oose.
Innovative Informatik

Impulsvortrag:

Steht alles im Wiki?
Das kleine 1x1 der Architekturdokumentation

Stefan Zörner, oose Innovative Informatik GmbH, Hamburg

Gesellschaft für Informatik e.V.,
Regionalgruppe Dortmund, 04.10.2010

Zusammenfassung

Steht alles im Wiki?

Das kleine 1x1 der Architekturdokumentation

Architekturdokumentation wird oft als lästige Pflicht angesehen. Dabei ermöglicht das angemessene Festhalten die Kommunikation Ihrer Konzepte im Team und dem Auftraggeber gegenüber überhaupt erst. Der Vortrag stellt anhand konkreter Beispiele bewährte Arbeitsergebnisse und mögliche Strukturierungen vor. Häufige Herausforderungen werden ebenso diskutiert wie typische Werkzeugketten. Wiki oder UML-Tool? Oder was dazwischen? Welche Notationen haben sich in der Praxis bewähren? Und wie kommt man falls verlangt jederzeit zu einer druckbaren Dokumentation?

Stefan Zörner, Stationen

- 1991-94 Ausbildung Math.-techn. Assistent bei der **Bayer AG**
- **Studium** Mathematik (Diplom 1998), Schwerpunkt Informatik
- 1998-2001 **Mummert + Partner** AG, Berater, u.a. Sun-Trainer
- 2001-2006 **IBM** e-business Innovation Center, IT-Architekt
- Seit Juli 2006: Berater und Trainer bei **oose** in Hamburg
  - Schwerpunkt: Softwareentwurf und Java-Technologien
  - Stefan.Zoerner@oose.de

Veröffentlichungen, Vorträge

- **Bücher** „Portlets“, 2006
  „LDAP für Java-Entwickler“, 3. Auflage 2007
- Artikel u.a. in Java Magazin und bei IBM developerWorks
- Vorträge bei JAX und W-JAX seit 2002, Advisory Board

Dies und das

- Seit 2005 Mitarbeit im **Apache** Directory Project, szoerner@apache.org
- iSAQB Certified Professional for Software Architecture
- OMG Certified UML Professional (Intermediate)
- SpringSource Certified Spring Professional

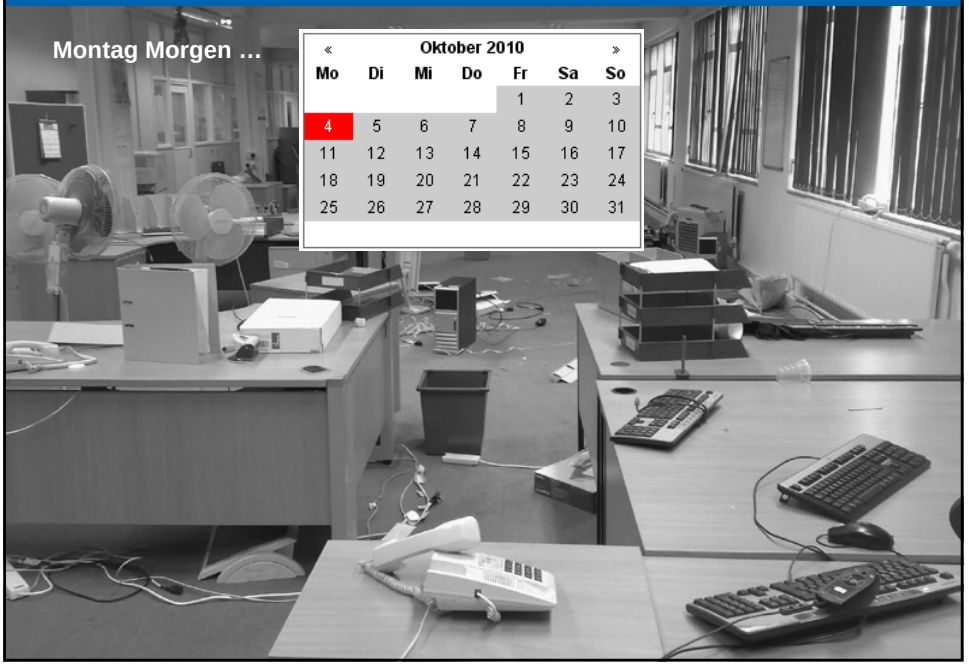
Steht alles im Wiki?
oose. Innovative Informatik

Agenda

Agenda

1. Montag Morgen
2. Entscheidungen
3. Sichten
4. Bewerten
5. Strukturieren
6. Schluss und Aus(-blick)

Steht alles im Wiki?
oose. Innovative Informatik



Steht alles im Wiki?

oose. Innovative Informatik

Fragen, die neue Mitarbeiter so stellen …

- Wie checke ich die Sourcen aus, und wie baue ich die Software?
- Warum sind bei mir Tests rot?
- Was brauche ich für Tools?
- Wenn ich neue Funktionalität hinzufügen soll – wie stelle ich das an? Hier ist doch schon was Ähnliches, kann ich das wiederverwenden?
- Was leistet das System überhaupt?
- Aus welchen Bestandteilen besteht die Software?
- Wie arbeiten diese zusammen?
- Ist das irgendwo beschrieben?
- Warum benutzt ihr noch JDK 1.3?
- Wieso habt Ihr das denn so gemacht?
- …

Steht alles im Wiki?

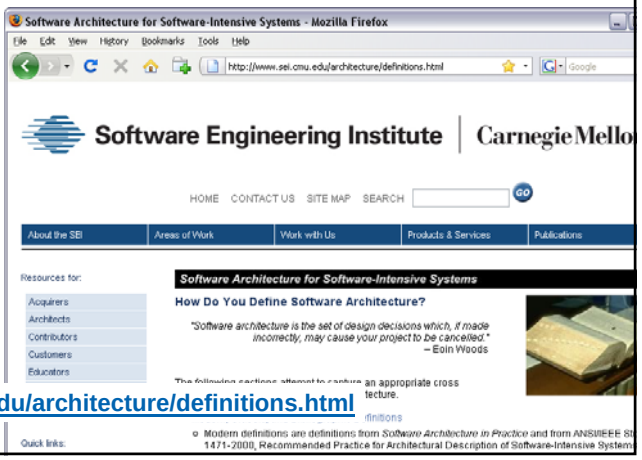
oose. Innovative Informatik

Antworten, die neue Mitarbeiter daraufhin erhalten

- Steht alles im Wiki.
- Das haben wir nicht dokumentiert – wir gehen agil vor.
- Das war schon so, als ich neu war.
- Das ist historisch gewachsen.

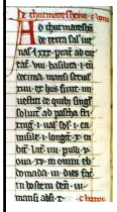
Definitionen zu Softwarearchitektur

- Es gibt nicht die eine allgemein akzeptierte Definition für Softwarearchitektur
- Das Software Engineering Institute (SEI) sammelt sogar Definitionen:



➔ http://www.sei.cmu.edu/architecture/definitions.html

Eine konkrete Definition



Architektur := $\sum$ wichtige Entscheidungen

Softwarearchitektur umfasst die Summe verschiedener wichtiger Entscheidungen über

- die Auswahl von Strukturelementen und deren Schnittstellen, aus denen das System zusammengesetzt ist
- das Verhalten und Zusammenspiel dieser Elemente
- den hierarchischen Aufbau von Subsystemen
- den zugrunde liegenden Architekturstil
- …

G. Booch, P. Krutchen, K. Bittner and R. Reitman. The Rational Unified Process — AnIntroduction. 1999.

Agenda

1. Montag Morgen
2. Entscheidungen
3. Sichten
4. Bewerten
5. Strukturieren
6. Schluss und Aus(-blick)

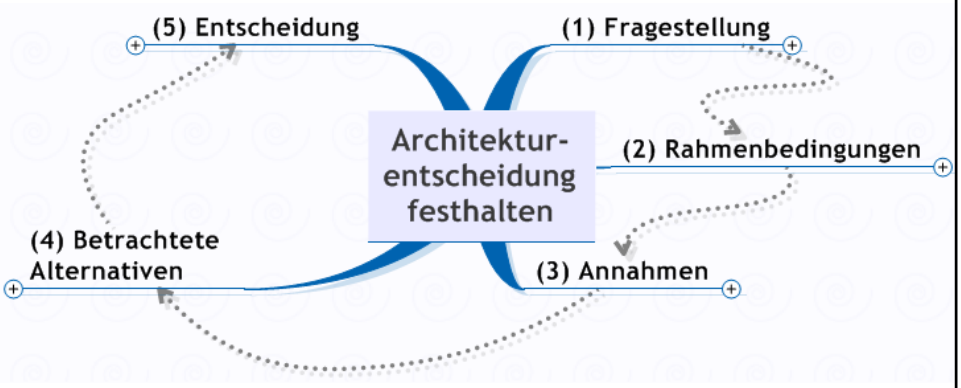
Steht alles im Wiki?
oose. Innovative Informatik

