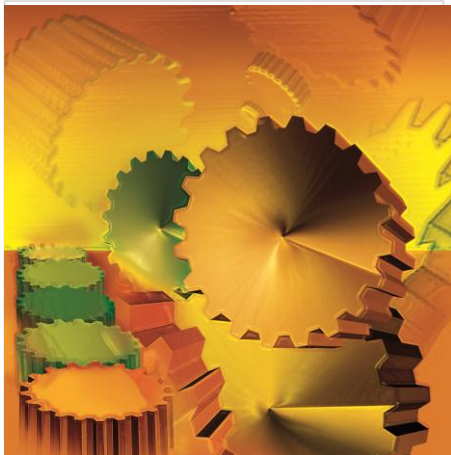


Software Engineering und Projektmanagement 2.0 VO



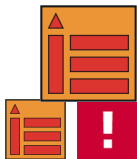
Vorlesung Inbetriebnahme & Wartung

2015W

Wolfgang Gruber

Roman Trabitsch

www.inso.tuwien.ac.at



INSO - Industrial Software

Institut für Rechnergestützte Automation | Fakultät für Informatik | Technische Universität Wien

Software Engineering und Projektmanagement VO

Agenda

1

Einführung

2

Projekt-Infrastruktur

3

Betrieb

4

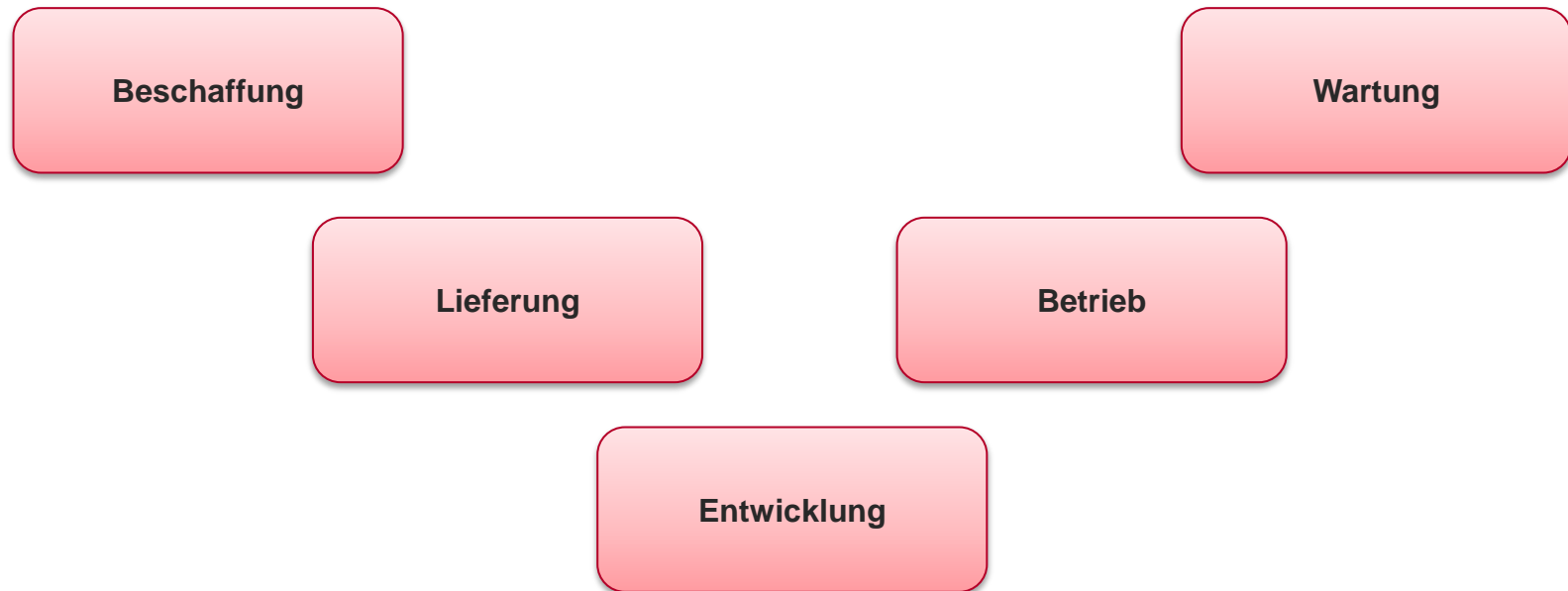
Release Management & Delivery

5

Wartung

Phasen des Software Lifecycle

„IEEE/EIA 12207.0 – Standard for Information Technology – Software Life Cycle Processes“ definiert fünf primäre Prozesse:



- **Zwei große Phasen nach der Entwicklung**
 - Betrieb
 - Wartung
- **Wesentlich für Erfolg**
- **Kostenintensiv**
- **Häufig vernachlässigt**

Definition Betriebs- & Wartungsphase

IEEE 610.12-1990 Standard Glossary of Software Engineering Terminology:

Operation and maintenance phase: The period of time in the software life cycle during which a software product is employed in its operational environment, monitored for satisfactory performance, and modified as necessary to correct problems or to respond to changing requirements.

