 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung

6.2.3. Versionsfortschritt

Verzweigungen möglich!



6. Evolution

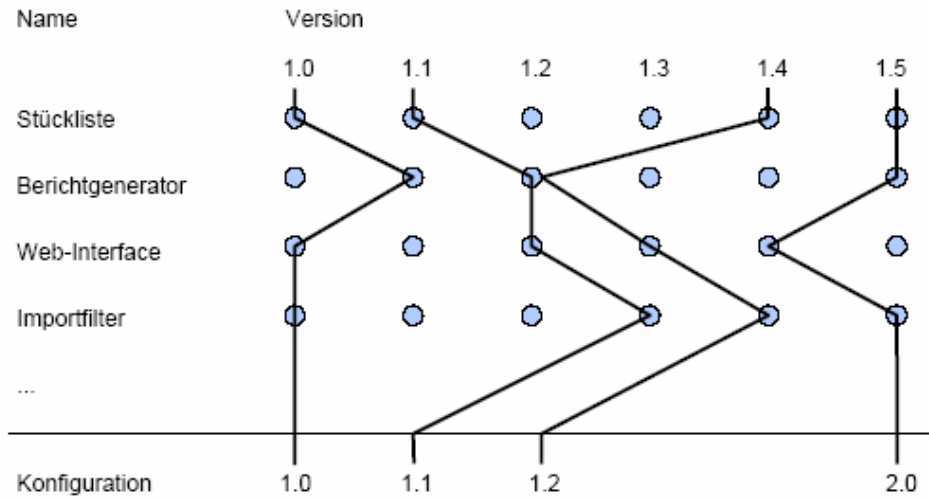
- 6.1. Überblick
- 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung

6.3. Konfigurationsfortschritt

Auch hier Verzweigungen möglich!

