

München, 13. April 2010

# **Requirements Engineering – unausgesprochene (?) Probleme, pragmatische Lösungen**

Dr. Markus Pizka

**1** itestra GmbH

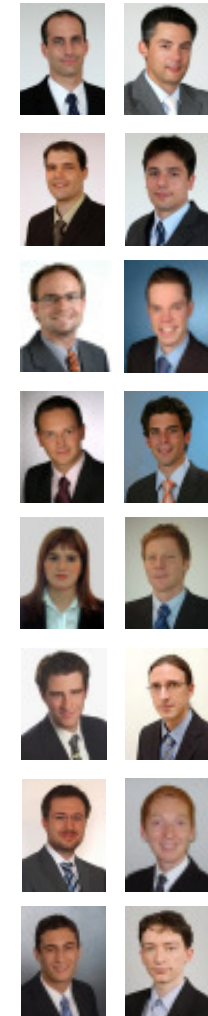
**2** Was ist besonders an RE in/für KMUs?

**3** Reale Herausforderungen – Erfahrungen / Anekdoten\*

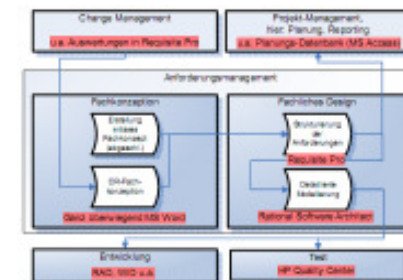
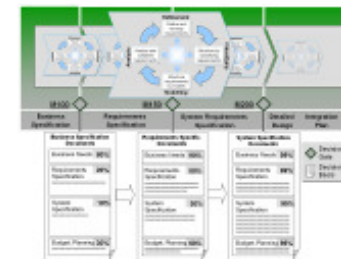
**4** Pragmatische Lösung

*\*Aus drei Anekdoten ist es möglich, das Bild eines Menschen zu geben [Nietzsche]*

- 15 Jahre Forschung
  - Verteilte Systeme, High Performance Computing
  - Software (Re-)Engineering
  - 80+ internationale Publikationen
  - QuaMoCo Initiator und Projektpartner
- Informationstechnologie und -strategie GmbH
  - seit 2004
  - 30% - 40% Wachstum p.a.
  - München, Tallinn (2008), San Francisco (2009)
- Kernkompetenzen
  - Informatik, Software-Ökonomie, (Re-)Engineering, Prozessverbesserung
  - Software-Performance, Software-Architektur
- Leistungen
  - Solution Engineering
  - Software Governance



- RE ist unbestritten von zentraler Bedeutung!
- HSE 2009
  - *RE Reference Model*: Modellbasiertes RE, Prozess, **Formales CM**, Domänenspezifische Artefaktmodelle, **Innov. Tools**
  - *RE – von der Kunst zu spezifizieren*: Musterformulierungen, **RE-Abteilung**, enger Kontakt, **Knowledge Management**
  - *Ableitung von Lösungen aus Anforderungen*
  - *Anforderungsmanagement ... Fahrwerksregelsysteme*
  - *Anf.management in einem industriellen Großprojekt*: Rev. Eng. Fachkonzept 6000S., **IBM Tools**, Kunde def. Vorgehen
- **Summerville – KMU Spezifika?**
  - 5 – Softwareanforderungen (fkt. / n.-fkt)
  - 6 – Abläufe (Bestimmung **gemeinsam mit Kunde**, Verwaltung)
  - 7 – Modelle ... Prototypen ... **formale Spezifikation**



- Intern: Low(er)-Budget

- keine teuren Werkzeuge
- keine RE-Abteilung, i.d.R. auch kein Analysten-Team
- keine umfangreichen sondern einfache / effiziente Prozesse

⇒ **Pragmatische Lösungen statt teure Heilsbringer**

- Extern / als Lieferant: hoher Erfolgsdruck

- keine Großprojekte mit hunderten PT und hoher Anzahl PT für RE
- hoher Aufwand wird als Scheitern wahrgenommen,  
(bei Großlieferanten erwartungskonform; | Lieferant | ~ | Seiten | ~ Kosten?)
- muss durch Leistung überzeugen  
(weniger Möglichkeiten durch Vertrieb, Key Account Mgmt., ...)
- Scheitern hätte katastrophale Konsequenzen