Fragestellungen

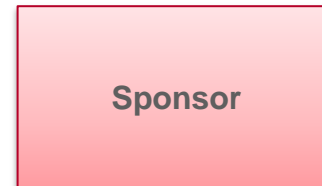
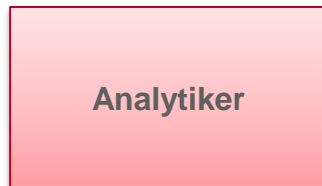
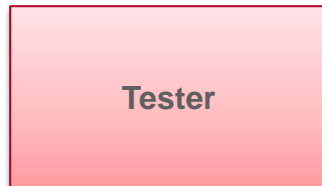
- **Worauf muss man bei der Einführung der erstellten Software beim Kunden achten?**
- **Wie wird die Software erfolgreich beim Kunden in Betrieb genommen?**
- **Welche Anforderungen ergeben sich beim Betrieb einer Applikation?**
- **Wie kann eine Software gewartet werden, um sie möglichst lange am Leben zu erhalten**

Szenario

Supplier



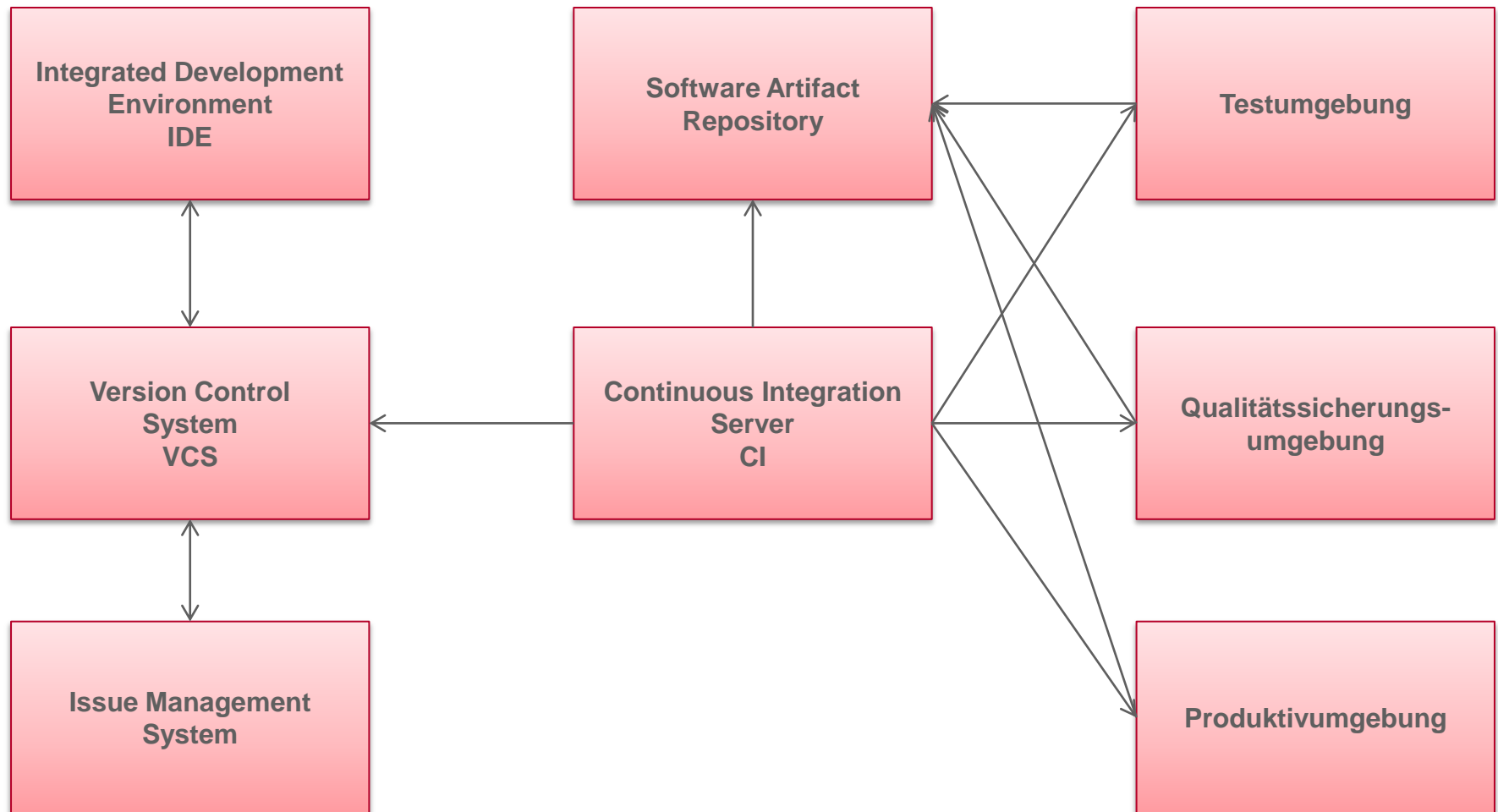
Customer



- **Rich Client Applications**
 - Individuelle Installationsroutinen
 - Komplexe Verteilung
- **Apps**
 - Zentrale Verteilung & Installation durch App Store
- **Web Applications**
 - Deployment auf eigenen Servern oder in Cloud
- **Embedded Systems**
 - Auslieferung mit Hardware