6. Evolution

- 6.1. Überblick
- 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung



F.Houdek: Vorlesung Projektmanagement WS2002/2003

Folie 17

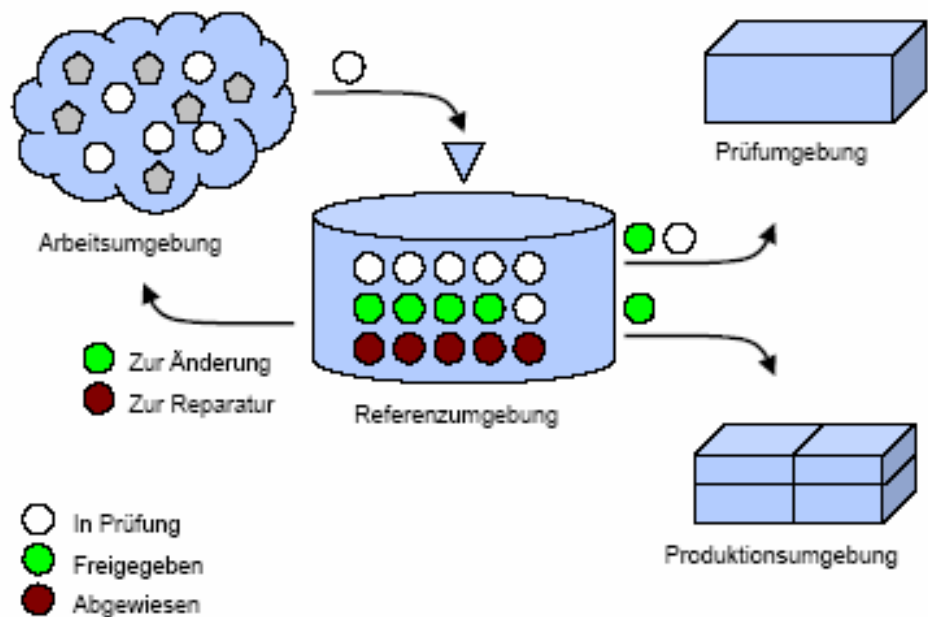
Barbara Paech

Vorlesung - Software Engineering - SS 2005

6.2.3. Statusverwaltung

6. Evolution

- 6.1. Überblick
- 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung



F.Houdek: Vorlesung Projektmanagement WS2002/2003

Folie 18

Barbara Paech

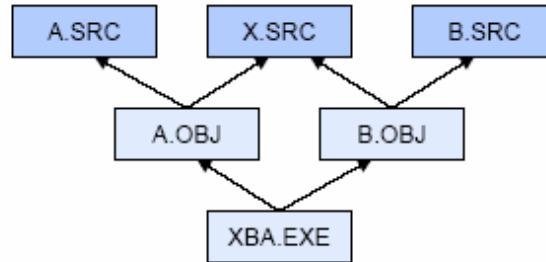
Vorlesung - Software Engineering - SS 2005

6.2.4. Erstellung einer Version

- ◆ **Originale** (z.B. Anforderungsdokumente, Quellcode, Projektplan)
- ◆ **Evtl. Ableitungen** (z.B. Object-Files, Executable, Cross-Reference-List)

6. Evolution

- 6.1. Überblick
- 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung



F.Houdek: Vorlesung Projektmanagement WS2002/2003

- ◆ **Verfahren zur Ableitung** (Makefiles, Compiler, etc)

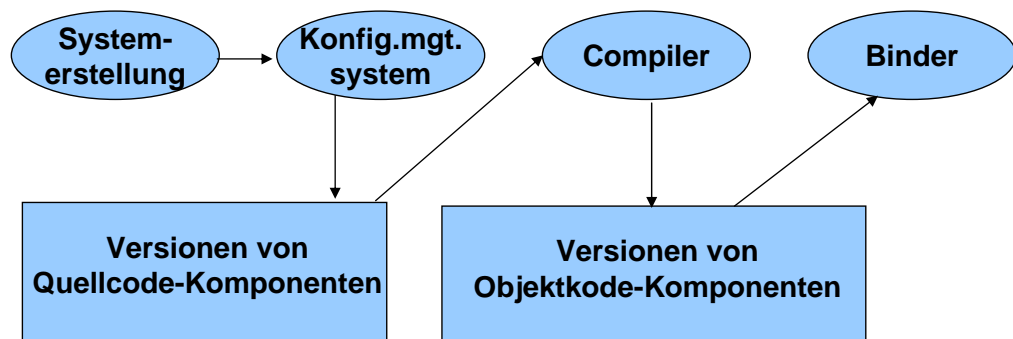
Folie 19

6.2.4. Systemerstellung

- ◆ **Durch Erstellungsskript**

6. Evolution

- 6.1. Überblick
- 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung



Folie 20

6.2.5. Verteiltes Arbeiten

- ◆ Definition und Überprüfung von **Zugriffsrechten**
- ◆ Definition und Überprüfung **gleichzeitigen Zugriffs**
 - **Elementbasiert**: jeder Entwickler kann jederzeit auf bestimmte Menge von Komponenten zugreifen
 - **Taskbasiert**: Entwickler kann nur zugreifen, wenn er spezifische Aufgabe durchführt

6. Evolution

- 6.1. Überblick
- 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung

6.3. Ziele des Änderungsmanagements



- ◆ **Konflikte** zwischen geforderten Änderungen möglichst früh entdecken (stellt konfliktfreie Implementierung sicher)
- ◆ Geforderte Änderungen **priorisieren** (erlaubt Release-Planung und gewährleistet Kundenzufriedenheit)
- Änderungen **konsistent** durchführen
- Durchführung von Änderungen **messen** (stellt bessere Planung sicher)
- Aus Änderungsprojekten **lernen** (gewährleistet bessere Planung)

