



Nicht-funktionale Anforderungen und Quality of Service

AK Requirements am 19.09.2005

Michael Brenner
Dr. Markus Garschhammer
Munich Network Management Team

Dr. Friederike Nickl
Sepis GmbH, ein Unternehmen der Swiss Life

- Funktionale und nicht-funktionale Anforderungen
- Standard-Gliederungen – Beispiel: VOLERE Template
- Qualitätsmaße zu nicht-funktionalen Anforderungen
- Was hat das mit IT Service Management zu tun?
- Quality of Service
- Requirements und IT Service Management
- Beispiele von QoS-Parametern
- Service Provider Anforderungen
- Ausblick und Zusammenfassung

Funktionale und nicht-funktionale Anforderungen

Definition nach
Suzanne und James Robertson (VOLERE-Template)

Functional Requirement

An action that the product must be able to take, something that the product must do

Non-functional Requirement

A property of quality that the eventual product must have.

Es handelt sich um Charakteristika oder Qualitäten, welche das System attraktiv, benutzbar, schnell oder verlässlich wirken lassen.

Wiederverwendbarkeit nicht-funktionaler Anforderungen

- Nicht-funktionale Anforderungen beschreiben Qualitätsaspekte, die generell für Software-Systeme relevant sind
- Nicht-funktionale Anforderungen haben ein großes Wiederverwendungspotential über konkrete Projekte hinaus
- daher sinnvoll
 - ⇒ Einsatz einer Standard-Gliederung (Qualitätsmodell)
 - ⇒ Einsatz einer Referenzdatenbank
(Bsp. IVENA-Prozess und Referenzdatenbank von SOPHIST)

VOLERE-Template

(Suzanne und James Robertson)



Kategorien nicht-funktionaler Requirements

- Look and Feel
- Benutzbarkeit
- Performance
 - Zeit- und Verbrauchsverhalten
 - Schutz und Zuverlässigkeit (Safety)
 - Verfügbarkeit
 - Genauigkeit
 - Robustheit/Belastbarkeit/Skalierbarkeit/Langlebigkeit
- Operational Reqs
 - Einsatzumgebung/techn. Umgebung/Nachbarsysteme
- Wartbarkeit
- Security Requirements
 - Zugriffssicherheit/Integrität/Vertraulichkeit
- Kulturelle und Politische Reqs
- Gesetzliche Reqs

Konflikte zwischen nicht-funktionalen Anforderungen

Konflikte zwischen Requirements müssen erkannt und gelöst werden

Beispiel: Schnelle Antwortzeit und Security Requirements
Schnelle Antwortzeit und Wartbarkeit, Änderbarkeit

Analyse beispielsweise durch

- Operationalisierung von vagen Requirements
- Darstellung in einer Zielstruktur

[z.B. wie im NFR Framework nach Mylopoulos, Chun, Nixon, Yu]

Qualitätsmaße zu nicht-funktionalen Anforderungen

Zur Überprüfung des Erfüllungsgrades nicht-funktionaler Anforderungen müssen die Anforderungen messbar gemacht werden

Beispiel

Keine überprüfbare Anforderung: Das System soll schnell auf Benutzereingaben reagieren

Überprüfbar: Die Reaktionszeit des Systems auf Benutzereingaben soll unter 1 Sekunde liegen.

Was hat das mit IT Service Management zu tun?

In der Praxis beim Kunden oft zwei primäre Stakeholder bezüglich Requirements

- Anwender
 - primäre Quelle für funktionale Anforderungen
 - betreiben Systeme meist nicht selbst
 - nehmen Systemqualität im Kontext der täglichen Nutzung wahr
- Service Provider (externe und unternehmensinterne)
 - installieren, betreiben und warten Systeme
 - stellen damit Dienste zur Verfügung (z.B. als ASP)
 - müssen oft bestimmte Dienstqualität (QoS) garantieren
 - tragen damit zur Erfüllung vieler nicht-funktionaler Anforderungen der Anwender bei bzw. sind dafür verantwortlich

→ Beide sind gleichermaßen wichtig!

Quality of Service



Quality of Service (QoS)

beschreibt die (nicht funktionale) Qualität die ein Dienst, gemessen an den Anforderungen des Dienst-Konsumenten bzw. Kunden, zum Zeitpunkt der Erbringung besitzt.