Architekturentscheidungen …

"Software architecture is the set of design decisions which, if made incorrectly, may cause your project to be cancelled." (Eoin Woods)

- Architekturentscheidungen sind diejenigen, die sich im weiteren Verlauf nur sehr schwer revidieren lassen.
- Konsequenzen: höhere Kosten, Zeitverlust, ggf. scheitert das Vorhaben

Entscheidungen treffen und festhalten. Ein Werkzeug



Fragestellung, Rahmenbedingungen

Fragestellung – Leitfragen

- Was genau ist das Problem?
- Warum ist es für die Architektur relevant?
- Welche Auswirkung hat die Entscheidung?

Rahmenbedingungen – Leitfragen

- Welche festen Randbedingungen haben wir einzuhalten?
- Welche Einflussfaktoren sind zu beachten?

Annahmen

“The life of a software architect is a long and rapid succession of suboptimal design decisions taken partly in the dark.” (Philippe Kruchten)

Annahmen – Leitfragen

- Welche Annahmen haben wir getroffen?
- Welche Annahmen können wie vorab überprüft werden?
- Mit welchen Risiken müssen wir rechnen?

Steht alles im Wiki?
oose. Innovative Informatik

Betrachtete Alternativen, Entscheidung

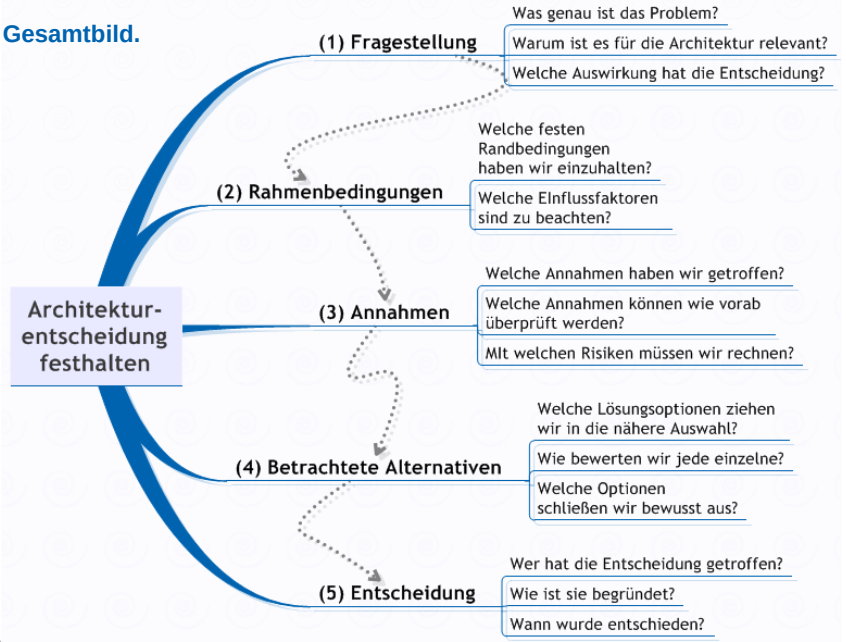
Betrachtete Alternativen – Leitfragen

- Welche Lösungsoptionen ziehen wir in die nähere Auswahl?
- Wie bewerten wir jede einzelne?
- Welche Optionen schließen wir bewusst aus?

Entscheidung – Leitfragen

- Wer hat die Entscheidung getroffen?
- Wie ist sie begründet?
- Wann wurde entschieden?

Gesamtbild.

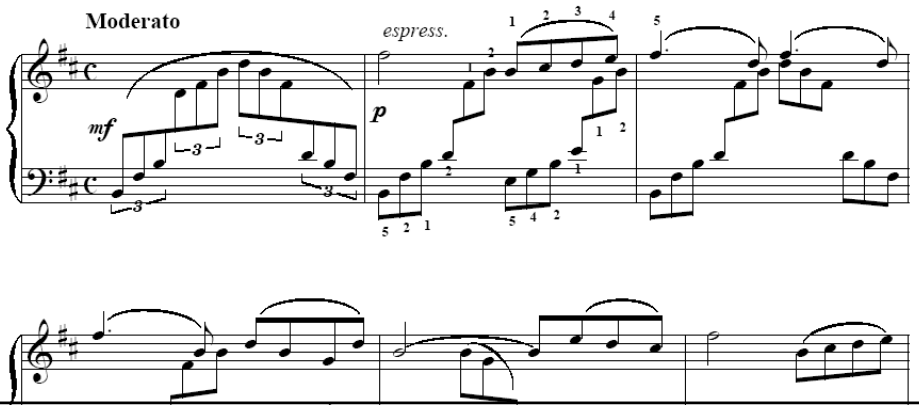


Steht alles im Wiki?
oose. Innovative Informatik

Agenda

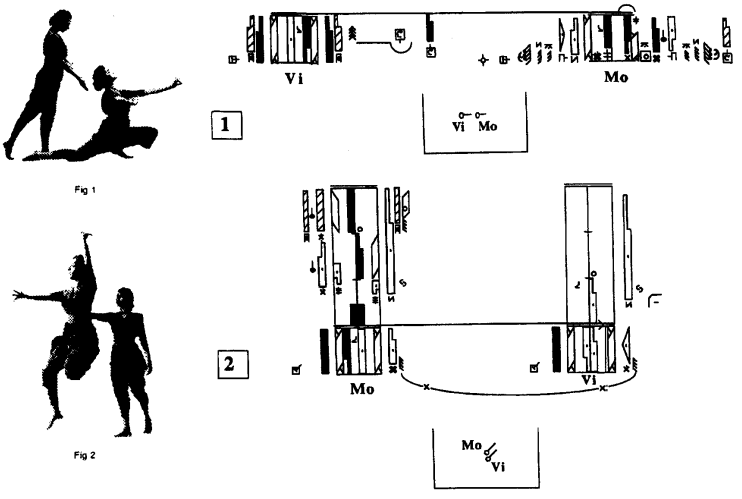
1. Montag Morgen
2. Entscheidungen
3. Sichten
4. Bewerten
5. Strukturieren
6. Schluss und Aus(-blick)

Schwanensee (1877)

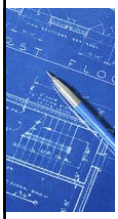


Steht alles im Wiki?
oose. Innovative Informatik

Beispiel Tanznotation



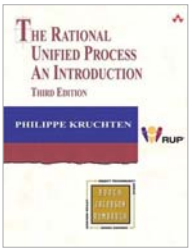
Analogie zur Softwarearchitektur: Views (Sichten)



- Es ist sinnvoll, bestimmte Aspekte einer Software mit Bilder statt textuell zu beschreiben
- Ein einzelnes Bild reicht in der Regel nicht aus
  - ➔ Unterschiedliche Sichten für unterschiedliche Aspekte

Beispiel: Rational Unified Process (P. Kruchten)

- 4 + 1 Views:
  - Logical View
  - Development View
  - Process View
  - Physical View
  - Scenarios



Steht alles im Wiki?
oose. Innovative Informatik

Literaturtipp zu dem Thema:

Effektive Software-Architekturen
Ein praktischer Leitfaden

von Gernot Starke
449 Seiten,
Hanser Fachbuch; 4. Auflage (2009)
ISBN 978-3446420083

Dort beschriebene Sichten (u.a.)