Oftmals Kombination unterschiedlicher Applikationstypen

Projekt-Infrastruktur

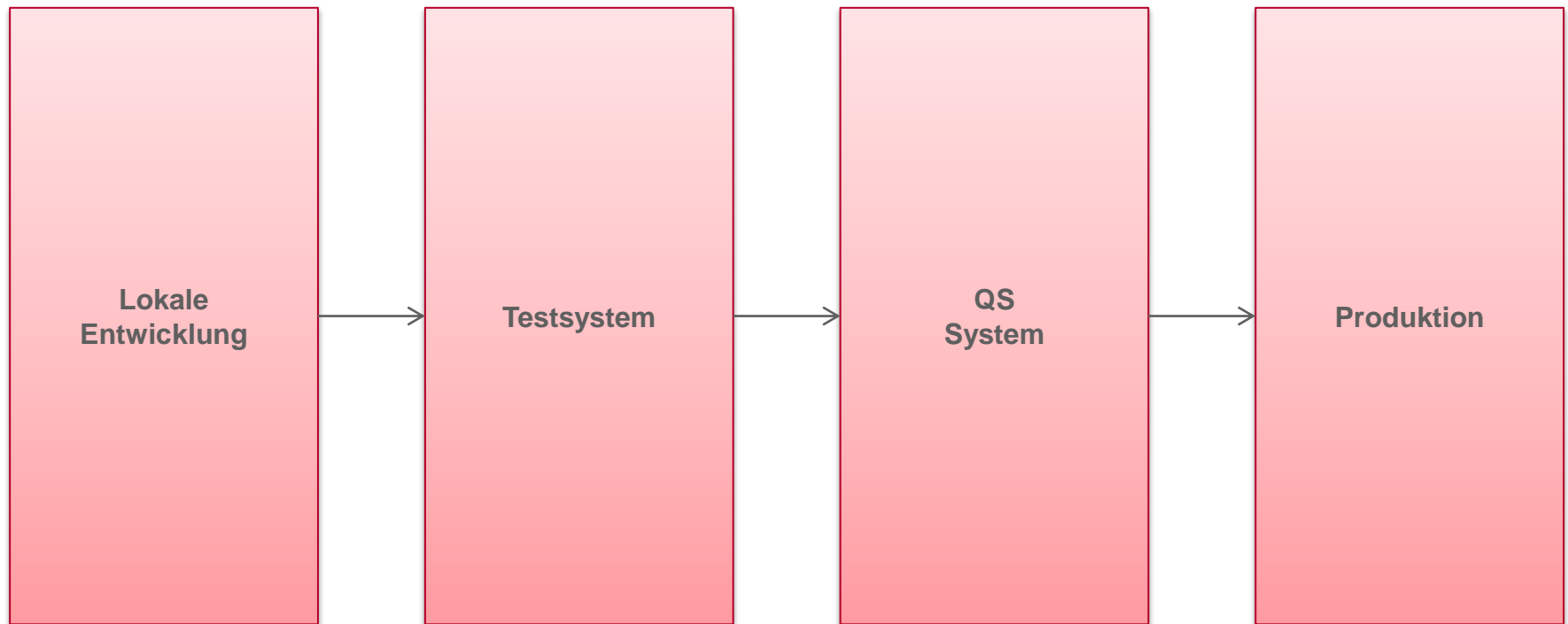


Tool-Chain (am Beispiel der Java-Plattform)

- **Integrated Development Environment**
 - Eclipse
 - Netbeans
- **Version Control System**
 - Subversion
 - git
- **Issue Management System**
 - Redmine
- **Continuous Integration Server**
 - Hudson / Jenkins
- **Software Artifact Repository**
 - Nexus
 - Artifactory

Development Stages

- Software durchläuft mehrere Stufen
- Qualität nimmt mit jeder Stufe zu



- **Zentrale Organisationseinheit, die für die Verwaltung der IT-Ressourcen in einem Unternehmen zuständig ist**
 - Verwaltung der Server (Rechenzentrum)
 - Verwaltung des Netzwerks
 - Oftmals auch Verwaltung der Arbeitsplatzrechner der Benutzer
 - Support-Dienstleistungen für Benutzer
- **Sehr stark formalisiert durch Prozess-Frameworks wie die „Information Technology Infrastructure Library“ (ITIL) bzw BS15000, ISO 20000**

- **Operator**
 - Laufende Überwachung der IT-Infrastruktur (24/7)
 - Systeme als Black Box
- **Netzwerk-Administrator**
 - Zuständig für Netzwerk-Infrastruktur
- **Technischer Administrator**
 - Administration einzelner Systeme (Applikation, DB, Server)
 - Low-Level-Zugriff: Log Files, Management Console
- **Fachlicher Administrator**
 - Administration von Applikationen über eigene User Interfaces
 - Meist kein Zugriff auf darunterliegende Infrastruktur
- **Support**
 - Einteilung in verschiedene Level (zB First Level, Second Level)
 - Ansprechpartner für Nutzer -> Eingang von Fehlermeldungen

Anforderungen des Betriebs

- **Hohe Stabilität & Fehlertoleranz**
- **Hohe Sicherheit**
- **Gute Dokumentation**
 - vor allem Referenzteil mit Checklisten, Fehlermeldungen, Prozessbeschreibungen
- **Wiederherstellbarkeit**
 - bei Restart oder Recovery
- **Analysierbarkeit**
- **Anpassbarkeit & Konfigurierbarkeit**
- **Geringer Verbrauch von Ressourcen**
 - Zum Beispiel vermeiden von Memory Leaks oder exzessiver CPU-Nutzung
- **Keine gegenseitige Beeinflussung von unabhängigen Applikationen**