Service Level

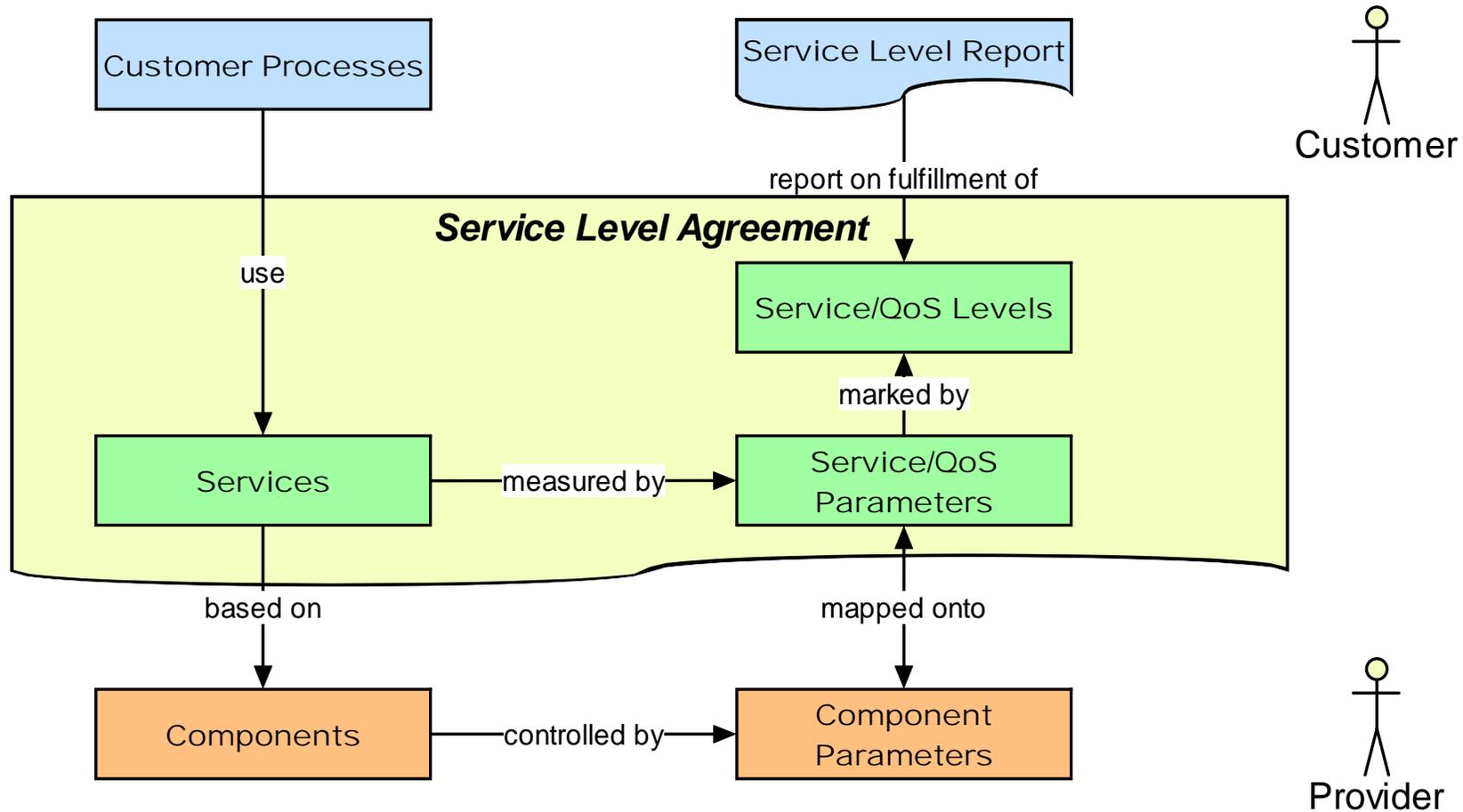
Meist in einem Service Level Agreement (SLA) festgelegte Wertebereiche der QoS-Parameter die eine bestimmte (z.B. „silver“) Dienstqualität repräsentieren.

IT Service Management (ITSM)

Alle Mittel zur Kontrolle und Steuerung einer effektiven und effizienten Dienstleistung. Diese können technischer (z.B. Network QoS Management) und organisatorischer Natur (z.B. Management-Geschäftsprozesse nach ITIL, eTOM) sein.

