



Produktlinien-bewusste Anforderungserhebung durch maßgeschneiderte Erhebungsprozesse

Sebastian Adam

Hamburg, GI Fachgruppentreffen, 25.11.2011

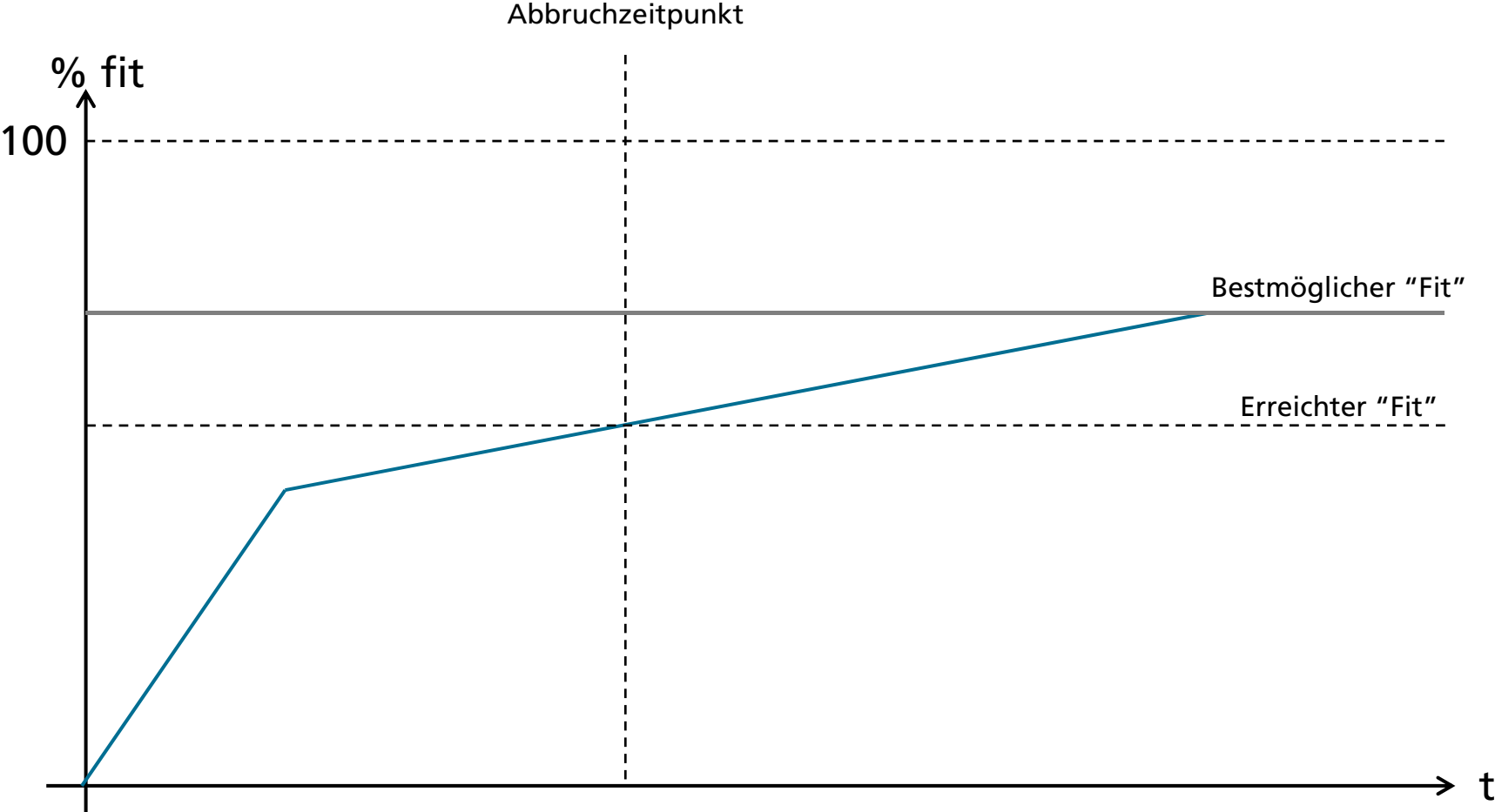
Kontext

- **Softwareproduktlinien (SPL)** gelten als der systematischste Ansatz um die Softwareentwicklung durch **strategische Wiederverwendung** effizienter zu gestalten
- **Customizable Information Systems** können als spezielle Form von Softwareproduktlinien (SPL) angesehen werden
- **Beispiele:** Geschäftsprozessanwendung, Dokumentenanalyzesystem, ...
- Wichtige Anforderungen lassen sich hier **nicht immer explizit antizipieren** (offene Variabilität) **35%**
 - explicitly anticipated requirements (commonalities, variants), implicitly anticipated requirements, non-anticipated requirements