Zwei wichtige Aufgaben während des Betriebs:

- Überwachung
- Optimierung

- **Überprüft Erreichbarkeit und ausreichende Erfüllung der Aufgaben der Applikation**
- **Erlaubt schnelle Reaktion bei Ausfällen**
- **Erlaubt internes Qualitätsbenchmarking**
 - Messung von Verfügbarkeit und Antwortzeiten
- **Notwendig zur Erfüllung von Service-Level Agreements (SLAs)**
 - Enthält Kriterien zur Einhaltung von nicht-funktionalen Anforderungen

Überwachungssoftware - Nagios

Service Status

All services

Host Status Summary

Up	Down	Unreachable	Pending
17	13	1	1
Unhandled	Problems	All	
13	14	32	

Last Updated: 2011-04-09 11:17:54

Service Status Summary

Ok	Warning	Unknown	Critical	Pending
74	4	8	16	1
Unhandled	Problems	All		
46	68	143		

Last Updated: 2011-04-09 11:17:55

Showing 1-25 of 46 total records

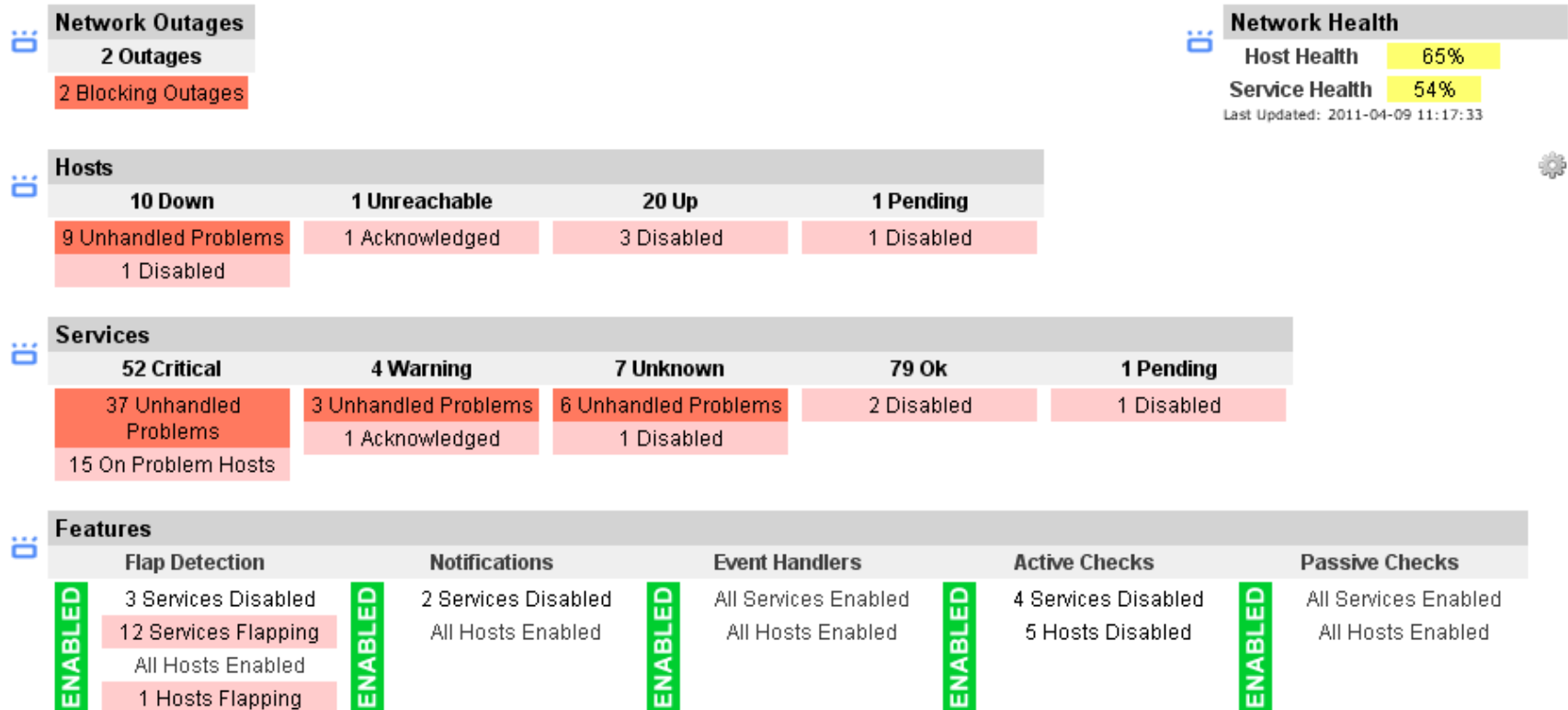
Filters: **Host**=Up **Service**=Warning,Unknown,Critical,Not Acknowledged,Not In Downtime