IT Service Level Management

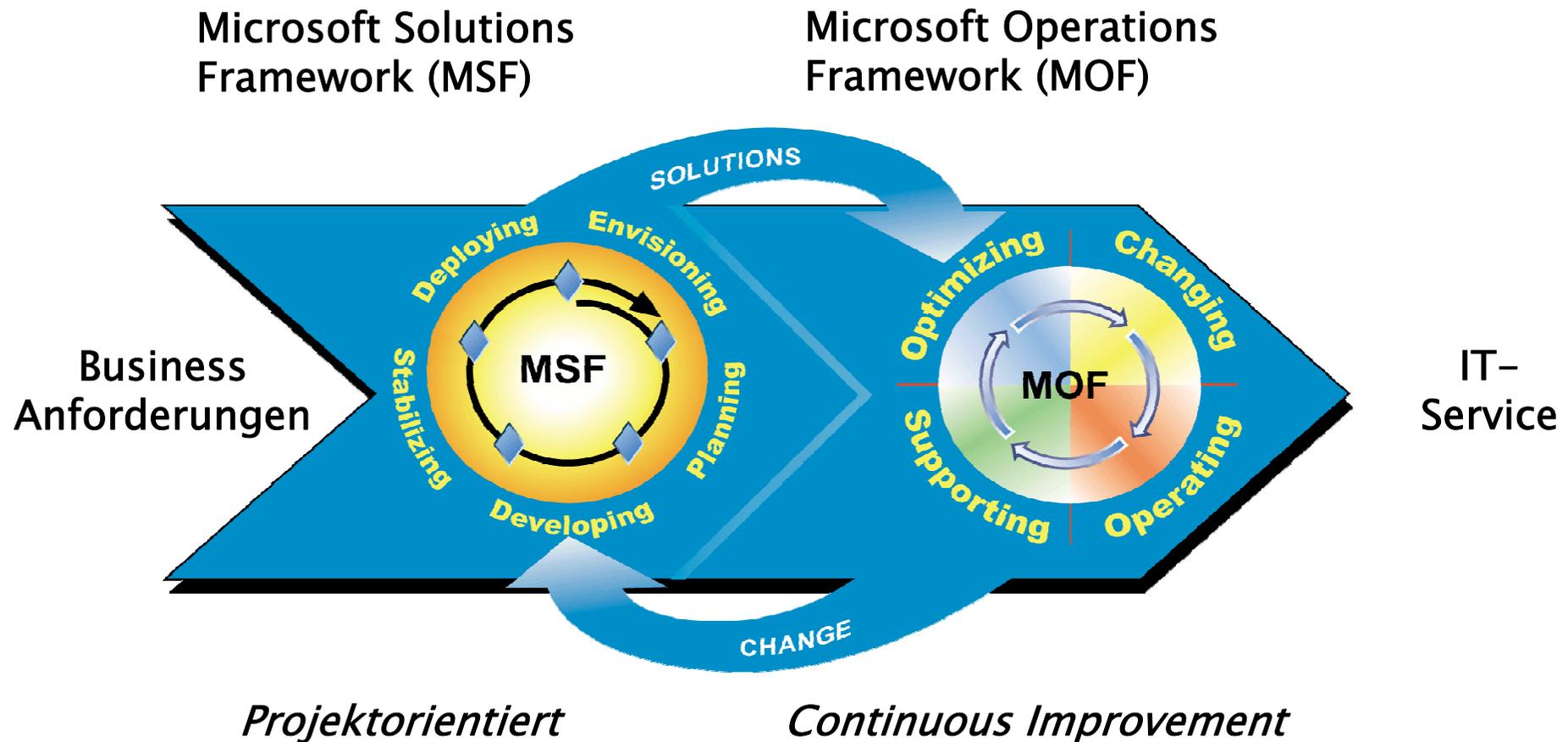
Begriffe



in Anlehnung an L. Lewis

Schnittstelle Entwicklung–Betrieb

am Beispiel der Frameworks MOF und MSF



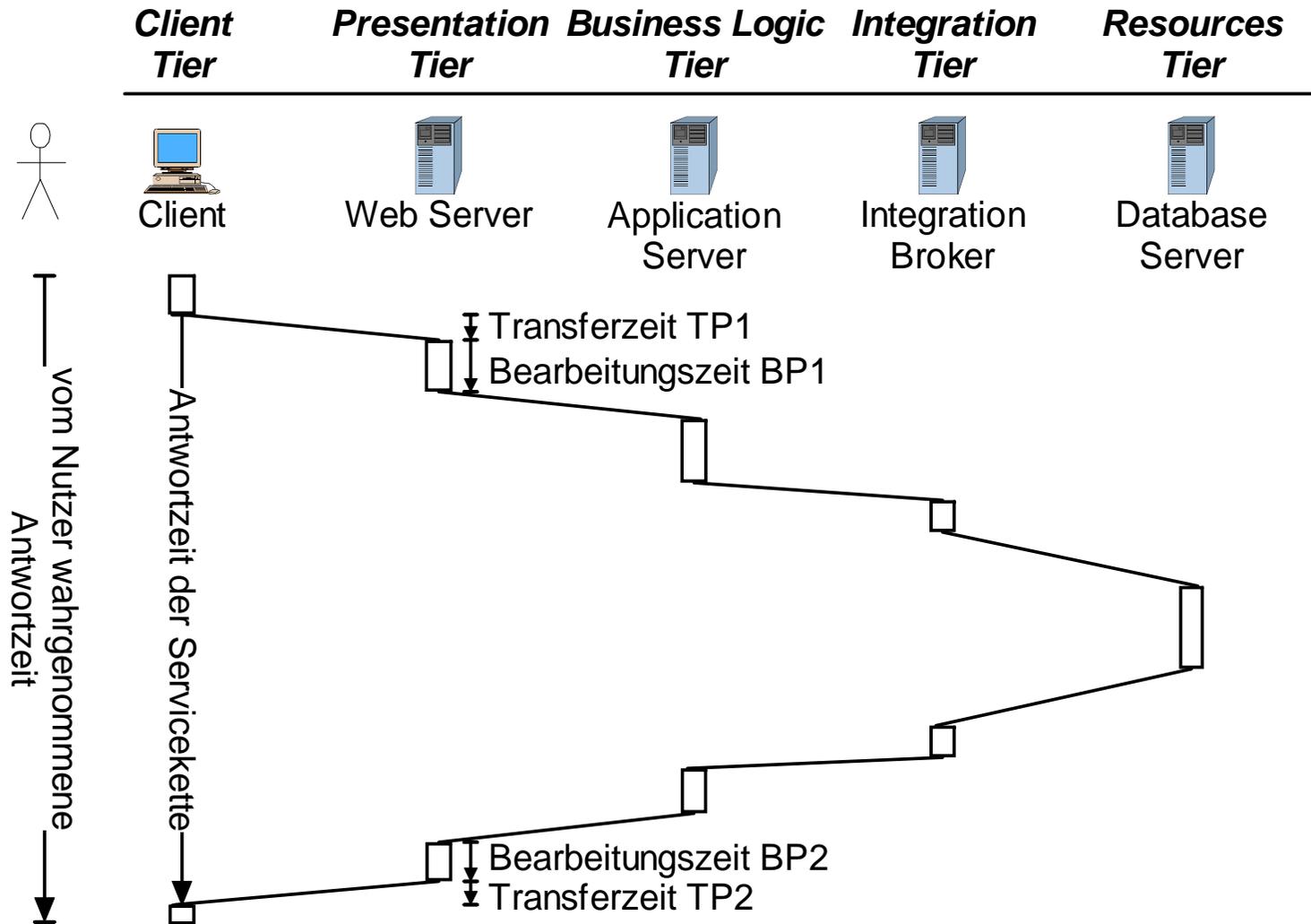
Quelle: Microsoft

Requirements und ITSM

Beispiele

Non-functional Requirement (nach VOLERE)	QoS-Parameter	Verantwortlicher ITIL-Prozess
Speed and Latency Requirements	z.B. Response Time	Capacity Management
Reliability and Availability Requirements	Availability	Availability Management
Supportability Requirements	<i>Kein technischer Parameter</i>	Service Desk, Incident Management

Antwortzeit



Antwortzeit



- Die vom Nutzer wahrgenommene Antwortzeit hängt wesentlich von zentralen Komponenten der produktiven Umgebung ab wie
 - dem Netz (Transferzeit)
 - den Datenbankservern
- Bei Anforderungen des Kunden an die Antwortzeit müssen diese auf Machbarkeit hinsichtlich der gegebenen produktiven Umgebung geprüft werden
- bei dieser Machbarkeitsanalyse ist das Expertenwissen des Service Provider Personals unerlässlich.