6. Evolution

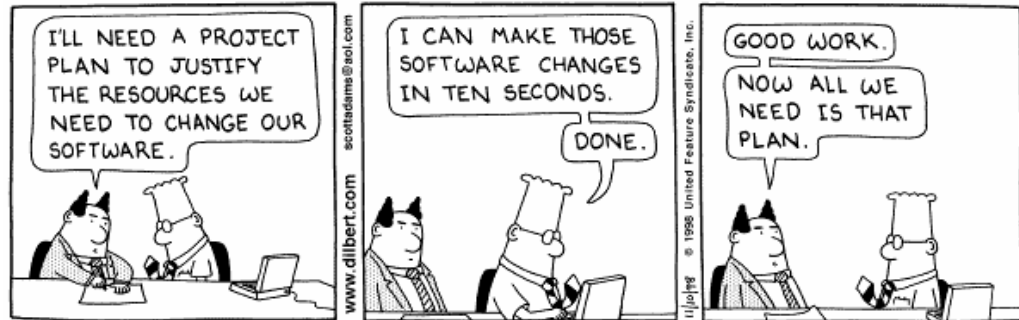
- 6.1. Überblick
- 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung

6.3. Änderungsmanagement

- 6.3.1. Vorbereitung für Änderungen: **Verfolgbarkeit**
- 6.3.2. Identifikation von Änderungen
- 6.3.3. Planung von Änderungen
- 6.3.4. Durchführung von Änderungen

6. Evolution

- 6.1. Überblick
- 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung



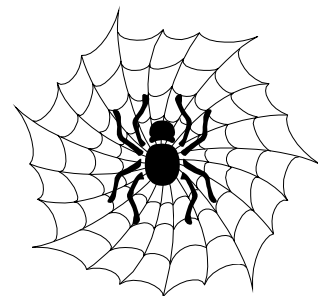
Folie 23

Barbara Paech

Vorlesung - Software Engineering - SS 2005

6.3.1. Was ist Verfolgbarkeit ?

Dokumentation von Beziehungen Zwischen Elementen



6. Evolution

- 6.1. Überblick
- 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung

- Einer Stufe (z.B. zwischen Anforderungen) (**horizontal**)
- Verschiedener Stufen (z.B. Anforderungen zu Komponenten) (**vertikal**)
- Verschiedener Versionen (**evolutionär**)

Folie 24

Barbara Paech

Vorlesung - Software Engineering - SS 2005

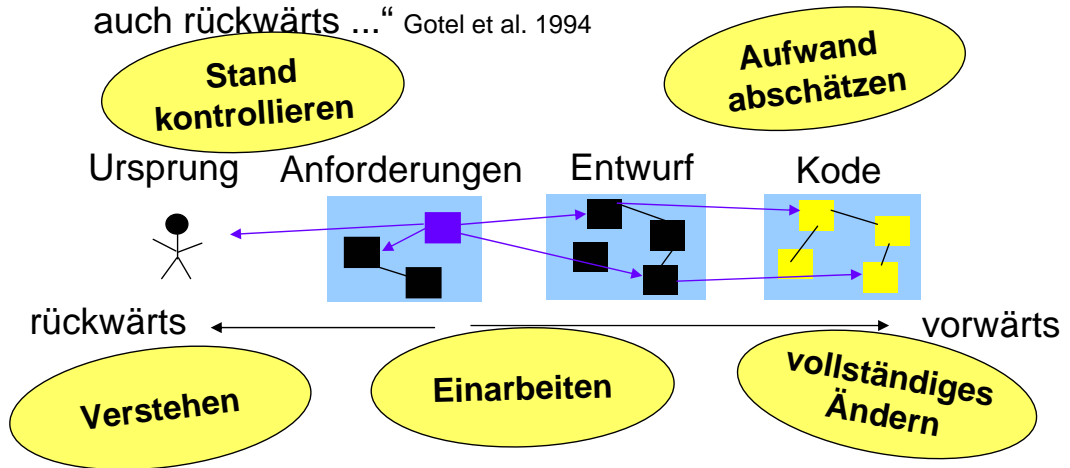
6.3.1. Warum Verfolgbarkeit?