- Kontextsicht
- Bausteinsicht (= Struktur)
- Laufzeitsicht (= Verhalten, Dynamik)
- Verteilungssicht

Die Kontextsicht – Software agiert nicht allein …

- Stets gibt es Beteiligte außerhalb des Systems:
  - Anwendergruppen, die Funktionalität nutzen und erwarten
  - Fremdsysteme, die zur Ausführung erforderlich sind

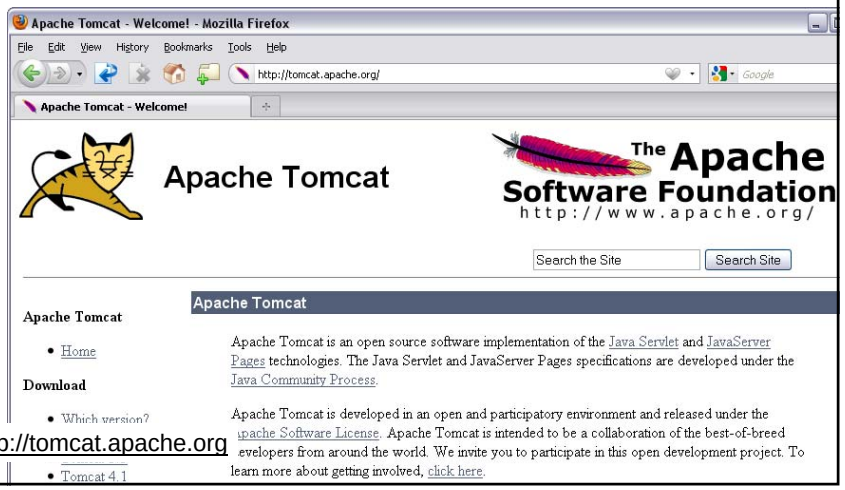
Der **Kontextsicht** zeigt das Umfeld, d.h. alle außerhalb des eigenen Systems liegenden Akteure, mit denen direkt kommuniziert wird.

Systemkontextdiagramm: Visualisierung des Umfelds

- das zu beschreibende System im Mittelpunkt als Blackbox
- drum herum die direkt beteiligten Benutzer und Fremdsysteme
- Verbindung zwischen einem solchen Akteur und dem System drückt Interaktion aus.

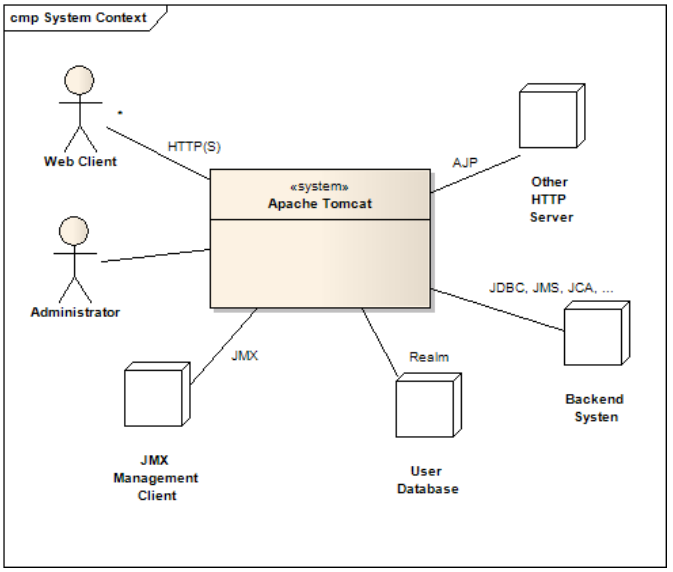
Fallbeispiel: Apache Tomcat

Apache Tomcat ist ein in Java geschriebener Webapplikationsserver zum Betreiben Java EE-konformer Webapplikationen.

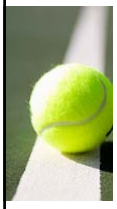


➔ http://tomcat.apache.org

Eine Kontextsicht für Apache Tomcat in UML.



Warum ist das so wichtig?



Potenzielle Schnittstellen zum Zielsystem finden!

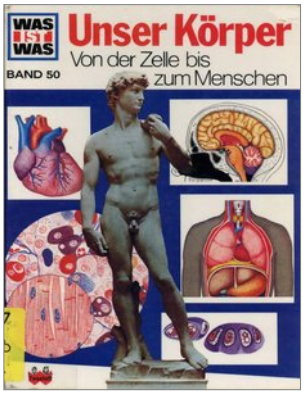
- Sie Anbindung eines Fremdsystems ist regelmäßig ein technisches Risiko.
- Solche frühzeitig zu erkennen kann entscheidend sein für den Erfolg Ihres Projektes.
- Architekturentscheidungen ableiten.

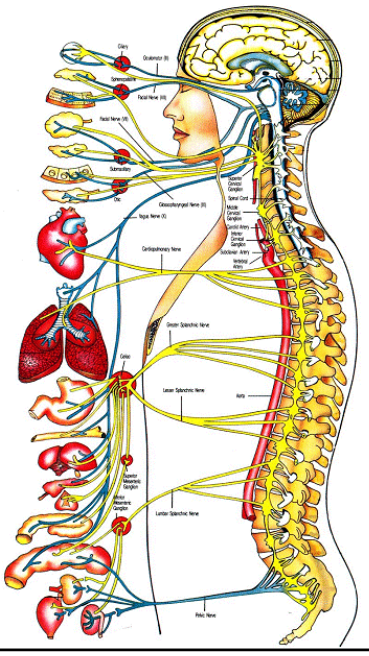
Startpunkt, um Verantwortlichkeiten zu klären

- Welche Fremdsysteme müssen wir integrieren?
- Was leistet unser System, und vor allem: was leistet es nicht?
- Wo ist die Systemgrenze?

Steht alles im Wiki?

oose. Innovative Informatik





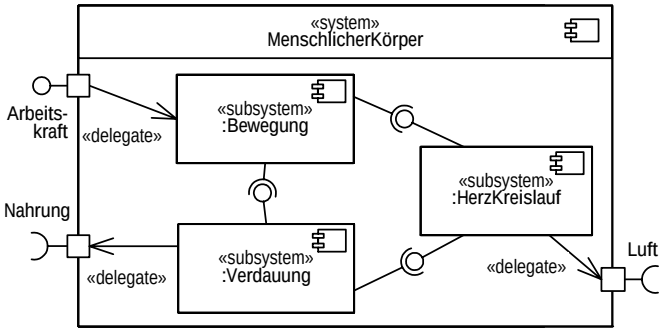
Steht alles im Wiki?

oose. Innovative Informatik

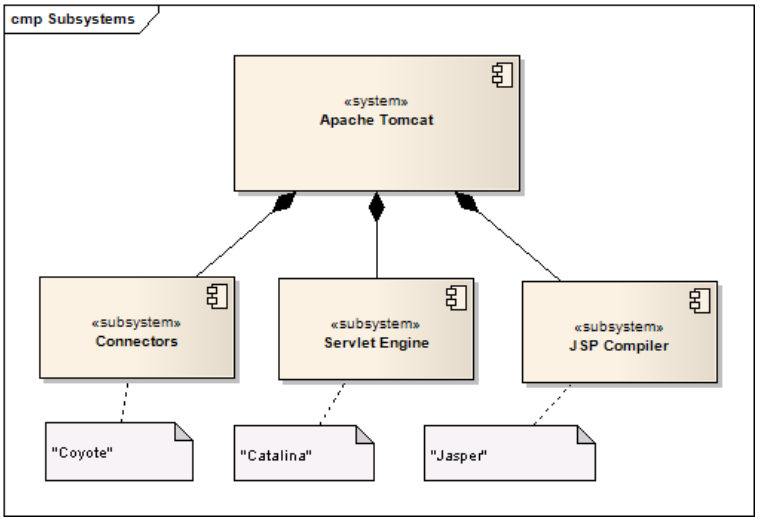
Die Bausteinsicht

„Die Bausteinsicht bildet die Funktionalität des Systems auf Software- oder Implementierungsbausteine ab. Die Sicht macht Struktur und Zusammenhänge zwischen den Bausteinen der Struktur explizit “ (G. Starke)

Beispiel (UML, Kompositionsstrukturdiagramm)