Problem (1)

- Auswirkungen „offener“ Anforderungen auf die Architektur und Systementwicklung zu einem frühen Zeitpunkt **schwer abschätzbar**
 - Nicht-passenden Anforderungen werden erhoben, welche aus Gründen **technischer oder wirtschaftlicher Machbarkeit** spät **nachverhandelt** werden müssen (oder das Gegenteil!) **35%**
 - Den bestmöglichen „Fit“ zwischen Anforderungen und wirtschaftlich vertretbaren Möglichkeiten zu erreichen **dauert entweder zu lange oder funktioniert überhaupt nicht**
 - **Gesamteffizienz leidet**
 - entweder geringe Wiederverwendung oder lange Nachverhandlungen
-

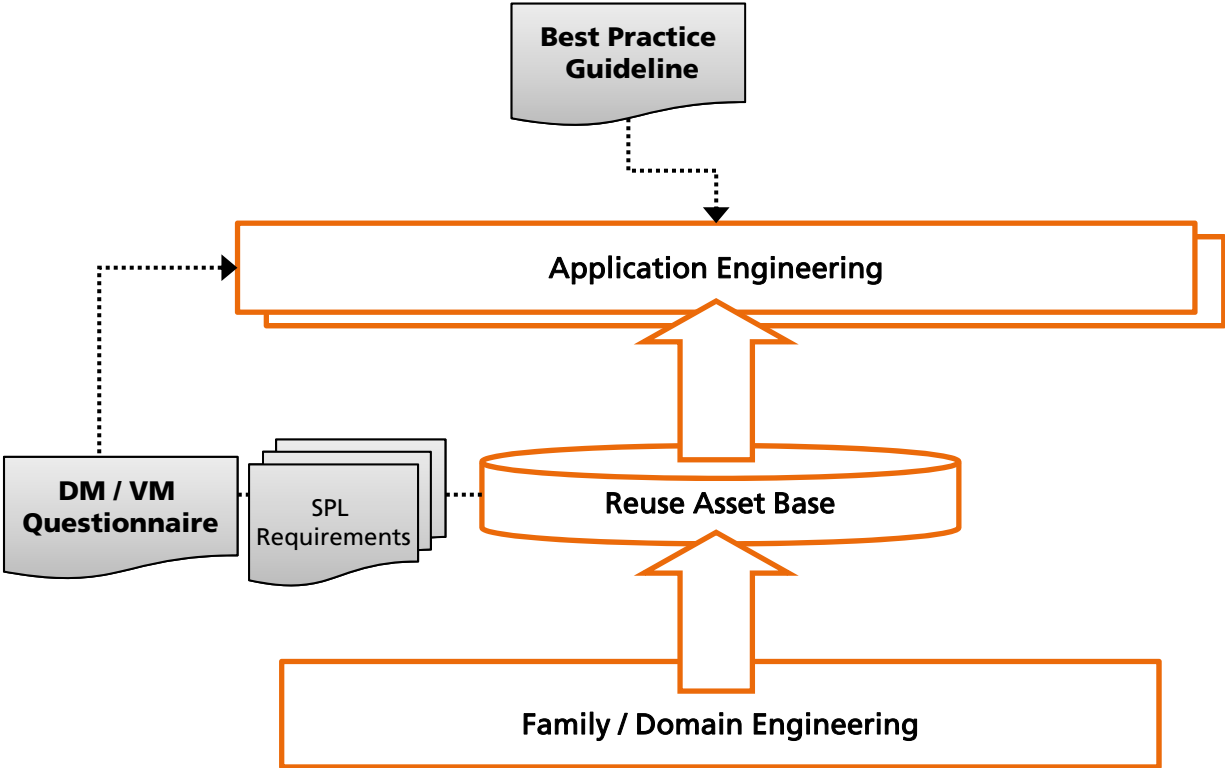
Problem (2)