Verfolgbarkeit

„Fähigkeit, das Leben eines Elementes zu beschreiben und verfolgen zu können und zwar sowohl vorwärts als auch rückwärts ...“ Gotel et al. 1994

6. Evolution

- 6.1. Überblick
- 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung



Folie 25

Barbara Paech

Vorlesung - Software Engineering - SS 2005

6.3.1. Wie repräsentiere ich Beziehungen? - Namen/Struktur

6. Evolution

- 6.1. Überblick
- 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung

Das TSG übernimmt die Funktion **Sitzeinstellung** und *Benutz* im Fahrzeug:

- **Sitzeinstellung**

/* ... */

3 Eingänge der Stecker

Charakteristiken

- *Sensor-Eingänge Türgriff*
Anschlüsse: S1.12 (T_OFFEN)
Charakteristik: Nicht entprellter Mikroschalter gegen
Abbildung 8)

/* ... */

Beschreibung	Bezeichnung	Anschluss	TSG _F	TSG
Fahrer bzw. Beifahrer-Tür offen	T_OFFEN	S1.12	Belegt	Bele

7 Funktionen

7.1 Sitzeinstellung

7.1.1 Eingänge

- Zustand der Vordertür
T_OFFEN

Intern gibt es eine Verbindung zum Benutzermanagement.

Folie 26

Barbara Paech

Vorlesung - Software Engineering - SS 2005

6.3.1. Wie repräsentiere ich Beziehungen– Querverweis

6. Evolution

- 6.1. Überblick
- 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung

Temp-BA	Szenariol	3.	Der Benutzer setzt die ZeitspanneAufheizenNormal auf 9 Minuten.	Temp-BA-F3
		4.	Das System ignoriert die neu eingestellte Zeit, da sie nur 4 Minuten über der ZeitspanneEintrittNormal liegt und liefert eine Fehlermeldung.	Temp-BA-F11
		5.	Der Benutzer setzt die ZeitspanneAufheizenNormal auf 10 Minuten.	Temp-BA-F3
		6.	Der Benutzer setzt die AbwesenheitsTempNormal auf 16°C	Temp-BA-F2
		7.	Der Benutzer setzt die ZeitspanneVerlassenNormal auf 10 Minuten.	Temp-BA-F4

Temp-BA-F15	Temp-BA-F15	Wenn die Zimmertemperatur bei Anwesenheit einer Person die Temperatur aus																					
<table border="1"> <thead> <tr> <th>ID</th> <th>Car user requirements parsed in</th> <th></th> </tr> </thead> <tbody> <tr> <td>UR14</td> <td>Five average size adults shall be able to travel in comfort for a period of 3 hour.</td> <td></td> </tr> <tr> <td>UR15</td> <td>Two average size adults and 3 average size children shall be able to travel in comfort for a period of 3 hours.</td> <td>3 out-links</td> </tr> <tr> <td>UR16</td> <td>Users shall have easy entry and exit.</td> <td></td> </tr> <tr> <td>UR18</td> <td>3.1.1.2 Amount of luggage</td> <td></td> </tr> <tr> <td>UR19</td> <td>Users shall be able to carry 200 Kilograms of luggage.</td> <td></td> </tr> <tr> <td>UR20</td> <td>Users shall be able to carry a single piece of luggage of size 1.2 meters by 0.4 meters by 0.3 meters totally within the car.</td> <td></td> </tr> </tbody> </table>			ID	Car user requirements parsed in		UR14	Five average size adults shall be able to travel in comfort for a period of 3 hour.		UR15	Two average size adults and 3 average size children shall be able to travel in comfort for a period of 3 hours.	3 out-links	UR16	Users shall have easy entry and exit.		UR18	3.1.1.2 Amount of luggage		UR19	Users shall be able to carry 200 Kilograms of luggage.		UR20	Users shall be able to carry a single piece of luggage of size 1.2 meters by 0.4 meters by 0.3 meters totally within the car.	
ID	Car user requirements parsed in																						
UR14	Five average size adults shall be able to travel in comfort for a period of 3 hour.																						
UR15	Two average size adults and 3 average size children shall be able to travel in comfort for a period of 3 hours.	3 out-links																					
UR16	Users shall have easy entry and exit.																						
UR18	3.1.1.2 Amount of luggage																						
UR19	Users shall be able to carry 200 Kilograms of luggage.																						
UR20	Users shall be able to carry a single piece of luggage of size 1.2 meters by 0.4 meters by 0.3 meters totally within the car.																						