⇒ **Besonderer Druck zu Effizienz und Ergebnisqualität**

Adressieren Kernherausforderungen  
ohne hin unzureichend

# H1: Ausrichtung der Organisation

- Fertiges Anforderungsdokument mit 20 MB / 450 Seiten als „nicht geeignet für Implementierung“ verworfen
- Projekt nach 1 Jahr mit 18 MA gestoppt, Ergebnis: ca. 150 Dokumente, Software nicht ausführbar
- Anforderungsdokument mit >1.000 Seiten
- Öffnen des Anforderungsdokuments benötigt 20 Minuten
- Stundenlange (=kostspielige) Abstimmmeetings mit zahlreichen Teilnehmern ohne Agenda & Protokoll

⇒ **Vollständigkeit** und **Korrektheit**  
**nicht** die dringlichsten Probleme!  
Ausrichtung auf Wirtschaftlichkeit?



# H1 – Fehlende Ausrichtung auf Wertschöpfung



- Meetingzeit, prosaischer Umfang, Perfektionieren von Grafiken wird als „Arbeit“ empfunden und mit Leistung gleichgesetzt!
- Angst ... vor Diskussion, Fehler u.a.
  - Erheblich zu große Projektteams
  - Unklare Verantwortlichkeiten
  - Wartezeiten
  - Bsp.: - „Es darf nichts im Konzept enthalten sein, das nach Geschäftsprozess aussieht, denn über den sind sich die Abteilungen noch nicht einig“
    - Reviewer-Liste mit 50+ Personen
  - Ursache: Unternehmenskultur
- Unkoordiniertes Vorgehen
  - Bsp.: zeitgleiche Beauftragung Anf.-Analyse & Implementierung ohne Interaktion

- Abteilungsgrenzen verhindern Abstimmung z.B. zwischen IT/Fachbereich
    - Verantwortung läuft u.U. erst bei Vorstand zusammen
  - Abteilungsrivalitäten auf dem Rücken des Projekts/Lieferanten ausgetragen
  - Angst um eigene Features bzw. mögliche Budgetkürzungen:
    - möglichst viele Features frühzeitig & detailliert spezifizieren /  
möglichst viel in aktueller Leistungsstufe umsetzen, **obwohl Bedarf unklar**
      - Beispiel: Benutzerdefinierte Attribute für Datenobjekt gefordert  
(da Bedarf an weiteren Attributen vermutet aber noch nicht genau bekannt)
    - nach Genehmigung Budget i.d.R. **kein Anreiz, auf Features zu verzichten**
- ⇒ konterkariert iteratives Vorgehen
- Trotz Reviews, QA-Rolle etc. meist keine wirksame Qualitätssicherung  
(lediglich Prüfung, ob eigene Anforderungen enthalten sind)

## H3 – Rollenzuteilung

- Informatiker *reden* und *schreiben* i.d.R.
  - nicht gerne und
  - auch nicht gut.
  - ⇒ RE-Dokumentation ist unbeliebt und wird gerne „Fachexperten“ überlassen
- Mitarbeiter aus Fachbereichen (BWL, Medizin, Grafik, ...) sind für RE nicht ausgebildet und hierfür ggf. nicht die richtigen
  - Kennen Methoden nicht (UML, Use Cases, ...)
  - Ungeübt in den für Systementwicklung relevanten Disziplinen
    - Präzision
    - Automatisierung
    - Abstraktion und Modellbildung
    - Technische Ressourcen