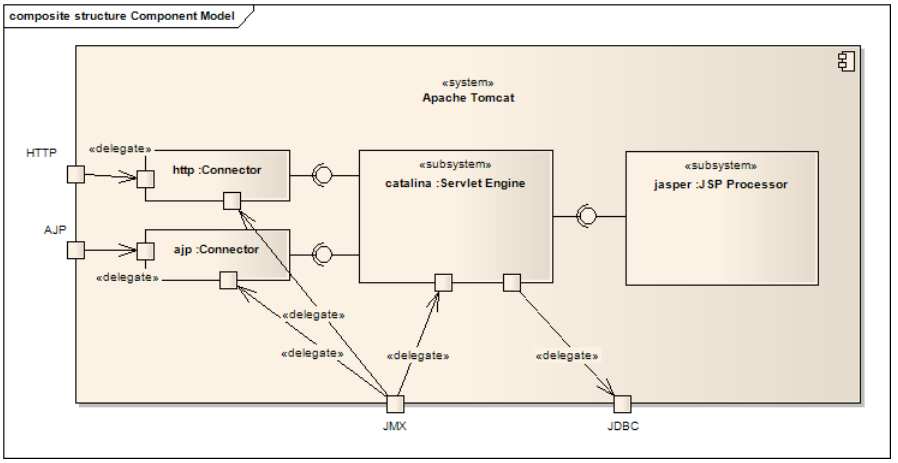


Apache Tomcat: Komponentendiagramm

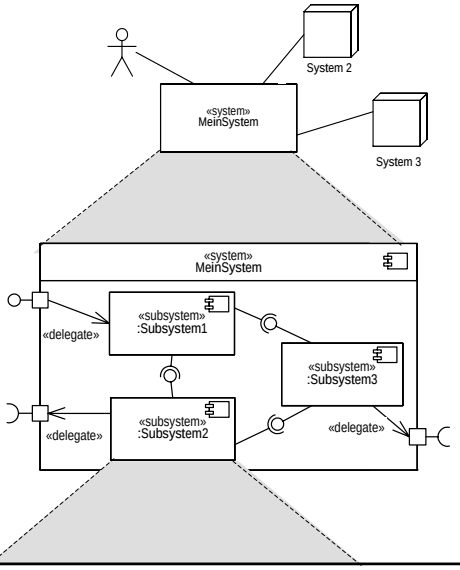


Steht alles im Wiki?
oose. Innovative Informatik

Apache Tomcat: Kompositionsstrukturdiagramm



Zusammenspiel Kontextsicht / Bausteinsicht



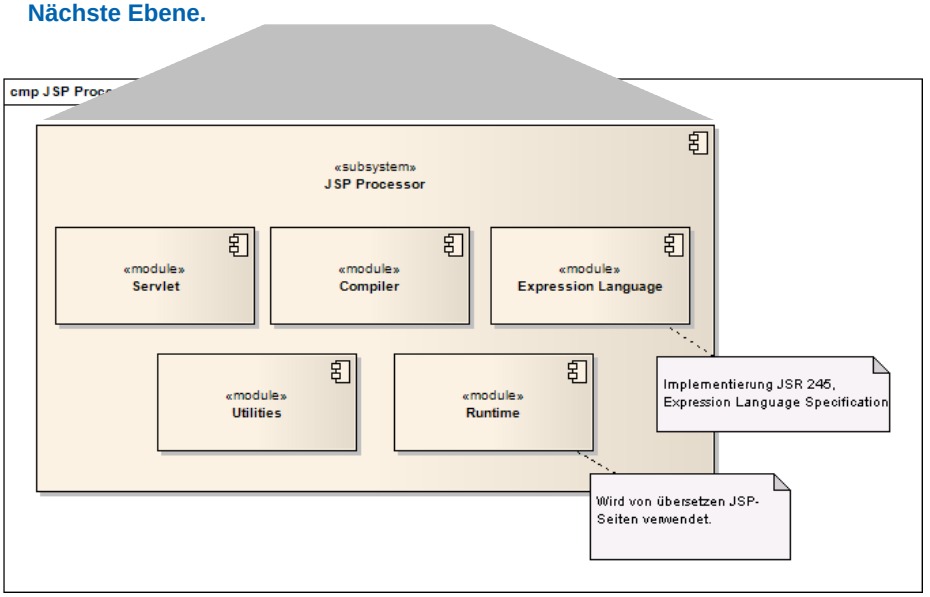
Blackbox

Whitebox

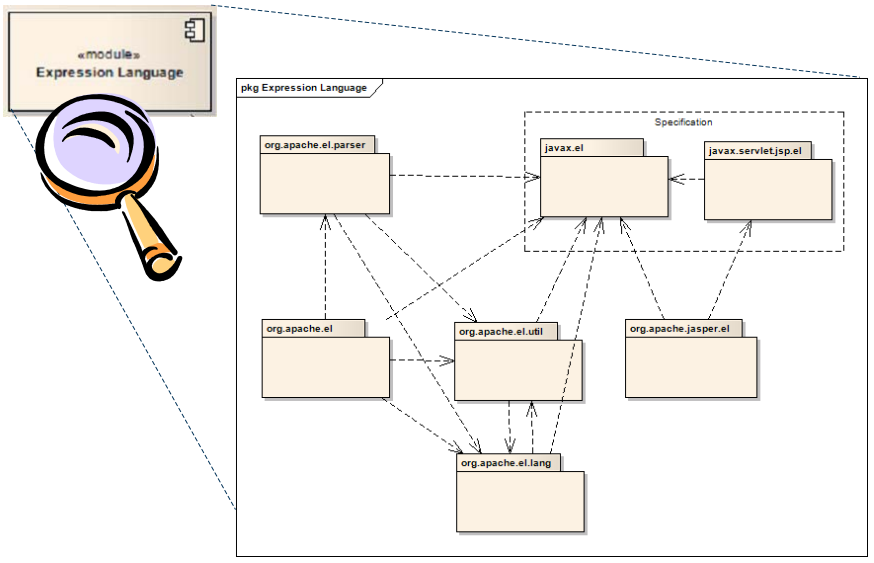
Steht alles im Wiki?

oose.
Innovative Informatik

Nächste Ebene.



Nächste Ebene ...



Die Laufzeitsicht – In Bewegung

- Die Bausteinsicht bietet lediglich eine statische Sicht
- Oft bringt erst die Zusammenschau mit dynamischen Aspekten Einsichten, wie das System eigentlich funktioniert, bzw. zu verwenden oder zu erweitern ist.

Die **Laufzeitsicht** (alternativ: Verhaltenssicht) beschreibt, wie Softwareelemente zur Laufzeit interagieren, bzw. wie ein Element selbst sich verhält.

Laufzeitsicht und UML

- Die UML bietet verschiedene Modellelemente und Diagrammtypen für die Laufzeitsicht an, z.B.
  - Aktivitätsdiagramm
  - Sequenzdiagramm
  - Zustandsdiagramm

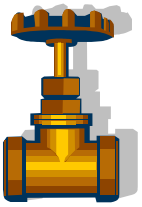
Beispiel: Apache Tomcat

Implementierung einer eigenen Tomcat-Komponente

„Ein Design sollte offen für Erweiterungen, aber geschlossen für Änderungen sein.“ (Open Closed Principle)

Bertrand Meyer 1988

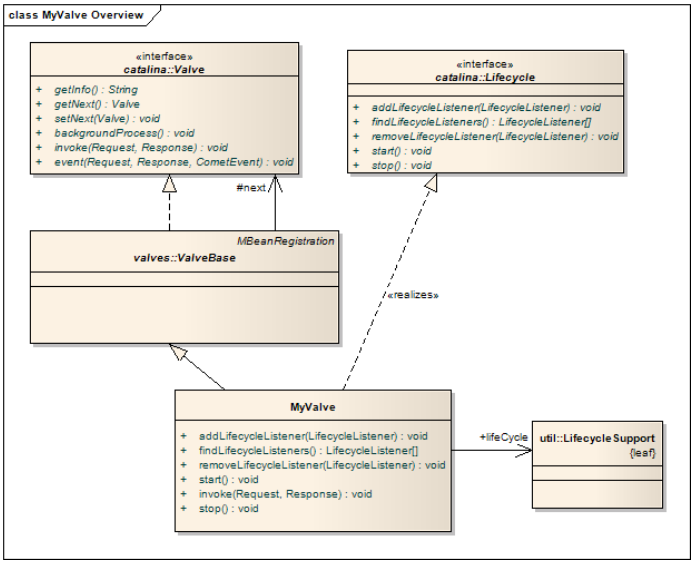
- Tomcat kennt verschiedene Abstraktionen, die gewollte Erweiterungspunkte darstellen (z.B. Connector, Realm)
- Frage: Wie dokumentiert man die Implementierung von Erweiterungen?