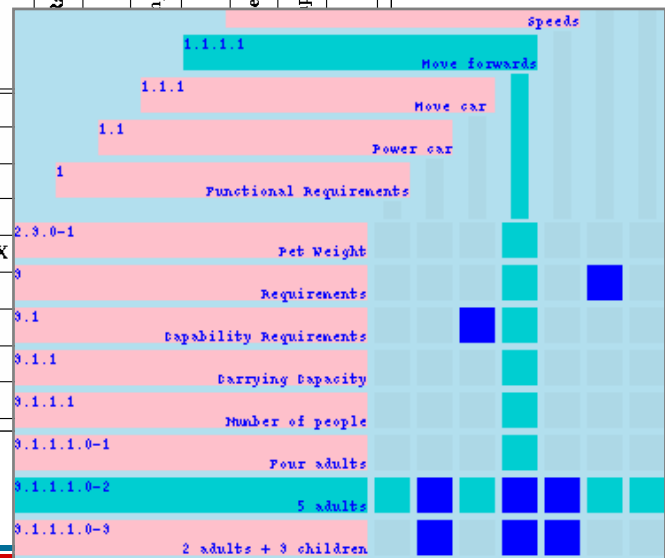
Folie 27

6.3.1. Wie repräsentiere ich Beziehungen? - Matrizen

6. Evolution

- 6.1. Überblick
- 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung

	CentralHeatingSite	Contact	House	Motor	PRTempController	PresenceController	Radiator	RainChecker	Room	TemperatureController	Section	Temperatureregulator	TemperatureSensor	Window
Temp-BA-F1														
Temp-BA-F10		X			X	X								
Temp-BA-F11					X	X								
Temp-BA-F12					X	X								
Temp-BA-F13	X	X					X							
Temp-BA-F14		X	X	X										
Temp-BA-F15		X		X	X	X								
Temp-BA-F16														
Temp-BA-F17			X											



6.3.1. Wie analysiere ich Beziehungen?

6. Evolution

- 6.1. Überblick
- 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung

ID	Car user requirements parsed in	Out-links at depth 1	Out-links at depth 2	Out-links
UR15	Two average size adults and 3 average size children shall be able to travel in comfort for a period of 3 hours.	System requirements SR1001 Power car	Architectural design AD5002 Shell Car user reqts UR133 Architectural design AD5004 Front Doors Car user reqts UR3 Nationalities	Car scen OS2 Car user UR20

Folie 29

Barbara Paech

Vorlesung - Software Engineering - SS 2005

6.3.2. Änderungswünsche identifizieren



6. Evolution

- 6.1. Überblick
- 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung

Wer: Wartungsingenieur

Wie: Befragung unterschiedlicher Benutzergruppen (z.B. System-Administratoren, Entscheidungsträger, Marketing). Hierzu Organisation z.B. von System-Reviews oder Benutzer-Workshops (siehe Anforderungserhebung)

Überprüfung: Z.B. Bewertung der Kundenzufriedenheit

Ergebnis: Beschriebene Änderungswünsche

Folie 30

Barbara Paech

Vorlesung - Software Engineering - SS 2005

6.3.2. Beschriebene Änderungswünsche

System: _____	Change request: _____
Objective:	
Originator:	
Change Priority Routine <input type="checkbox"/> Urgent <input type="checkbox"/> Emergency <input type="checkbox"/>	
Change discovering activity:	
Description of change:	

