# TMS II

# **Beariffsdefinitionen**

## Konfigurationsmanagement

KM stellt sicher, dass Produkte eindeutig identifizierbar sind, Zusammenhänge und Unterschiede von verschiedenen Versionen einer Konfiguration erkennbar bleiben und Produktänderungen nur kontrolliert durchgeführt werden können. Arten der Durchführung: - Automatisch (z.B. Git)

- Semi-automatisch (z.B. unterstützende Plugins)

### **Build-Management**

Unter Build-Management wird diejenige Funktion verstanden, die alle Bauteile einer Konfiguration erzeugt, die nicht direkt vom Benutzer erstellt oder durch das System vorgegeben wurden. Zentrale Fragen: Wie ist die Software zusammengestellt? Welche Abhängigkeiten müssen berücksichtigt werden? Beispiele für klassische Abhängigkeiten: o Entwicklungsumgebung: Editor, Generatoren, Compiler, Linker, . . .

- o Basissoftware: LAN, ...
- o Betriebssystem
- o Hardware

#### Change-Management

Im Änderungswesen muss definiert werden, wie Änderungswünsche (Change Requests) erfasst werden, wie und durch wen diese bewertet werden und wie die Durchführung von Änderungen zu erfolgen hat.

- Versions-Management (Wer hat wann was geändert?) Versionsmanagement befasst sich (in erster Linie mit der Verwaltung der zeitlich aufeinander folgenden Revisionen

## eines Dokuments.) Release-Management

Ein Release ist eine an Kunden ausgelieferte Konfiguration eines (Software-)Systems, bestehend aus ausführbaren Programmen, Bibliotheken, Dokumentation, Quelltexten, Installationsskripten und so weiter. Das

Release-Managementdokumentiert ausgelieferte Konfigurationen und stellt deren Rekonstruierbarkeit sicher. Geplant

### Produktzustände

#### In Bearbeitung

Das Produkt wird bearbeitet. Es befindet sich entweder im privaten Entwicklungsbereich des Entwicklers oder unter Kontrolle des Entwicklers innerhalb der Produktbibliothek.

### Vorgelegt

Das Produkt ist aus der Sicht des Erstellers fertig und wird der Konfigurationsverwaltung übergeben. Ab jetzt kann es einer Prüfung durch die Qualitätssicherung unterzogen werden. Wird das Produkt hierbei abgelehnt, so geht es wieder in den Zustand in Bearb. zurück, andernfalls rückt es in den Zustand akzeptiert vor. Ab dem Zustand vorgelegt an kann der Ersteller nur unter Fortschreibung der Versionsangabe Modifikationen durchführen.

### Akzeptiert

Das Produkt wurde durch die QS überprüft und freigegeben. Es darf nur noch innerhalb einer neuen Version geändert werden.

### CTL

#### Pfadoperatoren:

- $A\phi$  auf allen Pfaden folgt  $\phi$  (englisch: All)
- E  $\phi$  auf mindestens einem Pfad folgt  $\phi$  (englisch: Exists)

### Pfad-spezifische Operatoren:

- $X\phi$  unmittelbar folgt  $\phi$  (englisch: neXt state)
- $F\phi$ -irgendwann folgt  $\phi$  (englisch: some Future state oder
- $G\phi$  auf dem folgenden Pfad folgt in jedem Zustand  $\phi$ (englisch: Globally)
- $\phi U\psi \phi$  folgt bis zum Erreichen des Zustands  $\psi$ (enalisch: Until)
- $\phi W \psi \phi$  folgt immer oder bis zum Erreichen des Zustands  $\psi$  (englisch: Weak Until)
- $EX\phi$  in (mind.) einem nächsten Zustand gilt  $\phi$
- $EF\phi$  in (mind.) einem der folgenden Zustände gilt  $\phi$
- EG  $\phi$  es gibt (mind.) einen Pfad, so dass  $\phi$  entlang des ganzen Pfades gilt
- $E[\phi U\psi]$  es gibt einen Pfad, für den gilt: bis zum ersten Auftreten von  $\psi$  gilt  $\phi$
- $AX\phi$  in jedem nächsten Zustand gilt  $\phi$
- AF  $\phi$  man erreicht immer einen Zustand, in dem  $\phi$  gilt
- $AG\phi$  auf allen Pfaden gilt in jedem Zustand  $\phi$
- $A[\phi U\psi]$  es gilt immer  $\phi$  bis zum ersten Auftreten von  $\psi$