Search... Go

Host	Service	Status	Duration	Attempt	Last Check	Status Information
mstarr	Memory Usage	Critical	234d 6h 44m 45s	5/5	2011-04-09 11:17:19	CRITICAL - Socket timeout after 10 seconds
	test	Critical	234d 6h 43m 24s	5/5	2011-04-09 11:17:19	CRITICAL - Socket timeout after 10 seconds
	Drive C: Disk Usage	Critical	234d 6h 45m 36s	5/5	2011-04-09 11:17:19	CRITICAL - Socket timeout after 10 seconds
	FTP	Critical	234d 6h 44m 50s	5/5	2011-04-09 11:17:19	CRITICAL - Socket timeout after 10 seconds
	CPU Usage	Critical	234d 6h 46m 27s	5/5	2011-04-09 11:17:19	CRITICAL - Socket timeout after 10 seconds
192.168.1.253	Port 9 Status	Critical	82d 1h 35m 28s	5/5	2011-04-09 11:17:14	CRITICAL: Interface EtherNet Port on unit 1, port:9 (index 9) down due to lower layer being down.
www.cnn.com	Web Transaction	Critical	1m 7s	1/5	2011-04-09 11:16:47	WebInject CRITICAL - Test case number 1 failed
192.168.1.253	Port 26 Status	Critical	131d 19h 21m 46s	5/5	2011-04-09 11:16:09	CRITICAL: Interface EtherNet Port on unit 1, port:26 (index 26) down due to lower layer being down.
	Port 21 Status	Critical	131d 19h 23m 2s	5/5	2011-04-09 11:16:09	CRITICAL: Interface EtherNet Port on unit 1, port:21 (index 21) down due to lower layer being down.
	Port 12 Status	Critical	72d 3h 56m 45s	5/5	2011-04-09 11:16:09	CRITICAL: Interface EtherNet Port on unit 1, port:12 (index 12) down due to lower layer being down.
	Port 7 Status	Critical	72d 3h 56m 44s	5/5	2011-04-09 11:16:03	CRITICAL: Interface EtherNet Port on unit 1, port:7 (index 7) down due to lower layer being down.
www.nagios.com	HTTP	Warning	12m 35s	5/5	2011-04-09 11:15:19	HTTP WARNING: HTTP/1.1 404 Not Found
localhost	MySQL InnoDB Buffer Pool Hit Rate	Critical	234d 6h 48m 50s	5/5	2011-04-09 11:15:11	(Return code of 127 is out of bounds - plugin may be missing)
192.168.1.4	SQL Server	Unknown	28d 10h 57m 56s	5/5	2011-04-09 11:15:11	NSClient - ERROR: Invalid password.
192.168.1.253	Port 8 Status	Critical	131d 19h 23m 2s	5/5	2011-04-09 11:15:11	CRITICAL: Interface EtherNet Port on unit 1, port:8 (index 8) down due to lower layer being down.
	Port 19 Status	Critical	131d 19h 23m 11s	5/5	2011-04-09 11:15:00	CRITICAL: Interface EtherNet Port on unit 1, port:19 (index 19) down due to lower layer being down.
	Port 25 Status	Critical	131d 19h 23m 7s	5/5	2011-04-09 11:14:58	CRITICAL: Interface EtherNet Port on unit 1, port:25 (index 25) down due to lower layer being down.
	Port 11 Status	Critical	131d 19h 23m 11s	5/5	2011-04-09 11:14:57	CRITICAL: Interface EtherNet Port on unit 1, port:11 (index 11) down due to lower layer being down.
	Port 5 Status	Critical	131d 19h 23m 2s	5/5	2011-04-09 11:14:56	CRITICAL: Interface EtherNet Port on unit 1, port:5 (index 5) down due to lower layer being down.
	Port 22 Status	Critical	131d 19h 23m 2s	5/5	2011-04-09 11:14:56	CRITICAL: Interface EtherNet Port on unit 1, port:22 (index 22) down due to lower layer being down.

Überwachungssoftware - Nagios

Tactical Overview



- **Dient der optimalen Auslastung der vorhandenen Infrastruktur**
 - Beispiel: Zusammenlegung von Services auf einen Rechner
- **Balance zwischen**
 - Hardwareeinsparungen und
 - Performanceeinbußen und Fehleranfälligkeit
- **Server-Virtualisierung**
- **Achtung vor Single Point of Failures (SPOFs)!**

Release Management & Delivery

- **Beschäftigt sich mit dem Prozess der Auslieferung der Software**
- **Planung der Inbetriebnahme muss frühzeitig beginnen**
- **Erfolgreiche Releases sind Selbstverständlichkeit für Kunden**
- **Hohes Risikopotential**
- **Lange Zeit vernachlässigt**
- **Durch DevOps und Continuous Delivery in den Fokus von Software Engineering gerückt**
- **Release Prozess nicht nur für erstmalige Installation, sondern auch für Weiterentwicklung & Wartung**

- **Definition des Release Prozesses**
- **Planung des Release Prozesses**
- **Herstellen der Anwenderakzeptanz**
- **Planung & Durchführung der Abnahme**
- **Planung & Durchführung von Migrationen**
- **Inbetriebsetzung**

Wird erreicht durch:

- **Integration der Systemanwender**
- **Zielgruppengerechte Anwenderdokumentation**
- **Umfangreiche Schulungsmaßnahmen**
- **Graduelle Softwareeinführung**
- **Testbetrieb inklusive Anwenderfeedback**