6. Evolution

- 6.1. Überblick
- 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung

Change request DR: Integration von Sicherheitsaspekten	
Objective	Die in dieser Änderungsanfrage geforderten Änderungen dienen der perfektionierenden Wartung, d. h. das existierende Softwaresystem wird um zusätzliche Funktionalität erweitert.
Originator	Der Kunde verlangt eine Erweiterung des existierenden Systems. Das installierte Softwaresystem genügt nicht mehr seinen gestiegenen Anforderungen.
Current sys. behavior	Das existierende Softwaresystem übernimmt die Halten und Erhöhen der Temperatur im Operationsmodus "heating mode" mit Hilfe der Heizkörper. Das System verwendet die Temperatur- und Lichtregelung im Raum die Jalousien zum Heizen und Kühlen / Abdunkeln und Aufhellen. Dies geschieht unabhängig von einer möglichen Beschädigung der Jalousien durch Sturm. Das Vermeiden einer Beschädigung der Jalousien bei Sturm obliegt dem Benutzer.
	Die existierende Temperaturregelung und die angestrebte Lichtregelung für einzelne Räume wird hinsichtlich der Berücksichtigung von Sicherheitsaspekten verfeinert. Sicherheit-BA-F1: Bei Sturm werden herabgelassene Jalousien in einem Raum nach oben gefahren. Sicherheit-BA-F2: Bei einem Control Panel leuchtet geöffnet ist oder bei Sturm Benutzer geschlossen wurden (die Lampe wird ausgegeben). Die Temperatur: Sicherheit-BA-F3: Eine...

Needed change	Existierende Regelungsaufgaben werden beschränkt.
Priority	Hoch. Der Kunde erwartet die Umsetzung seiner Anforderungen bis Juli 2002. Die Änderungen werden mit dem nächsten Release des Software-Systems (Mitte Juli 2002) herausgegeben.
Status information	In Bearbeitung. Die Problembeschreibung und die tabellarische B-Anforderungen sind zu erweitern. Eine Auswirkungsanalyse zur Berechnung eines Kostenvorschlags wurde noch nicht durchgeführt.

Folie 31

6.3.2. Typische Inhalte eines Änderungswunsches

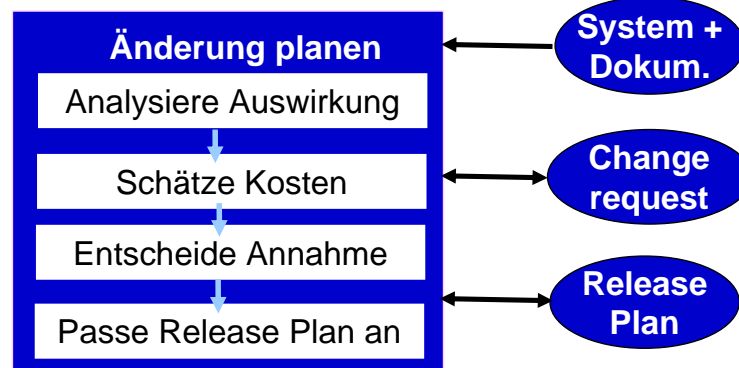
- ◆ Originator (Name, Adresse, Zeitpunkt)
- ◆ Betroffene Komponenten
- ◆ Falls Änderungswunsch aufgrund eines Problem: Reproduzierbarkeit
- ◆ Problembeschreibung
- ◆ Zu treffende Maßnahme
- ◆ Klassifizierung nach Schwere des Problems
- ◆ Status des Änderungswunsches
- ◆ Klassifikation (meist intern, zur Strukturierung großer Problemmeldung vom Kunden)

6. Evolution

- 6.1. Überblick
- 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung

Folie 32

6.3.3. Änderungen planen



6. Evolution

- 6.1. Überblick
- 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung

Wer: Projektplaner

- Wie:**
- Analyse der Auswirkungen von Änderungen basierend auf dokumentierten Beziehungen
 - Kostenschätzung mit Hilfe von Kostenschätzverfahren und Risikoanalyse
 - Entscheidung über die Annahme basierend auf Kosten und Risiko

6.3.3. Änderungen planen

Überprüfung: Z.B. Vergleich geschätzter/tatsächlicher Aufwand

Ergebnis: Akzeptierte Änderungswünsche

6. Evolution

- 6.1. Überblick
- 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung

Predicted change impacts and change activities:	
Entities affected (highest level):	
Related entities (highest level):	
Documents affected:	
Impact scale:	Cosmetic <input type="checkbox"/> Minor <input type="checkbox"/> Major <input type="checkbox"/>
Level of risk:	
Estimated effort:	

6.3.3. Change Control Board

- ◆ Entscheidet über die Annahme von Änderungswünschen
- ◆ Typische Mitglieder (Abdeckung der Gesichtspunkte): Projektmanager, Entwicklung, Interner Projektsponsor, Kunde (manchmal)
- ◆ Manchmal nur für Eskalationsstufe, d.h. kleinere Änderungswünsche kann Projektmanager selbst entscheiden