Availability – Verfügbarkeit

$$Availability = \frac{\sum Uptime}{\sum Agreed_Service_Time}$$

- 7 Tage Stillstand am Stück

$$\frac{358 \text{ Tage}}{365 \text{ Tage}} \geq 98\%$$

- jeden Tag 27 Minuten Stillstand

$$\sum_1^{365} \frac{27 \text{ min}}{365 \text{ Tage}} \geq 98\%$$

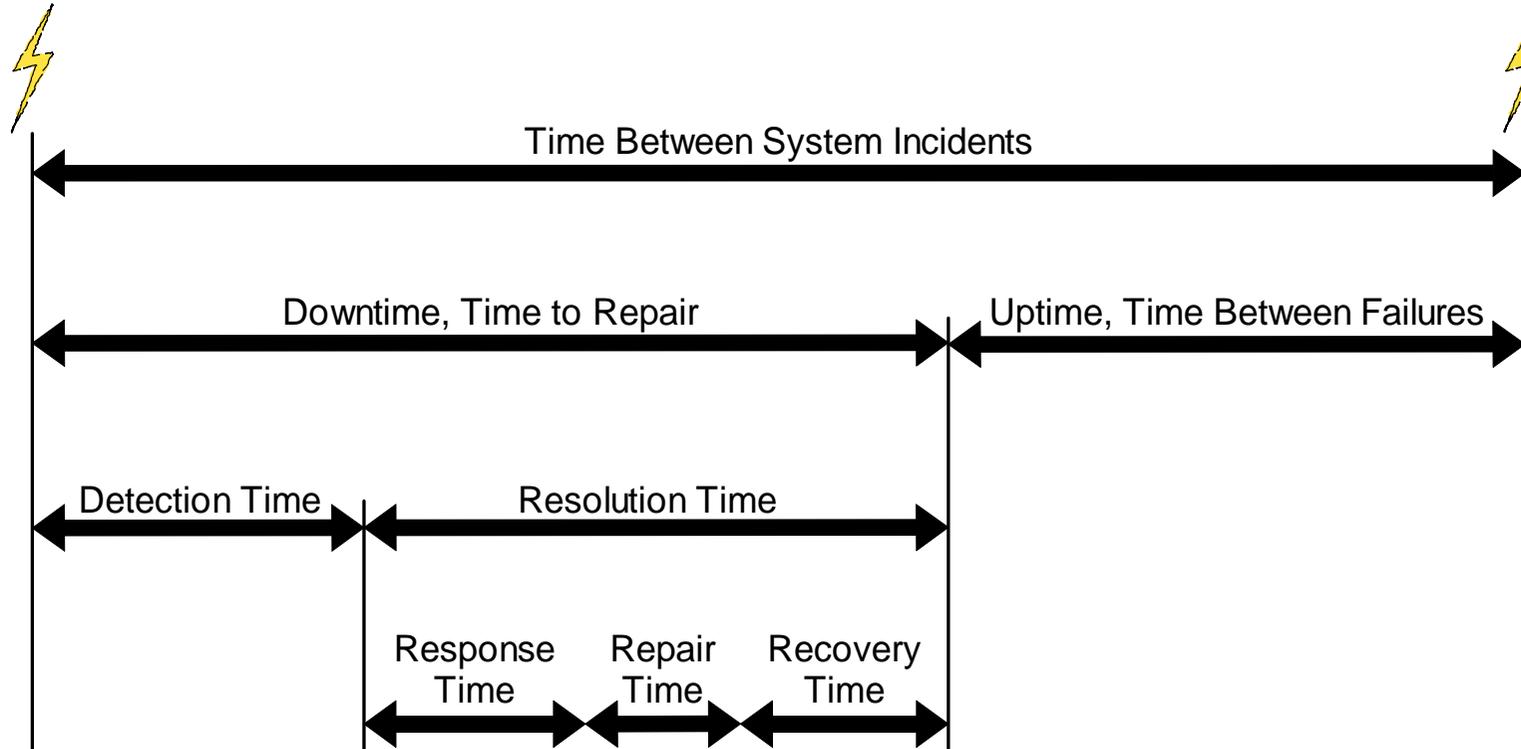
- sind 98% genug?
- sind 99,999% bezahlbar?
- reicht ein Parameter als Qualitätsmaß?