- **Angst vor Änderung**
 - Angst vor dem Unbekannten
 - Angst vor Versagen
 - Angst die Kontrolle zu verlieren
 - Angst an Bedeutung zu verlieren
- **Angst führt zu Widerstand**
- **Diffusion of Innovation (DOI) Theory von E. M. Rogers**
 - Innovators – 2,5%
 - Early Adopters – 13,5%
 - Early Majority – 34%
 - Late Majority – 34%
 - Laggards – 16%
- **Innovators integrieren & fördern**

- **Zielgruppe: Endanwender**
 - Anwendergerechtes Verfassen der Dokumentation
 - Vermeidung von technischen Fachbegriffen
- **Funktionen der Dokumentation**
 - Einführungsmaterial
 - Nachschlagewerk / Referenz
- **Aufbau daher anhand der Anwendung bzw der Anwendersicht, nicht dem technischen Aufbau**
- **(Externe) Technische Dokumentation üblicherweise nicht Teil der Anwenderdokumentation**

- **Umfang abhängig von Applikationen**
- **Meist mehrere Nutzergruppen**
 - Unterschiedliche Funktionalitäten
 - Unterschiedliche Berechtigungen
- **Schulungskonzept**
 - Abhängig von den Nutzergruppen
 - Modularer Aufbau sinnvoll

- **Ziel: Praxisnahe Vermittlung des Funktionsumfang**
 - Interaktive Übungen
 - Überschaubare Gruppengrößen (max. 10 Personen)
 - Professionelle Trainer ideal
- **Schulungsmaterial an Schulungskonzept angepasst**

- **Aufwand steigt erheblich mit Komplexität**
 - Überprüfung auf Vollständigkeit und Korrektheit bei großen Systemen fast unmöglich
- **Basis: Vollständige Spezifikation aller Abnahmestufen**
 - WAS wird
 - WANN
 - von WEM
 - nach WELCHEN KRITERIEN abgenommen?
- **Abnahme bei größeren Systemen durch mehrere Stakeholder**

- **Nachvollziehbare und vollständige Dokumentation**
 - Des Abnahmeprozesses und
 - Der Ergebnisse
- **Auch: Definition von Revisionsprozessen**

Ebenen der Abnahme

- **Technische Abnahme**
 - Bestätigt Lauffähigkeit
 - z.B.: Entwickler-, Betriebs- und Funktionstests
- **Inhaltliche Abnahme**
 - Bestätigt semantische Korrektheit
 - z.B.: Black Box- und User Acceptance Tests
- **Integrationstests**
 - Bestätigt Zusammenwirken unterschiedlich abgenommener Module

- **Ablöse eines bestehenden Systems durch ein neues System**
- **Darf laufenden Betrieb nicht beeinflussen**
- **Muss einem geordneten Prozess unterliegen**

Arten von Migration

- **Softwaremigration**
 - Austausch einer bestehenden durch eine neue Software
- **Datenmigration**
 - Transfer von Daten von einem System in ein anderes
 - Verwendung von Extract-Transform-Load-Tools (ETL)
 - Änderung des Datenschemas
- **Hardwaremigration**
 - Umstellung der verwendeten Hardware
- **Oftmals Kombination aus verschiedenen Arten**
 - Beispiel: Softwaremigration erfordert gleichzeitig Datenmigration

- **Ziel: Heterogene Daten aus unterschiedlichen Quellen in einer Applikation zu verwenden**
- **Physische Integration: Die Daten werden in eine gemeinsame Datenbank kopiert**
- **Logische Integration: Die Daten werden nicht kopiert, jedoch bei Abruf konvertiert**

Definition Inbetriebsetzung

Definition der Inbetriebsetzung nach den *Allgemeinen Forderungen an die Qualitätssicherung* der KTA Deutschland:

Inbetriebsetzung ist die Gesamtheit der Maßnahmen, die zur erstmaligen Funktionsübernahme von Komponenten und Systemen an deren endgültigen Aufstellungsorten notwendig sind.

- **Aufwand abhängig von der benötigten Software- und Hardwarelandschaft**
 - Eigene Maschine(n)?
 - Eigene Datenbank
 - Integration von Fremdsystemen?
 -
- **Klärung von Verantwortlichkeiten**
 - Installation der Software
 - Einrichtung der Datenbank
 - Bereitstellung von Soft-/Hardware
 - Einrichtung von Benutzerkonten
 - Einrichtung von Netzwerkfreigaben
 -

- **Verantwortlichkeiten für formale Abnahme**
 - Technisch
 - Inhaltlich
- **Umfangreiche Planung unbedingt notwendig!**
 - Entsprechendes Controlling während der Inbetriebsetzung
- **Dokumentation des Inbetriebsetzung-Prozesses**
 - Release-Checkliste
 - Release-Drehbuch

- **Zusammenarbeit zwischen Entwicklungsteam und Betrieb**
- **Prozesse definieren oder adaptieren**
- **Verantwortlichkeiten festlegen**
- **Applikation für Betrieb adaptieren**
 - scheitert oft an Zeit und Kosten
- **Frühzeitiger Testrollout**