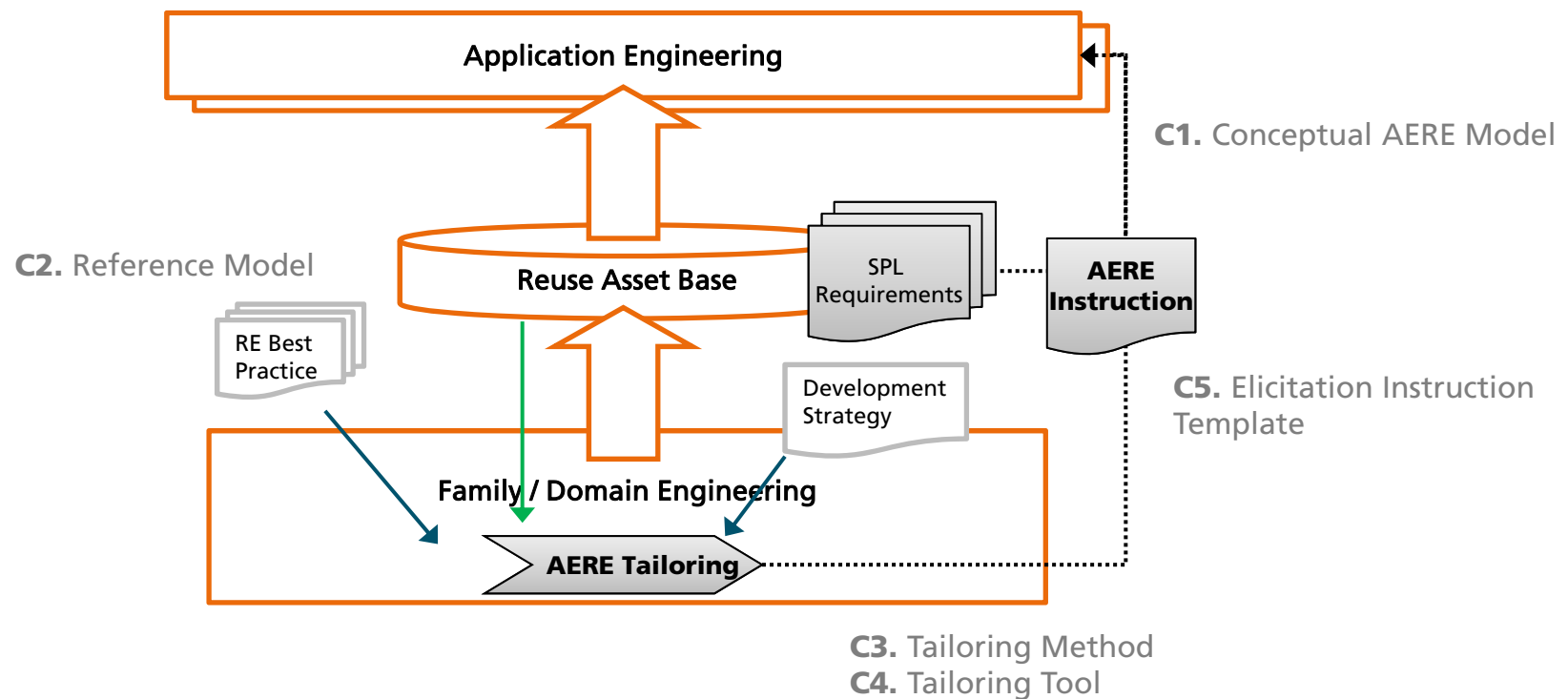
Stand der Technik (1)

- Derzeit **keine Ansätze**, welche Anforderungsingenieure während der Erhebung von nicht-explicit antizipierten Anforderungen unterstützen
- **Effektivität und Effizienz** hängt daher oftmals davon ab
 - welche **persönliche Kenntnis** der durchführende Anforderungsingenieur über die Produktlinie hat
 - zu welchem Grad sich Kundenanforderungen vorab explizit antizipieren und beschreiben lassen
- Und:
 - Bestehende Ansätze adressieren **unzureichend** wenn schon etwas „Gewachsenes“ da ist

Stand der Technik (2)