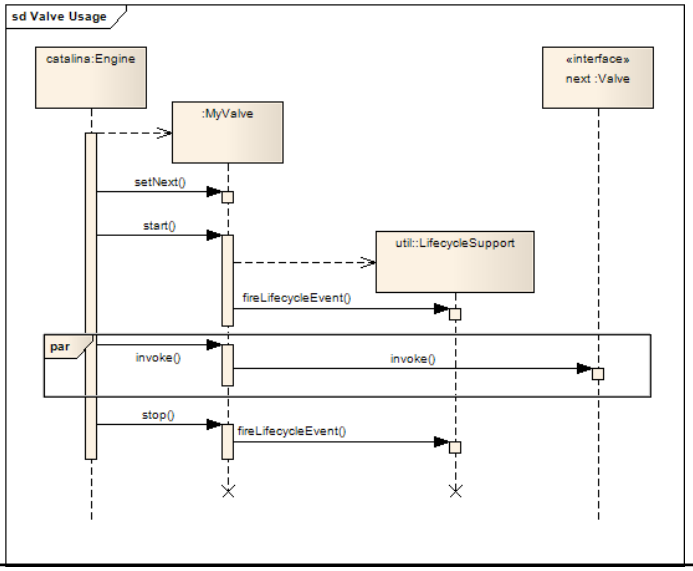
Beispiel: Valve

- Ein Valve (dt. "Ventil") ist eine Anfragen verarbeitende Komponente, die mit einem Container assoziiert ist.
- Üblicherweise bilden eine Kette von Valves eine Pipeline (d.h. ein Valve kennt seinen Nachfolger).

Statische Sicht.



Dynamische Sicht.



Steht alles im Wiki?
oose. Innovative Informatik

Die Verteilungssicht – Ja wo laufen sie denn?

- Die bisherigen Sichten blenden Betriebsaspekte völlig aus.
- Wie verteilt sich die Lösung auf z.B. auf unterschiedliche Rechner?

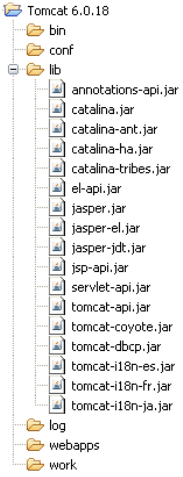
Die **Verteilungssicht** beschreibt, welche physikalischen Informationseinheiten (Jar-Files, DLLs, ...) im Rahmen des Entwicklungsprozesses erstellt bzw. benötigt werden, welche Komponenten sie manifestieren, und wie sie für den Betrieb zu verteilen sind.

Verteilungssicht und UML

- Die UML bietet eigene Modellelemente und ein Diagramm für die Verteilungssicht an
  - Verteilungsdiagramm
  - Knoten, Artefakte

Beispiel: Apache Tomcat

Beispiel 1: Welche Deployment Units brauche ich für was?



Konkret bei Tomcat:

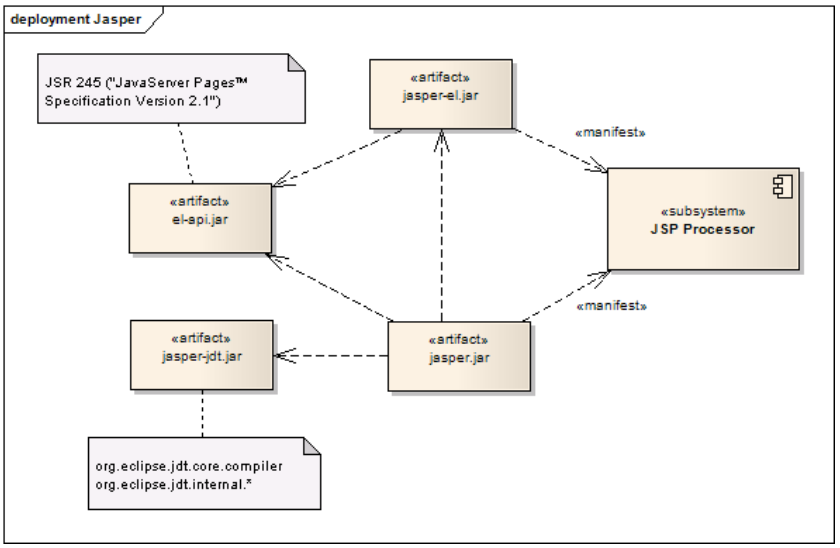
- Tomcat wird in reichlich JAR-Files (Java Archive) ausgeliefert
- Wenn ich nur Teile von Tomcat verwenden möchte (z.B. nur den JSP Compiler), welche JAR-Files werden benötigt?
- Wenn neue Komponenten realisiert werden (z.B. eine Valve), wogegen muss kompiliert werden?

Allgemeine Fragen:

- Welche Deployment Units (JARs, DLLs, SOs, …) manifestieren welche Bausteine?
- Welche Deployment Units müssen wo installiert werden (z.B. bei Client Server)

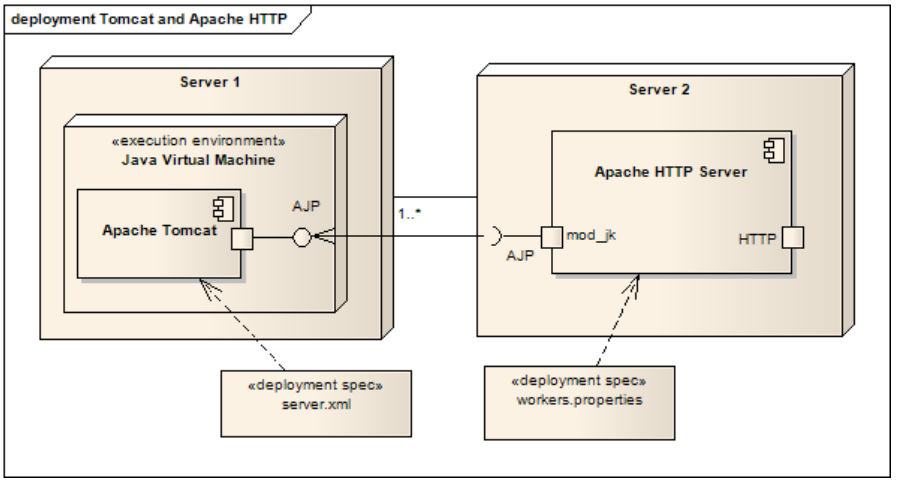
Steht alles im Wiki?
oose. Innovative Informatik

UML Deployment Diagram



UML Deployment Diagram

Beispiel 2: Szenario: Tomcat + Apache HTTP Server



Steht alles im Wiki?
oose. Innovative Informatik

Agenda

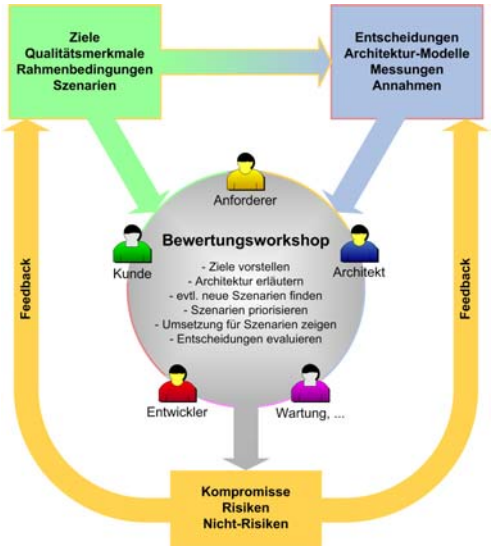
Architekturbewertung: Reflektieren bevor man dokumentiert…

Kommunikation
Meeting
Workshop
Priorisierung
Feedback
Review
Kundenorientierung
Zusammenarbeit
Präsentation
Planung
Risikominderung
Transparenz
Qualitätssicherung
Flexibilität

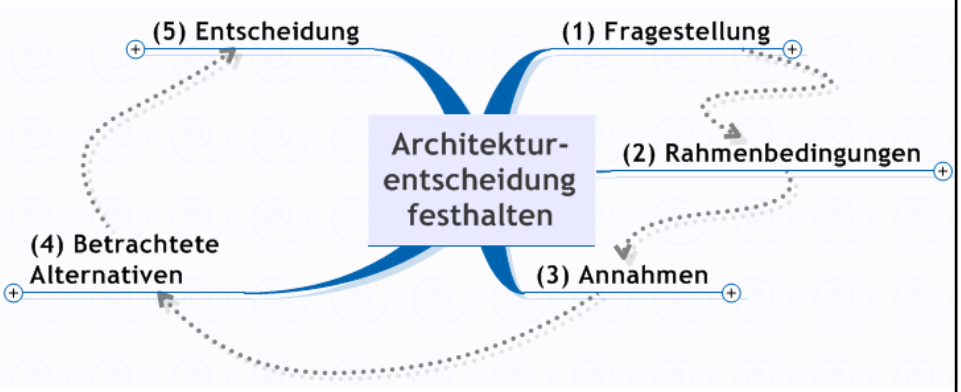
Zentrale Frage: Erreicht man die geforderte Qualitätsmerkmale mit den getroffenen Entscheidungen?