Release Anti-Patterns

- **Manuelle Installation der Software (Deployments)**
- **Deployments erst nach Abschluss der Entwicklung**
- **Manuelle Konfiguration der Betriebsumgebung**
- **Änderungen direkt in Betriebsumgebung (Hot-Fixes)**

- **Kontinuierliche Auslieferung von Software durch automatisierte Prozesse**
- **Beispiel: Tägliche Releases ohne Service-Unterbrechung bei Facebook**
- **Prinzipien**
 - Zuverlässiger & wiederholbarer Release-Prozess
 - Alles ist versioniert
 - „If it hurts, do it more frequently and bring the pain forward“
 - „Done means released“
- **Vorteile**
 - Höhere Zuverlässigkeit aufgrund weniger Fehler
 - Weniger Stress
 - Deutlich kürzere Cycle-Time

IEEE 610.12-1990 Standard Glossary of Software Engineering Terminology:

- (1) “The process of modifying a software system or component after delivery to correct faults, improve performance or other attributes, or adapt to a changed environment.”
- (2) “The process of retaining a hardware system or component in, or restoring it to, a state in which it can perform its required functions.”

Auslöser von Wartungstätigkeiten

- **Gefundene Fehler**
- **Performance-Verbesserungen**
- **Anpassung an geänderte Umgebung**
- **Update von verwendeten Bibliotheken & Frameworks**
- **Behebung von Sicherheitslücken**
- **Optimierung der Konfiguration**

- Umfang wird in SLAs geregelt
- Umfangreiche Wartungsarbeiten häufig eigenes Projekt
- Wiederholte Tätigkeit
- Wird oft nicht durch die ursprünglichen Software Entwickler durchgeführt
- Oftmals outsourced bzw. Junior Entwickler
- Wartbarkeit ist eine nicht-funktionale Anforderung, die bereits bei der Entwicklung berücksichtigt werden muss
- Analyse der Auswirkungen & Testen der Änderungen
- Wartung erhöht Lebenszeit einer Applikation

Wartung versus Weiterentwicklung

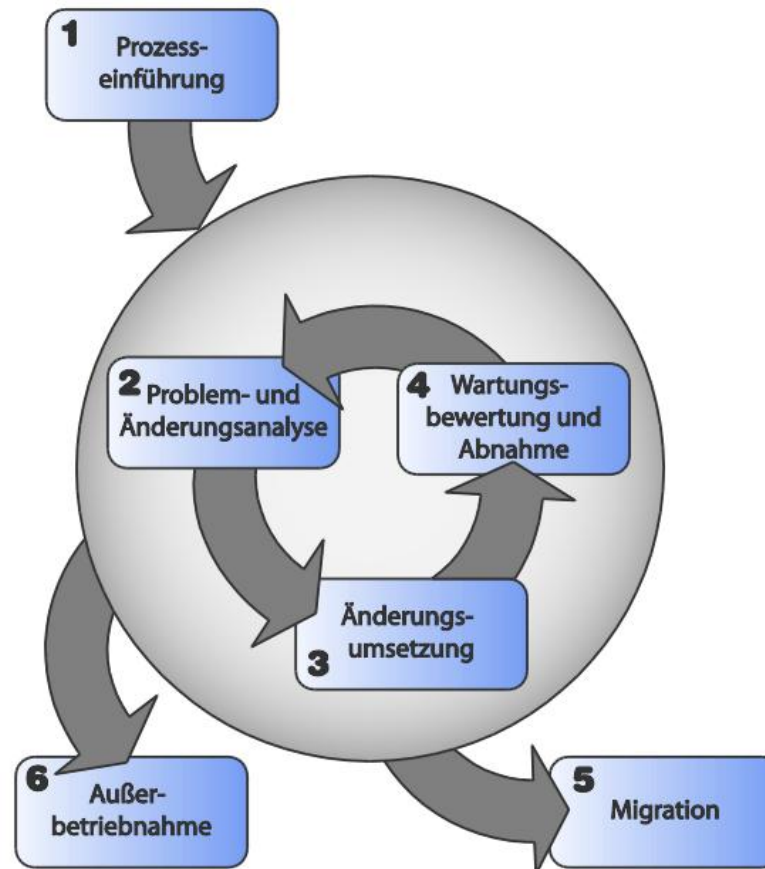
- **Wartung ist keine Weiterentwicklung**
- **Keine Änderung des funktionalen Zustands**
- **Wartung und Weiterentwicklung laufen parallel ab**
- **Eventuell verschiedene Teams für Wartung und Weiterentwicklung zuständig**

Definiert in IEEE 1219-1998 Standard for Software Maintenance

- **Korrektive Wartung**
 - Zur Behebung von Fehlern
 - Abweichungen von der Spezifikation
 - Nicht-Erfüllung von SLAs
- **Präventive Wartung**
 - Zur Verhinderung vermutlicher zukünftiger Fehler (vgl. Jahr-2000-Problem)