Availability – Verfügbarkeit

Mögliche Definitionsgrundlagen

System Incident

System Incident



$$\frac{\sum Uptime}{\sum Agreed_Service_Time} \geq 98\% \quad \wedge \quad \forall downtime \leq 27 \text{ min}$$

Quelle: OGC

Availability

Die Verfügbarkeit eines Systems ist abhängig

- von der Architektur des Gesamtsystems
- der Verfügbarkeit der beteiligten Komponenten (Netz, Server, Anwendungskomponenten, DB-Systeme)

Probleme:

- wie lassen sich aus der geforderten Verfügbarkeit des Gesamtsystems Anforderungen an die Verfügbarkeit der beteiligten Komponenten herleiten ?
- Wie läßt sich die Verfügbarkeit des Gesamtsystems aus den Verfügbarkeiten der beteiligten Komponenten ermitteln ? (Stichworte: Abhängigkeitsgraphen, CFIA, ...)

Rolle des Service Provider

Zusammenfassung



Mögliche eigene Anforderungen aus Service Provider Sicht

- Nicht-funktional: aus praktisch allen VOLERE-Kategorien (bis auf Look&Feel Req und Usability Req), z.B. Operational Req: Productization (aus *Release Mgmt*)
- Funktional: z.B. Logging-Funktionen, Management-Instrumentierung (z.B. zur Messung von Antwortzeiten)

Service Provider haben zwei wichtige Rollen im Entwicklungsprozess

- als Stakeholder mit eigenen, meist nicht-funktionalen Anforderungen
- als Experten und Partner zur Erfüllung nicht-funktionaler Anforderungen der Anwender

Ausblick

Umsetzung nicht-funktionaler Anforderungen



- Einsatz von Patterns und Architekturtaktiken (z.B. Wartbarkeit, Änderbarkeit, Security)
- Einsatz von Frameworks (z.B. Look&Feel, Usability, Security)
- frühzeitige Validierung von Architekturmodellen (Antwortzeit, Verfügbarkeit)
- frühzeitige Erstellung von Prototypen (Look&Feel, Usability)

Ausblick



Integration nicht-funktionaler Aspekte in Modellierungssprachen und -methoden

Beispiele für Security

- Prozessmodell PROSECO, Model Driven Security [Breu, Breu]
- Modellbasiertes Security Engineering mit UMLSec [Jürjens]

Sicherheitsanforderungen werden in das Modell integriert
Validierungen von Modellen nach Security-Aspekten sind (tool-
unterstützt) möglich

UML-Profile der OMG (Spezialisierungen der UML für bestimmte Einsatzbereiche)

- UML Profile for QoS and Fault Tolerance
- UML Profile for Schedulability, Performance and Time

Ausblick

Aktuelles Forschungsthema: Early Aspects



Aspekte sind in der Aspekt-orientierten Programmierung (AOP) Konstrukte, mit denen Querschnittsthemen (Cross Cutting Concerns) modularisiert und verkapselt werden können

Cross Cutting Concerns sind Querschnittsthemen wie Zugriffskontrolle, Logging, Fehlerbehandlung

Ziel: Separation of Concerns, Vermeidung von überall verstreuten Codesegmenten zu einem Concern

Weaving bedeutet das ‚Verweben‘ des zu einem Aspekt gehörigen Codes mit dem Programmcode an eigens dafür spezifizierten Join Points

Ausblick

Early Aspects und Nicht-funktionale Requirements



- **Early Aspects** sind Cross Cutting Concerns in den frühen Phasen einer Aspekt-Orientierten SW-Entwicklung
- **Cross cutting concerns** sind oft Maßnahmen zur Erfüllung nicht-funktionaler Requirements
(Beispiel: Berechtigungsprüfung als Security-Maßnahme)
- **Separation of Concerns** bedeutet hier: Getrennte Beschreibung mit Bezügen zu anderen Concerns (Join Points)
(Beispiel: separate Beschreibung der Berechtigungsprüfung, Angabe einer Liste der betroffenen Use Cases/Use Case Schritte)
- **Aktuelles Forschungsthema:** Wie lassen sich Cross Cutting Concerns in den frühen Phasen identifizieren und systematisch behandeln?

Zusammenfassung



- Service Provider sind bei der Einhaltung der Service Levels von Eigenschaften der betriebenen Systeme und Software abhängig
- Service Provider Anforderungen überdecken fast alle Kategorien nicht-funktionaler Requirements
- Durch Messung von QoS-Parametern wird die Erfüllung nicht-funktionaler Requirements wie Geschwindigkeit und Verfügbarkeit geprüft
- die Definition von QoS-Parametern wirft derzeitig noch viele Fragen auf
- nicht-funktionale Requirements und Querschnittsaspekte werden in gängigen SW-Entwicklungsmethoden eher vernachlässigt, sie sind aber ein Gegenstand aktueller Forschung