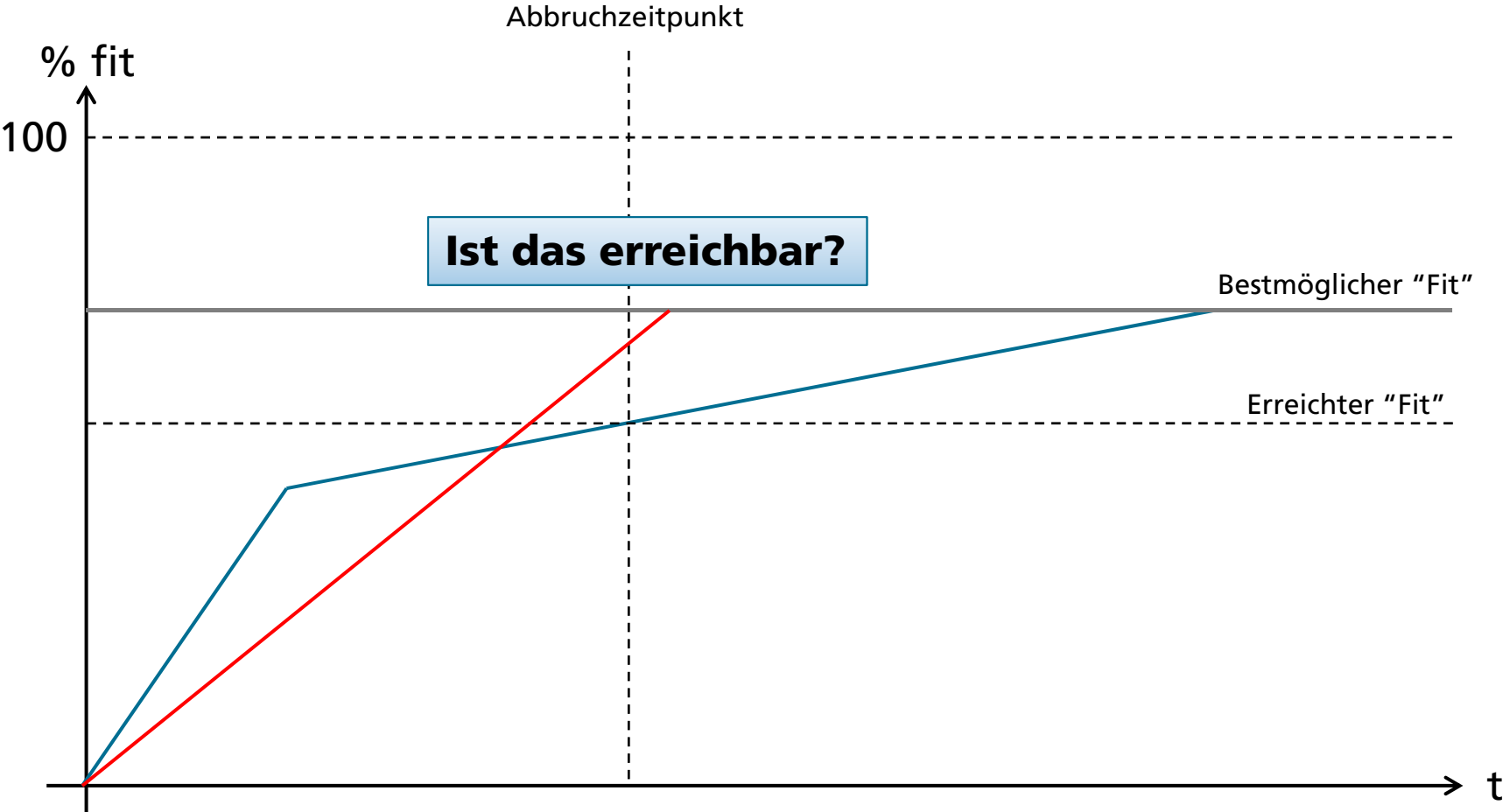
Lösungsidee und -komponenten



Zielsetzung (1)

- Anforderungsingenieure können eine **effektivere Erhebung durchführen** und besser verhandeln
- „Böse Überraschung“ und **kostspielige Änderungen erst spät im Kundenprojekt können dadurch minimiert werden**, was zur Gesamteffizienz beiträgt

Zielsetzung (2)