Steht alles im Wiki?
oose. Innovative Informatik

Der Architekturbewertungsprozess

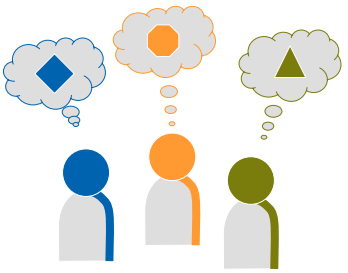


Entscheidungen vor / im / nach dem Workshop festhalten



Steht alles im Wiki?
oose. Innovative Informatik

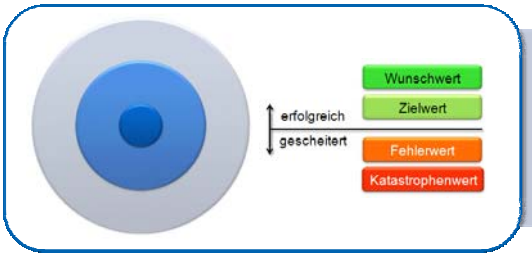
Qualitätsanforderungen dokumentieren



Qualitätsmerkmale für die Umsetzung **spezifischer** zu **formulieren** hilft:

- beim Treffen von Design-Entscheidungen
- bei der Priorisierung der Qualitätsmerkmale
- bei der Bewertung von Entscheidungen

Möglichkeiten Qualitätsanforderungen zu beschreiben



- Strukturierung [Gilb]

- Szenarien

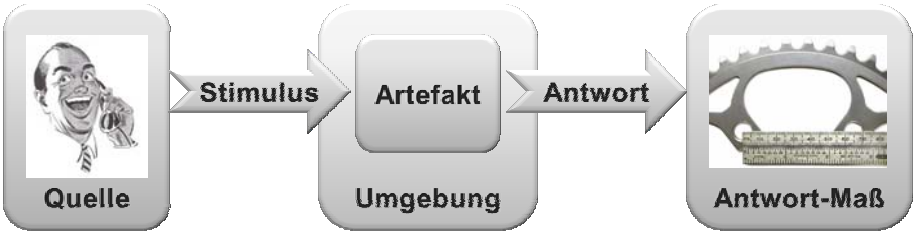
Steht alles im Wiki?
oose. Innovative Informatik

Szenarien

- Ein Szenario ist ein Beispiel für die Verwendung des Systems
- Ein Szenario ist eine „Manifestation“ eines Qualitätsmerkmals
- Ein Szenario ist überprüfbar (zumindest theoretisch)
- Ein Szenario gibt dem „Umsetzer“ Anforderungen auf richtigem Niveau

Ein entfernter Benutzer sendet bei Hochauslastung
eine Anfrage für einen Datenbank-Report über das Internet
und erhält den Report innerhalb von 5 Sekunden.

Mögliche Teile eines Szenarios

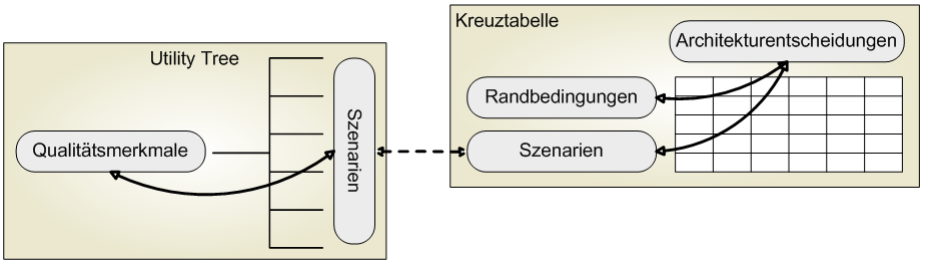


Ein entfernter Benutzer sendet bei Hochauslastung
eine Anfrage für einen Datenbank-Report über das Internet
und erhält den Report innerhalb von 5 Sekunden.

Steht alles im Wiki?
oose. Innovative Informatik

Entscheidungen im Kontext

- Die Verbindung zwischen Rahmenbedingungen, Anforderungen und Architekturentscheidungen ist essenziell!
- Durch die Verbindung können Änderungen
  - in ihren Auswirkungen abgeschätzt werden
  - weniger aufwändig durchgeführt werden (und konsistent erfolgen)



Architekturbewertung – keine Noten aber mehr Durchblick

➔ http://it-republik.de/business-technology/

Steht alles im Wiki?

oose. Innovative Informatik

Agenda

1. Montag Morgen
2. Entscheidungen
3. Sichten
4. Bewerten
5. Strukturieren
6. Schluss und Aus(-blick)

© 2010 by oose GmbH

Steht alles im Wiki?

oose. Innovative Informatik

arc42 – Vorschlag für ein Template

(Gernot Starke, Peter Hruschka)

➔ http://arc42.de/

Steht alles im Wiki?

oose. Innovative Informatik

Struktur des Templates

Steht alles im Wiki?

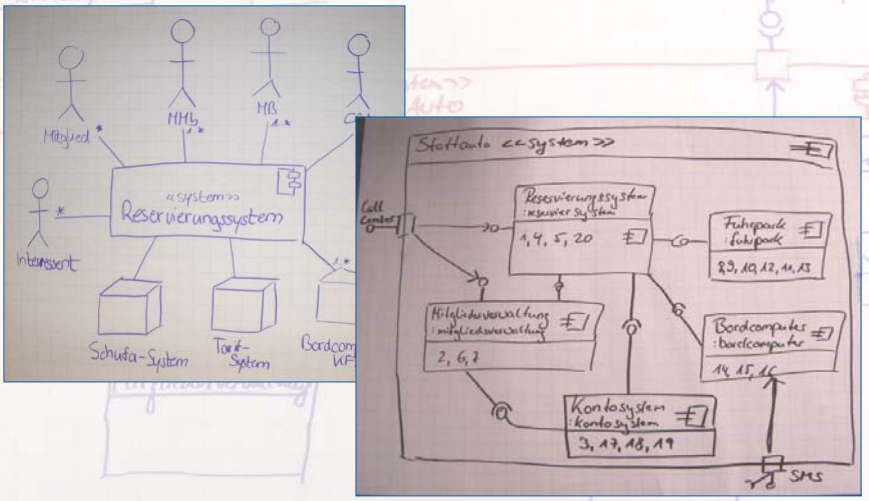
oose.
Innovative Informatik

Es muss nicht immer ein digitales Tool sein ...

Steht alles im Wiki?