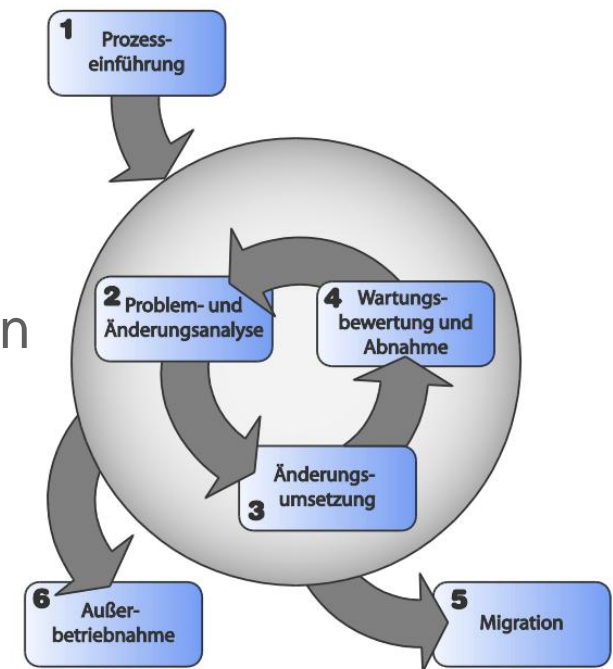
- **Adaptive Wartung**
 - Zur Anpassung an eine geänderte Umgebung
 - Änderung verwendeter Software
 - Änderung von Schnittstellen
 - Änderung der Hardware
- **Perfektionierende Wartung**
 - Zur Verbesserung der Applikation
 - Verbesserung der Performance
 - Verbesserung der Wartbarkeit
 - Verbesserung in Usability

Definiert in ISO/IEC 14764:2006 – Software Engineering – Software Life Cycle Processes – Maintenance



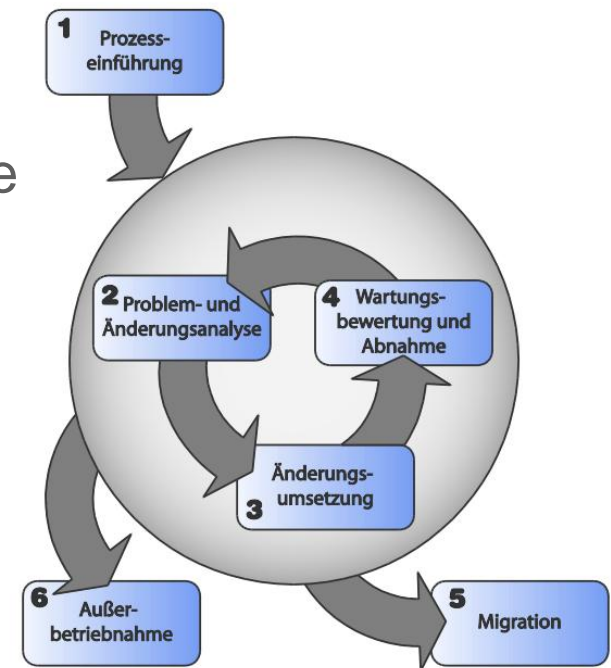
Wartungsprozess II

- **Prozesseinführung**
 - Planung eines allgemeinen Wartungsprozesses
 - Enthält alle Pläne & Vorgehensweisen
 - Wartungsplan
- **Problem- und Änderungsanalyse**
 - Start eines Wartungszyklus
 - Auslöser: Change Request
 - Analyse des Problems
 - Entwicklung von Änderungsmöglichkeiten
 - Auswahl einer Lösung
- **Änderungsumsetzung**
 - Gewählte Lösung umsetzen



Wartungsprozess III

- **Wartungsbewertung und Abnahme**
 - Prüfung der Umsetzung
 - Problem behoben?
 - Bei Erfolg: Übernahme in Produktivbetrieb
 - Alternativ: Neustart des Zyklus
- **Migration**
 - Transferiert Applikation in eine geänderte Umgebung
- **Außerbetriebnahme**
 - Betrieb wird eingestellt



- **Software Reengineering**
 - Neuentwicklung bei gleichbleibender Funktionalität
 - Motivation meistens Qualitätssteigerung
 - Hohes Abstraktionsniveau
- **Reverse Engineering**
 - Gewinnung von Code & Modellen aus vorhandenen Artefakten
 - Notwendig wenn Source Code nicht verfügbar (Urheberrecht)
 - Re-Dokumentation
 - Trägt zum Code Verständnis bei
 - Oft Vorstufe zum Reengineering

- **Refaktorisierung**

- Während Initialentwicklung & Wartung bedeutend
- Erhaltung der Wartbarkeit
- Nutzung definierter Refactorings
- Unterstützung durch IDEs

- **Anti-Patterns**

- „Gegenstück“ zu Software Patterns
- Sammlung häufig angetroffener schlechter Lösungsansätze
- Wurden aus mangelnder Erfahrung & Qualifikation gewählt
- Änderung von Software Entwicklungsphilosophie
- Dienen der schnellen Erkennung
- Bieten Verbesserungsstrategien

- **Inbetriebnahme & Wartung wichtige Aspekte erfolgreicher Software-Projekte**
- **Betrieb folgt anderen Regeln als Software Entwicklung**
- **Release Management sollte möglichst frühzeitig beginnen**
- **Benutzer-Akzeptanz, Schulung & Dokumentation sind wichtige Aspekte**
- **Formale Abnahme in SE Projekten notwendig**
- **Genaue Planung der Inbetriebnahme**
- **Projekte erfordern oftmals Migrationen**
- **Wartung verlängert Lebenszeit der Software**

Vielen Dank für Ihre Aufmerksamkeit!