Evaluation

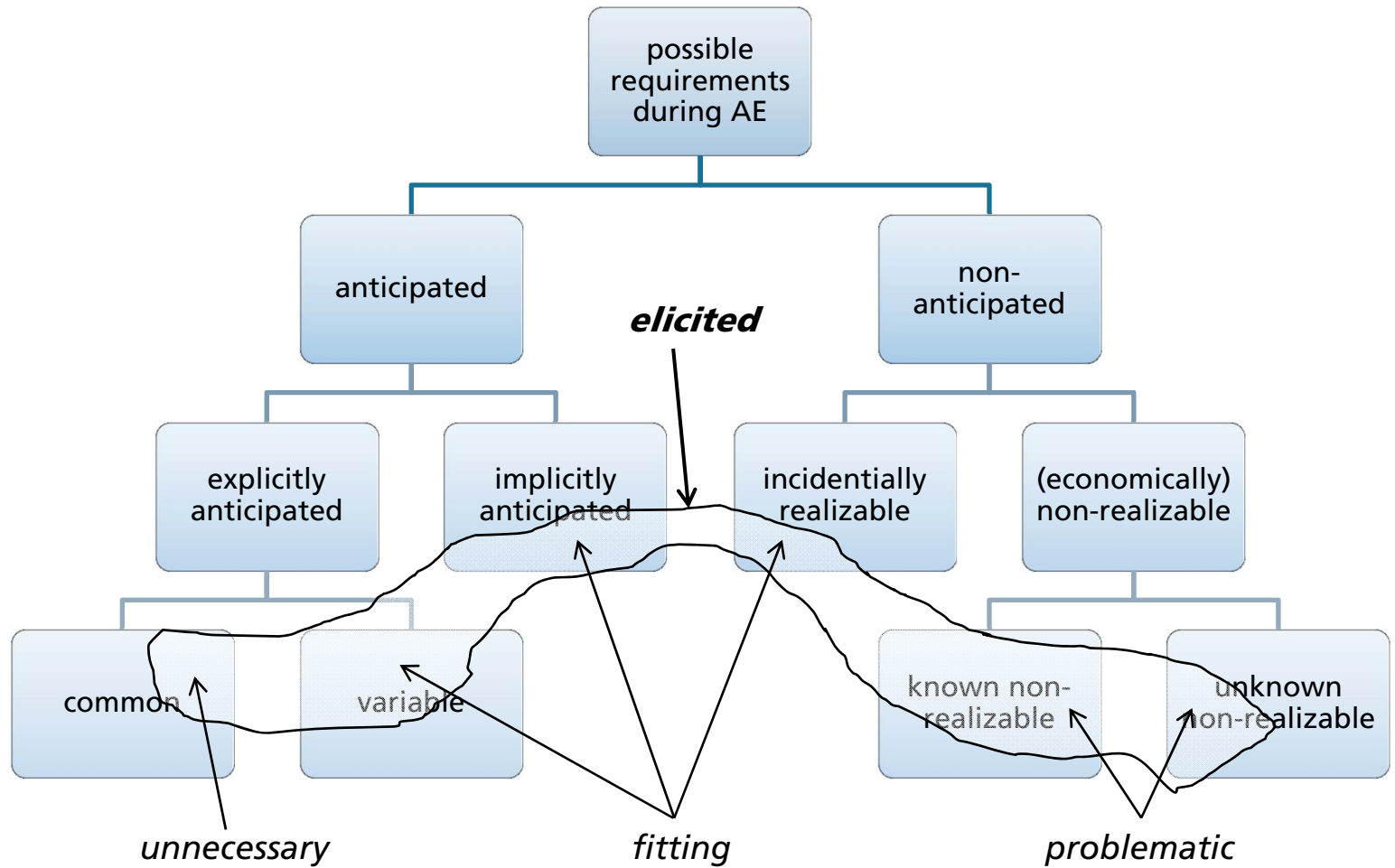
- **Kontrolliertes Experiment** mit Studenten der TU Kaiserslautern

 - Experimentaufgabe:
 - **Durchführung von Interviews (Rollenspielen)** mit kontrollierten Antworten durch einen (fiktiven) Stakeholder

 - **Randomisiertes, homogenes 2-Gruppen-Setting:**
 - Gruppe 1: Integrierte maßgeschneiderte Erhebungsscheckliste
 - Gruppe 2: Entscheidungs-/Variabilitätsmodell und BestPractice-Erhebungsscheckliste auf Stand der Technik

 - Gegenwärtig **22 von 29 Interviews ausgewertet**
-

Exkurs: Anforderungsarten



Zwischenstand Evaluationsergebnis (1)

- Probanden, die eine Erhebungsscheckliste nach der Methode verwenden
 - **wissen 1,65-mal mehr** über Rahmenbedingungen und Bedarfe im Application Engineering (sig = 0,005)
 - **erheben 68% weniger irrelevante** Informationen (sig = 0,001)
 - erkennen und **verneinen 3,7-mal mehr problematische** Anforderungen (sig = 0,000)
 - können **2,6-mal mehr** verhandlungskritische **Fragen** eines Stakeholders korrekt beantworten (sig = 0,000)
 - z.B. was kostet das extra?, warum geht das nicht?, ...