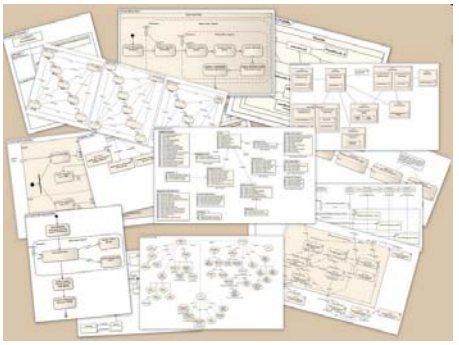
oose.
Innovative Informatik

Übungsergebnisse aus einem oose-Seminar zu Softwarearchitektur



UML = Unified Modeling Language

- etablierte, standardisierte Notation im Bereich Software-Engineering
- Primäre Disziplinen:
  - Analyse
  - Entwurf / Architektur
- umfangreich, 14 Diagrammtypen

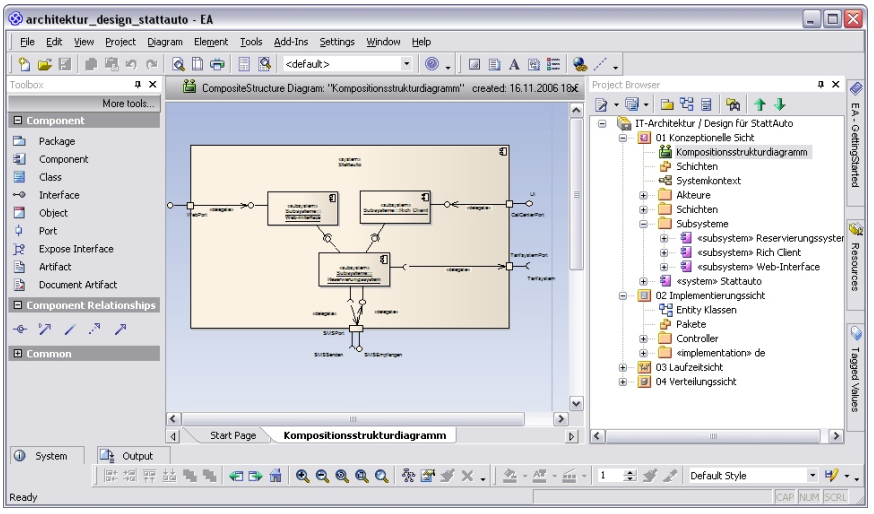


➔ http://www.uml.org/

Steht alles im Wiki?

oose.
Innovative Informatik

Diagramme == Sichten auf ein Modell



© 2010 by oose GmbH

Steht alles im Wiki?

oose. Innovative Informatik

Und im Wiki?

„Wikis ermöglichen das gemeinschaftliche Arbeiten an Texten. Ziel eines Wikis ist es im Allgemeinen, die Erfahrung und den Wissensschatz der Autoren **kollaborativ** auszudrücken."

wikipedia.de

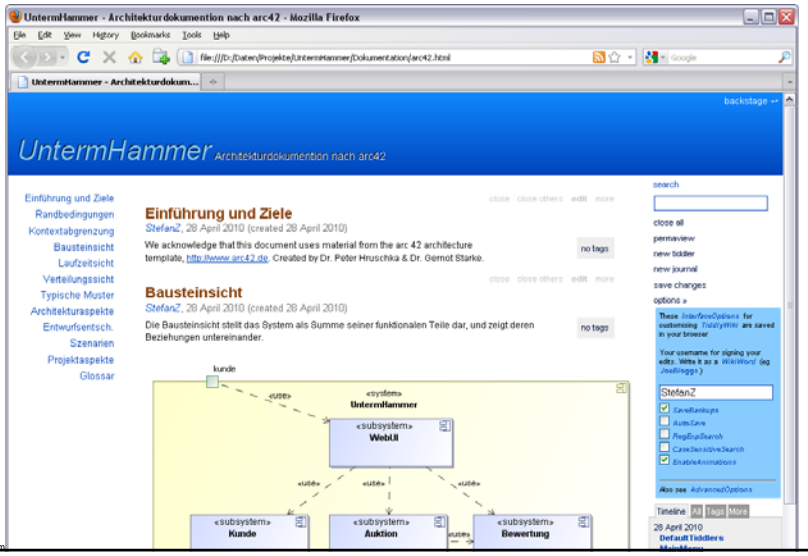
- Nachvollziehbarkeit von Ergänzungen und Änderungen
  - Autor, Historie, …
  - Benachrichtigungen
- Freies Verknüpfen von Inhalten (Links, Tags
- Leicht zugänglich für das ganze Team (kein spezieller Client)
- Lädt zum Kommentieren ein

> *Generell ein tolles Medium für Entwicklungsprojekte, um untereinander zu kommunizieren.*

Steht alles im Wiki?

oose. Innovative Informatik

arc42 in einem Wiki?



Herausforderungen

Diagramme

- Wie erstellt man Abbildungen im bzw. für das Wiki
- Wie hält man Abbildungen und Textinhalte konsistent?

Versionierung

- Wikis führen Versionen für einzelne Seiten
- Wie versioniert man die Dokumentation vollständig, z.B. für ein Release?

Drucken

- Wie gibt man die Dokumentation aus dem Wiki als Dokument (z.B. PDF) heraus?
- Wie befüllt man eine vorgegebene Struktur?

Steht alles im Wiki?
oose. Innovative Informatik

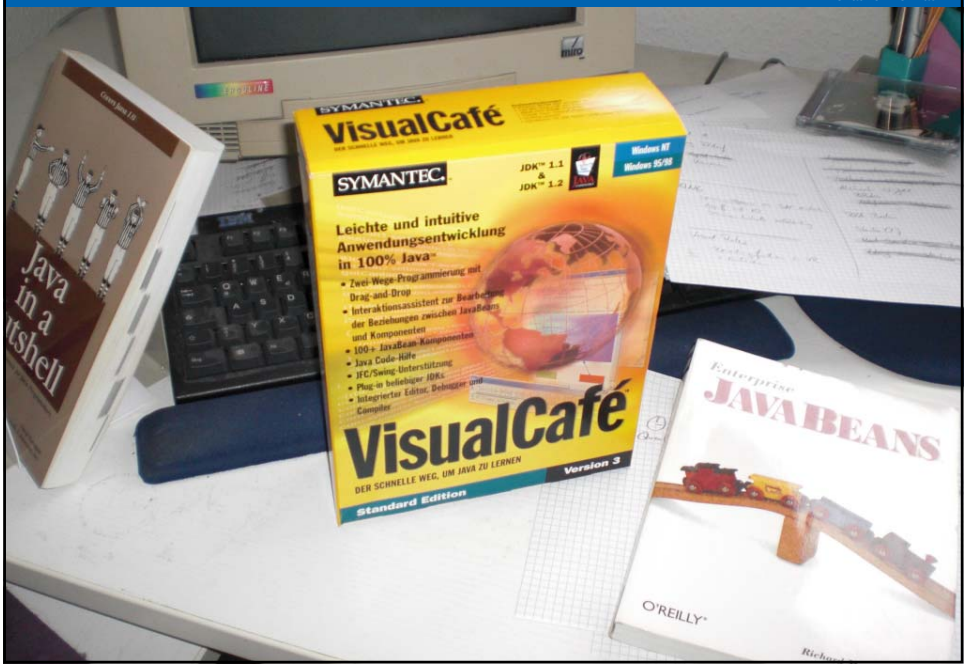
Entscheidungsfaktoren

© 2010 by oose GmbH

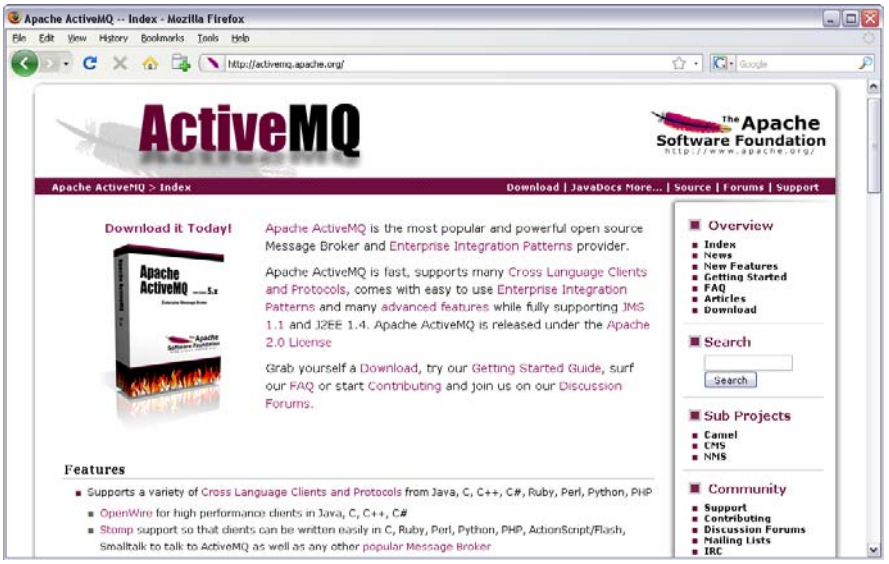
Agenda

1. Montag Morgen
2. Entscheidungen
3. Sichten
4. Bewerten
5. Strukturieren
6. Schluss und Aus(-blick)

Steht alles im Wiki?
oose. Innovative Informatik



Homepage ActiveMQ



Steht alles im Wiki?

oose. Innovative Informatik

Projektteams brauchen ein gemeinsames Ziel, eine Vision

- Was entwickeln wir eigentlich?
- Was ist die Idee des Systems?
- Wem nützt es?
- Wie unterscheidet es sich von Produkten der Mitbewerber?