6. Evolution

- 6.1. Überblick
- 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung

6.3.4. Änderung durchführen



Wer: Wartungsingenieur

Wie: Analyse der Auswirkungen von Änderungen basierend auf dokumentierten Beziehungen und Planungsergebnissen
 Durchführung der Änderung
 Anlegen einer Version für jede geänderte Komponente
 Anlegen einer Konfiguration für zusammenhängende Komponenten

Überprüfung: Anzahl der gefundenen Fehler bei Test, Inspektion

Ergebnis: Geändertes System, Tatsächlicher Aufwand

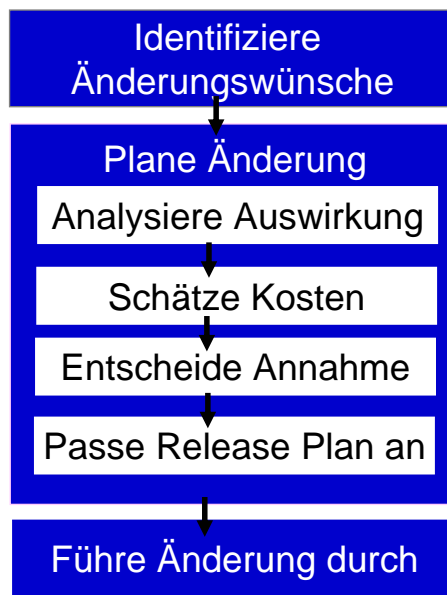
6. Evolution

- 6.1. Überblick
- 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung

6.3.4. Expliziter Änderungsprozess

6. Evolution

- 6.1. Überblick
- 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung



⇒ Änderungswünsche werden aufgedeckt
(gewährleistet Kundenzufriedenheit und Systemaktualität)

⇒ Risiko von Änderungen wird abgeschätzt
(gewährleistet, dass nur durchführbare Änderungen umgesetzt und Umsetzung genau geplant wird)

⇒ Akzeptierte Änderungen führen nicht zur Instabilität des Systems

⇒ Änderungen werden konsistent umgesetzt

Folie 37

6.4. Wiederverwendung

6. Evolution

- 6.1. Überblick
- 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung

- ◆ Nutzung vorhandener Komponenten in verschiedenen Umgebungen, z.B. Klassenbibliothek
- ◆ **Ziel:** Qualitätssteigerung (Komponente nur einmal prüfen) und Aufwandsreduzierung
- ◆ **Problem:**
 - NIH (Not Invented Here)
 - Anpassungsaufwand (direkte Wiederverwendung selten möglich)

Folie 38

6.4. Produktlinienentwicklung

- ◆ **Praktikable Wiederverwendung:** innerhalb einer Produktlinie
- ◆ Betrachte Produkte, die dieselben Komponenten beinhalten, als Varianten einer Produktlinie
- ◆ Plane und entwickle die **Gemeinsamkeiten und Variabilitäten** systematisch

Produktlinie



6. Evolution

- 6.1. Überblick
- 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung

6. Zusammenfassung Evolution

- ◆ Wird ermöglicht durch **datenbank-ähnliche Werkzeuge**, die komplexe Ableitungen, (verteilten) Zugriff und Änderungshistorie unterstützen.
- ◆ Entscheidung über Sinnhaftigkeit einer Änderung kann nur von Experten gefällt werden. In großen Unternehmen ist dazu expliziter **Änderungsprozess** nötig.

6. Evolution

- 6.1. Überblick
- 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung

- ◆ Karol Fruehauf, Jochen Ludewig, Helmut Sandmayr: Software-Projektmanagement und – Qualitätssicherung, VDF Verlag 2002
- ◆ Ian Sommerville, Software Engineering, Addison Wesley, 2003

6. Evolution

- 6.1. Überblick
- 6.2. Konfigurationsmanagement
- 6.3. Änderungsmanagement
- 6.4. Wiederverwendung