

Workshop Visualisierung 2011

Diskussionsnotizen

20.7.2011, Software-Architektur Jahrestagung 2011, Hamburg

Prof. Dr. Florian Matthes
Software Engineering for Business Information Systems (sebis)
Technische Universität München

1) Wie gewinne & definiere ich Architekturbeschreibungen für ein gegebenes System?

- Erwartungen
 - Komplexität für Menschen reduzieren (Entwickler, Betriebspersonal, Management)
 - Kommunikation von Architekt zu Entwickler
- Aktuell benutzte Werkzeuge?
 - Qualität der Dokumentation auf technischer Ebene besser als auf abstrakteren Ebenen (Software, Geschäftsprozesse, Rollen)
 - Idee: Dokumentationstool analog Google Maps (Zoom, Street View, Layer)
 - Ist-Aufnahme:
 - A) PPT, keine Legende, implizite Ontologie für Komponenten und Abhängigkeiten/Beziehungen, ...
 - B) Code-Sicht auf System, Abstraktion aus Code-Mustern (Verzeichnis, Namenskonventionen) auf semantisch „höhere“ (implizite) Konzepte
 - A+B zusammenführen, um Inkonsistenzen zu identifizieren
 - Entwicklung der tatsächlich gewünschten Soll-Architektur
 - Iterieren: Re-Engineering & Architecting
- Architekturverständnis ist Voraussetzung für erfolgreiche Kommunikation

- Visualisierung mit Hilfe von WAM-Konzepten hilft bei der Entwicklung einer gemeinsamen Sprache und eines gemeinsamen Modells in dieser Sprache.
- Stimmt intuitives Architekturmodell der Entwickler mit dem konzeptuellen „akademischen“ Architekturmodell überein?
- Orientierung an der bereits etablierten Sprache im Unternehmen, oder an Sprache der „Wissenschaft“?
- Beispiel Bombardier, LH, ... : Arbeitsteilung in der Systementwicklung ist wg. fehlender gemeinsamer Sprache problematisch.
 - Kontext V-Modell
 - Es werden mehrere Formalismen verwendet (SysML, TRAC), und spezielle Werkzeuge sichern die Konsistenz zwischen den Modellen in diesen Formalismen: Match, Lieber Lieber (Österreichisches Unternehmen).
 - Verlinkung mit Req-Eng. Artefakten in Rational System Architect erfolgt manuell und Konsistenz wird partiell durch Werkzeuge geprüft.
 - Governance: System-Architekten und Subsystem-Architekten
- Erfahrungswissen über Personen von Projekt zu Projekt weitergegeben. Schlüsselrollen (Architekt, Teamleiter)
- Unterschiede bei Reifegrad des Architekturverständnisses im Bereich Eingebettete Software vs. Informationssysteme?

- Erfahrungen bezüglich der Größe von Architekturmodellen? Anzahl der Elemente und Anzahl der Beziehungen? „Zoom“-Stufen?
- Bauvorschriften sind wie Gesetzestexte in Textform vorgegeben und können nicht direkt in Kartendarstellungen übernommen werden.
- Entwicklerrichtlinien detaillieren Bauvorschriften und machen diese handhabbar / überprüfbar.
- Ein zentrales Architekturmodell mit zahlreichen generierten und änderbaren Sichten (Views, Viewpoints) für verschiedene Stakeholder ist ein Idealzustand, der in der Praxis mit den heute verfügbaren Werkzeugen und Prozessen unerreichbar ist. → kein konsistenter MDA möglich.
- Stadtmetapher basiert auf stabiler (realer) Verortung, die bei Software nur „mental“ (und subjektiv?) existiert. Wie sehr dürfen diese Sichten voneinander abweichen?
- Entwicklung eines stabilen Bilds als Architekturrahmen dauert sehr lange und erfordert einen hohen Reifegrad (Bsp. Bombardier). Erfordert separaten Prozess und hohe Vertrautheit mit Fachvokabular, Anwendungsdomäne.
- Wie soll eine Visualisierung mit der Verschiedenartigkeit der Implementierungssprachen umgehen (Polymorphie, Higher Order Functions)? Sind geographische Metaphern immer geeignet? Was ist die richtige Granularität?
- Konkret: Welche Fragestellungen können besonders gut mit Visualisierungen gemäß der Städtemetapher beantwortet werden? Paper von Gail Murphy mit Katalog von 70 typischen Fragen, die Entwickler stellen.
- Weitere Beiträge (Tools, Erfahrungen, Ideen) und Wünsche (Veranstaltungen, ...) per E-Mail an matthes@in.tum.de oder andere Referenten

- Aus welchen Abstraktionsebenen?
- Mit welchen Formalismen?
- Welche Rolle spielt hier die Visualisierung?

2) Wie erkenne ich relevante Architekturbeschreibungen und wie erkenne ich deren Verletzung bei der Implementierung (Code)? => Visualisierung!

3) Wie reagiere ich auf Architekturverletzungen? => Visualisierung!

- Änderung des Code
- Änderung der Architektur
- Priorisierung, Agilität