### Semantik

$$T(s_0) \models \neg \phi \Leftrightarrow T(s_0) \not\models \phi$$
  
$$T(s_0) \models \phi \lor \psi \Leftrightarrow T(s_0) \models \phi \operatorname{oder} T(s_0) \models \psi$$

$$T(s_0) \models EX\phi \Leftrightarrow T(s_1) \models \phi$$

$$T(s_0) \models EG\phi \Leftrightarrow \forall i : T(s_i) \models \phi$$

$$T(s_0) \models EG\psi \quad \forall i : T(s_i) \models \psi$$

$$T(s_i) \models \phi EU\phi \quad \Rightarrow \quad \exists i : T(s_i) \models \psi$$

$$T(s_0) \models \phi E U \psi \quad \Leftrightarrow \quad \exists k : T(s_k) \models \psi \land \forall i < k : T(s_i) \models \phi$$

#### Transformationen

- $-\neg A\phi \equiv E\neg \phi$
- $\neg AF\phi \equiv EG\neg \phi$
- $\neg EF\phi \equiv AG\neg \phi$
- $\neg AX\phi \equiv EX\neg \phi$
- $-AG\phi \equiv \phi \wedge AXAG\phi$
- $-EG\phi \equiv \phi \wedge EXEG\phi$
- $-AF\phi \equiv \phi \lor AXAF\phi$
- $-EF\phi \equiv \phi \lor EXEF\phi$
- $-A[\phi U\psi] \equiv \psi \lor (\phi \land AXA[\phi U\psi])$
- $-E[\phi U\psi] \equiv \psi \vee (\phi \wedge EXE[\phi U\psi])$

#### Wahrheitstabelle

p	q	$p \wedge q$	$p \lor q$	$p \rightarrow q$
T	T	T	T	T
T	F	F	T	F
F	T	F	T	T
F	F	F	F	T

# Aufwandsschätzung

### **Entwicklungsleistung bekannt**

Gesamtleistung  $N_G(t) = f(t) + k \cdot \int f(t) \cdot dt + C$ Wartungsleistung  $N_W(t) = k \cdot \int f(t) \cdot dt + C$ Entwicklungsleistung  $N_E(t) = f(t)$ Gesamtarbeit  $A_G(t) = A_E(t) + A_W(t)$ Entwicklungsarbeit  $A_E(t) = \int f(t) \cdot dt + C$ Wartungsarbeit  $A_W(t) = k \cdot \int \left[ \int f(t) \cdot dt + C_1 \right] \cdot dt + C_2$ 

### **Gesamtleistung bekannt**

Gesamtleistung  $g(t) = N_q = [Mann]$  Anzahl Leute Wartungsleistung

$$N_W(t) = k \cdot e^{-kt} \left[ \int e^{kt} \cdot g(t)dt + C \right] \stackrel{C=0}{=} k \cdot e^{-kt} \cdot N_G \cdot \int_0^t e^{kt}dt = N_G \left( 1 - e^{-kt} \right)$$

Entwicklungsleistung

$$N_E(t) = N_G - N_W(t) = N_G \cdot e^{-kt}$$
 Gesamtarbeit  $A_G = N_G \cdot t$ 

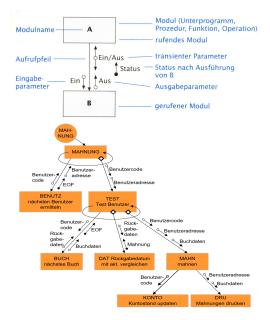
Entwicklungsarbeit  $A_E(t) = \frac{1}{h} \cdot N_G \left(1 - e^{-kt}\right)$ Wartungsarbeit  $A_W(t) = A_G(t) - A_E(t) = N_G \cdot t - A_E(t) = N_G \left[ t - \frac{1}{k} \cdot (1 - e^{-kt}) \right]$ 