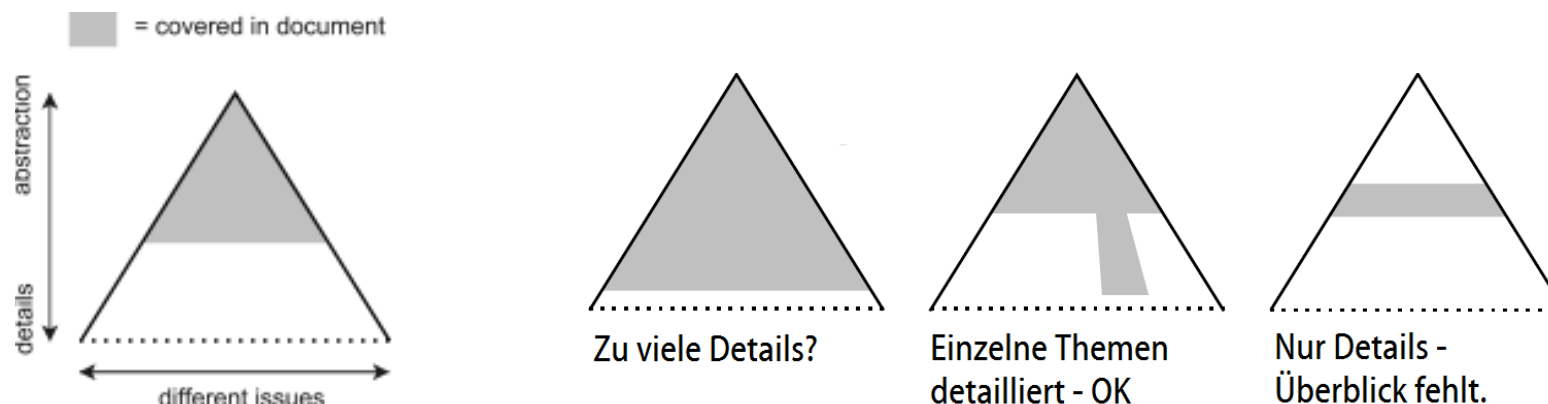
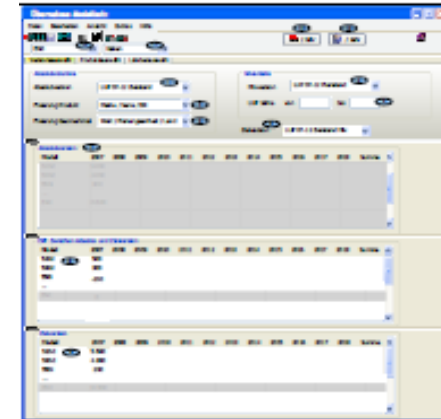
Auch durch Schulung i.d.R.  
kurz-/mittelfristig nicht veränderlich

## H4 – Mangelnde Fachkompetenz

- Annahme diverser Vorgehen: gemeinsame Erarbeitung der Anforderungen, d.h.
  - Anwender kennen die Anforderungen an die Software nicht vollständig, aber
  - Anwender (Fachbereich) kennen die Fachlichkeit.
- Realität: Fachbereich kennt häufig die fachlichen Anforderungen noch nicht!
  - Der Fachbereich lebt in Ist-Abläufen, nicht in Soll und ... hat nicht die Kernaufgabe, das Soll zu definieren.
  - *Produktkäufe werden in erster Linie angestrebt um fachliche Lösungen zu kaufen*
  - *Falsche Fehlermeldungen in Produktion sind Konsequenzen dieses Irrtums*
- Beispiele:
  - Von Fachexperten definierte Formel für Ausfallrisiko liefert Wahrscheinlichkeit für Risikoeintritt von -20%.
  - Bei der Realisierung eines Systems zur Bewertung von Kreditrisiken wird erstmals diskutiert, wie Sicherheiten im Detail zu verrechnen sind.

# H5 – Mangelnde Konsequenz

- Häufig: Abstraktionsebene zu niedrig oder heterogen; Beschreibung von Lösungen statt Anforderungen
  - z.B. aufwändige Dialogzeichnungen aber auch Code in Word (während Ziele und Geschäftsprozesse fehlen)
  - Design / Implementierung unnötig eingeschränkt, häufig auch fachlich keine gute Lösung
  - Bläht Dokument auf ⇒ teuer in der Erstellung und Pflege
  - Täuscht reifes Stadium der Anforderung vor



aus: Practically Relevant Quality Criteria for Requirements Documents. Simon et al., SERP 2008

# H7 – Falsche Anwendung von Methoden

- Trivialitäten ausführlich beschrieben, Beispiel: Ablauf behandelt i.W. Abbruch:
- Leere oder wiederholt identische Vor-/Nachbedingungen, Akteure, querschnittliche Anforderungen

<b>Vorbedingungen:</b>	
<ul style="list-style-type: none"> <li>Eine Referenz auf ein Objekt des Basisprozesses existiert.</li> </ul>	
<b>Standardablauf:</b>	
Aktivität	Systemantwort
1. Löschen einer Referenz	System löscht die Referenz.
<b>Varianten:</b>	
<b>Nachbedingungen:</b> Referenz ist gelöscht.	
<b>Nichtfunktionale Anforderungen: (optional)</b>	
<b>Quelle: (optional)</b>	
<b>Priorität: (optional)</b> 1	
<b>Schwierigkeitsgrad: (optional)</b>	
<b>Offene Punkte: (optional)</b>	

Use Case: Editing a data item		
1.	The actor chooses one data item from the list.	The system selects the item.
2.	The actor chooses "Edit item".	The system opens a dialog for editing the item.
2a.	The actor enters values and confirms.	The system stores the entered values.
2b.	The actor enters values but does not confirm.	The system asks back whether the changes shall be abandoned.
2ba.	The actor confirms and thus approves storage of the changes.	The system stores the entered values.
2bb.	The actor negates and thus disapproves storage of the changes.	The system discards the changes and does not store the entered values.