Es ist eine Ihrer Aufgaben als Softwarearchitekt, die Idee des Systems im Entwicklungsteamteam zu verankern.

Mission Statement

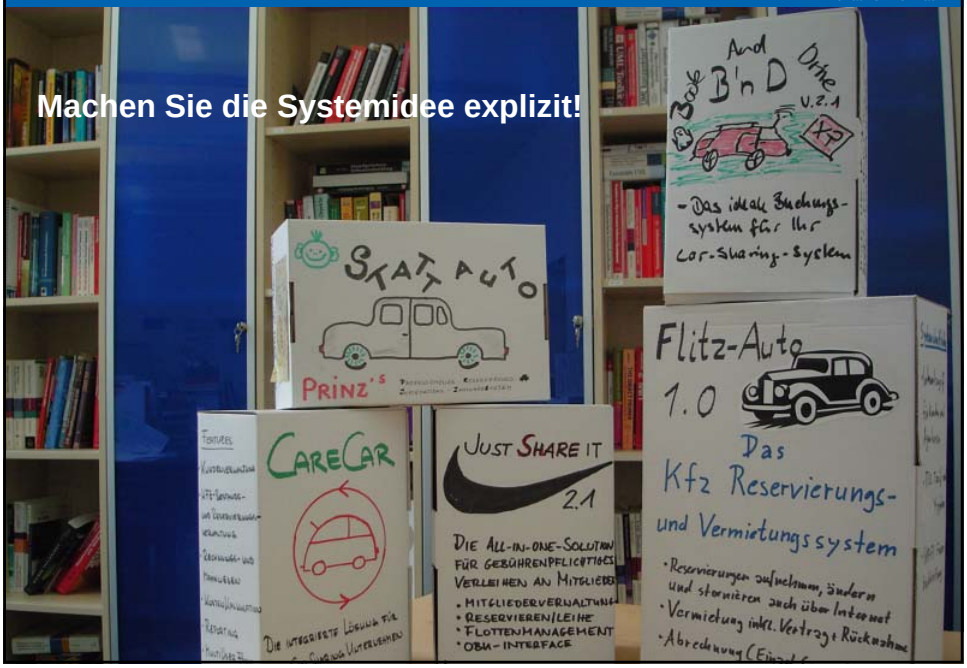
Apache ActiveMQ is the most popular and powerful open source Message Broker and Enterprise Integration Patterns provider.

Apache ActiveMQ is fast, supports many Cross Language Clients and Protocols, comes with easy to use Enterprise Integration Patterns and many advanced features while fully supporting JMS 1.1 and J2EE 1.4. Apache ActiveMQ is released under the Apache 2.0 License

Grab yourself a Download, try our Getting Started Guide, surf our FAQ or start Contributing and join us on our Discussion Forums.

Steht alles im Wiki?

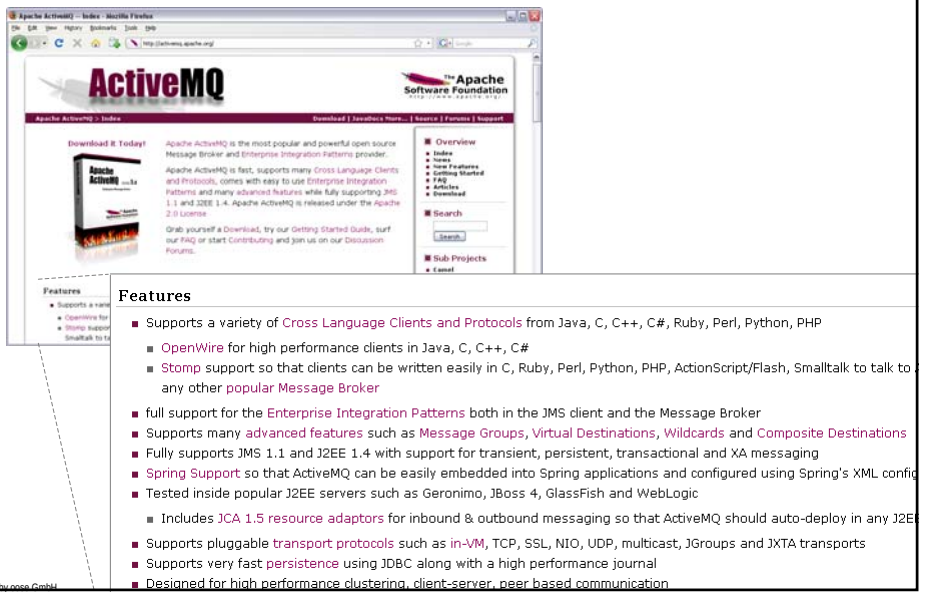
oose.
Innovative Informatik



Steht alles im Wiki?

oose. Innovative Informatik

Systemidee auf der Startseite des Projekt-Wikis …



Steht alles im Wiki?

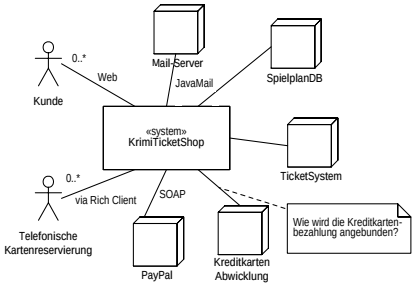
oose. Innovative Informatik

Virtuellen Produktkarton erstellen

z.B.

➔ http://www.wikihow.com/Create-a-Product-Box-in-Photoshop

Systemkontext und Systemidee



Was ist drum herum?

Was steckt drin?

Steht alles im Wiki?

oose.
Innovative Informatik

Kolumne „Architekturen dokumentieren“

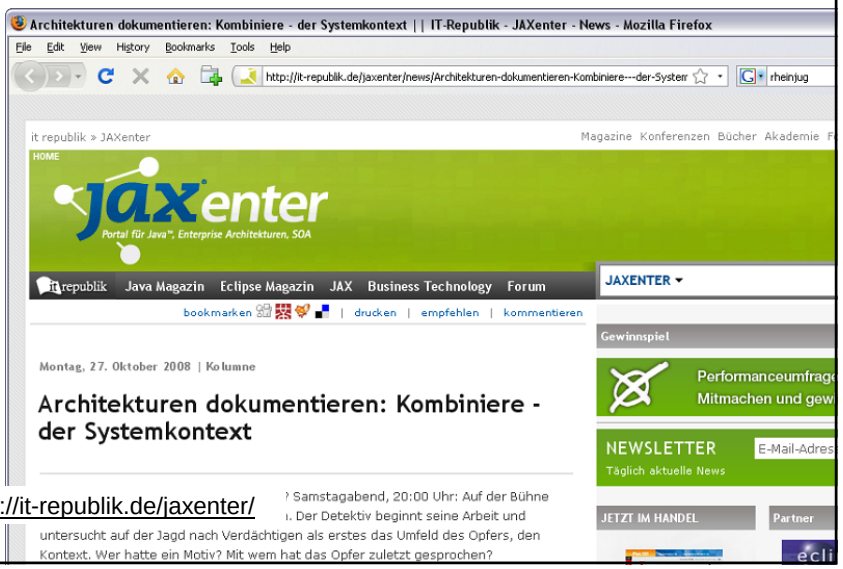
Java Magazin, 10.2008 – 09.2009

➔ http://javamagazin.de/

Steht alles im Wiki?

oose.
Innovative Informatik

Auch im Web



➔ http://it-republik.de/jaxenter/

oose.
Innovative Informatik

Vielen Dank!

Ich freue mich auf Ihre Fragen …

Stefan Zörner :: Stefan.Zoerner@oose.de