### **Entwicklungsarbeit bekannt**

Gesamtleistung  $N_G(t) = N_E(t) + N_W(t)$ Wartungsleistung  $N_W(t) = k \cdot h(t) + C$ Entwicklungsleistung  $N_E(t) = \frac{dh(t)}{dt}$  Gesamtarbeit  $A_G(t) = h(t) + A_W(t)$ Entwicklungsarbeit  $A_E(t) = h(t)$ Wartungsarbeit  $A_W(t) = \int [k \cdot h(t) + C] \cdot dt + C_1$  mit  $\left[rac{Mann}{Mannjahr}
ight]$  Wartungsintensität

# **Reverse Engineering**

### **Structure Chart**



## **Nesting tree**

Indent tree TODO Maybe add picture

# Datenbanken

### Normalformen

#### 1. Normalform

Jedem Datenfeld eines Datensatzes darf höchstens ein Wert zugewiesen sein. D.h. es dürfen keine Mehrfacheinträge in einem Datenfeld vorliegen und Attribute müssen atomar

### 1. Normalform

		1		1-		-
KdNr	Name	GebJahr	Alter		Bankverbindung	Н
1	Huwaldt, Alexander	1965	36		430002345, Kreissparkasse XYZ, BLZ 850340200	Ŀ
2	Riedel, Toralf	1964	35	!	672009001, SpardaBank ABC, BLZ 850200310	Н
		i				l:
				١_		

KdNr	Vorname	Name	GebJahr	Alter	Kreditinstitut	BLZ	Konto
1	Alexander	Huwaldt	1965	36	Kreissparkasse XYZ	850340200	430002345
2	Toralf	Riedel	1964	35	SpardaBank ABC	850200310	672009001

sein.

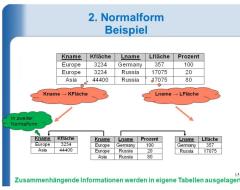
#### 2. Normalform

Erfüllt 1. Normalform

Eine Tabelle enthält nur Daten eines Themen- bzw. Informationsbereiches.

Aufteilung in mehrere Tabellen nach Themen/ Informationsgebieten.

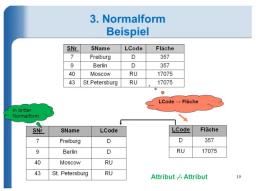
Jedes Nicht-Schlüsselfeld muss durch ein Schlüsselfeld identifizierbar sein und vom gesamten Schlüssel abhängen. Überprüfen und ggf. neue Schlüsselfelder hinzufügen oder zusammengesetzte Schlüssel definieren.



#### 3. Normalform

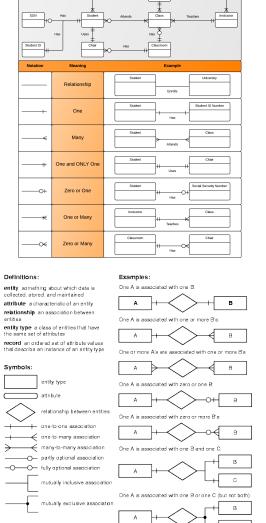
Ist dann erfüllt, wenn die Tabelle die 2. Normalform erfüllt Es dürfen keine transitiven (indirekten) Abhängigkeiten

Entfernung aller transitiven Abhängigkeiten durch Aufspalten der Tabelle in mehrere Tabellen, in denen alle Nicht-Schlüsselfelder direkt vom gesamten Schlüsselfeld abhängen



ERD "Crow's Foot" Relationship Symbols [Quick Reference]

# **ERD**



# GAIA

# **Role Model Template**

Role Schema:	
Description:	
Protocols and Activities:	
Permissions:	- reads:
	- changes:
	- generates:
Responsibilities:	- Liveness:
	- Safety:

# **Template for Interaction Model**

[Protoc	[Inputs]
[Initiator]	
[Desc	[Outputs]

· Description: Beschreibung

Initiator: RolleResponder: Rolle

· Inputs: Informationen des Auslösers

· Outputs: Informationen des Beantworters

## **Operatoren Ablaufbeschreibung**

Operator	Interpretation
x.y	x followed by y
x y	x or y occurs
$x^*$	x occurs 0 or more times
X+	x occurs 1 or more times
$x^{\omega}$	x occurs innately often
[x]	x is optional
x  y	x and y are interleaved(verschachtelt)

### Service Model

00.1100000.							
Service	Inputs	Outputs	Precondition	Post-condition			