- Methoden / Tools helfen nicht, wenn sie nicht verstanden werden.

Untersuchung itestra und TUM:

- Durchschnittlich **jeder achte Satz** eines Requirements Dokuments ist Bestandteil eines Clones mit **min. 20 identischen Wörtern**
  - Extremfall: 65% des Dokuments gecloned
- Beispiel: Redundante Use Case Beschreibungen
  - Use Cases *Add & Edit* für ein Datenobjekt nahezu identisch
  - Identische Use Cases für jedes Datenobjekt in voller Länge beschrieben
  - Identische Vor-/Nachbedingungen, querschnittliche Anforderungen, ...

The Curse of Copy & Paste – Cloning in Requirements Specifications. Domann et al., ESEM 2009  
Can Clone Detection Support QA of Requirements Specifications? Jürgens et al., ICSE 2010

## H9 – Ignoranz / Bsp. Versionierung

- Unzureichende Versionsverwaltung
  - über Dateinamen, z.B. **Fachkonzept\_v19\_20.03.doc**
  - über Farben
  - parallele Versionen durch Versand per E-Mail
  - Quellen für eingebundene Grafiken nicht mehr auffindbar

Version	Stand	Autor	Änderungen
1.0	25.10.2008		Erstellung nach Besprechung
1.1	07.03.2009		Beantwortung der Fragen (ToDo) (blaue Schrift)
1.2	23.03.2009		Geringfügige Ergänzungen (grüne Schrift)
1.3	30.03.2009		Geringfügige Abweichung bei XYZ (violette Schrift)
1.4	11.05.2009		Rückmeldung aus Erstellung Testfälle (orange Schrift)
1.5	29.05.2009		Ergänzungen (violette Schrift + gelber Hintergrund)

- Warum wird KM für Dokumente nach wie vor i.W. ignoriert?

*Herausforderung: Ausrichtung, Politik, Rollenbild, Fehlannahmen, Konsequenz, Letargie, Ignoranz*

Requirements Engineering ist eine wichtige Führungsaufgabe  
und muss als solche verstanden und gelebt werden!

- Führen heißt: informieren, entscheiden, ggf. durchsetzen, verantworten.
  - Führen erfordert: (genau 1) starke Persönlichkeit(en)
    - Methodenkompetenz ✓
    - Fachkompetenz ✓
    - Initiative – wenig (sinnvolles) geschieht ohne eigene Aktivität
    - Wille zur Perfektion
    - Mut und Übernahme der Verantwortung
    - Kommunikationsfähigkeit und Durchsetzungsfähigkeit
- } in der Regel weder  
explizit betrachtet noch  
gezielt geschult

# Ausbildung der gesuchten „Führungskräfte“

- Auswahl:
  - Profil Extrovert\*
  - Math. Denkstrukturen (Ziel: Systementwicklung)
- Ausbildung:
  - Aufklärung über die Tätigkeit, realistische Erwartungen.
  - Vermittlung von Methoden Know-How
  - Fordern / Übergabe von Verantwortung
  - Mentoring
    - Offene Diskussion der nicht-technischen Fähigkeiten
    - Intensives Coaching
    - Konsequente QS der Ergebnisse

... auch in KMUs leistbar

*\*gesprächig, bestimmt, aktiv, energisch, dominant, enthusiastisch und abenteuerlustig.*

**Bewährte** Methodiken und Tools sind ohne hohen Aufwand einsetzbar!

- Einsatz geeigneter Templates (inkl. Formatvorlagen)  
(nicht jedesmal neu erfinden, trotzdem notwendige Flexibilität zulassen)
- UML, Use Cases, Geschäftsprozessdiagramme  
(ggf. frei verfügbare Werkzeuge)
- Tabellen, Diagramme, Listen statt Prosa
- Konstruktive QS: Redundanzen, unnötiges, etc. radikal entfernen / ändern
- Zentrale Versionsverwaltung (CVS, SVN) für alle Artefakte wirkt Wunder



**HABEN SIE FRAGEN?**