als Probanden der Kontrollgruppe

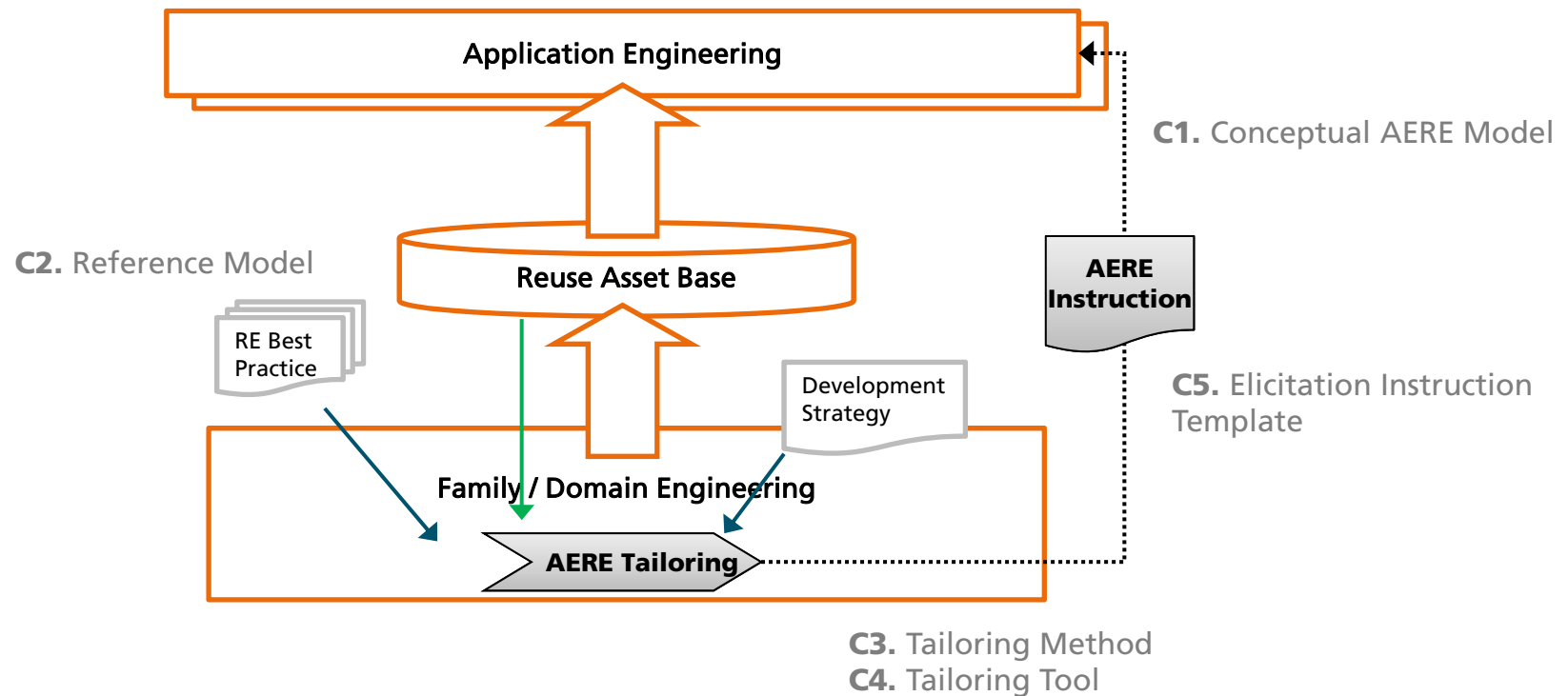
Zwischenstand Evaluationsergebnis (2)

- **Keine signifikanten Unterschiede** gibt es allerdings hinsichtlich
 - Anzahl der gestellten relevanten Fragen
 - Anzahl unnötig erhobener Commonalities
 - Anzahl der akzeptierten, machbaren Anforderungen („Satisfaction Fit“)
 - Anzahl der noch durch einen Experten zu klärenden Fragen
 - benötigten Zeit
- Qualitatives Feedback noch nicht ausgewertet, allerdings erster Eindruck
 - traditionelle Artefakte **nicht angemessen handhabbar** während der Erhebung

Wissenschaftliche Kernfrage

Wie kommt man zu diesen effektiveren Erhebungsschecklisten?

Lösungsidee und wissenschaftlicher Beitrag



Ablauf des Tailoring



- 1. Beschreibung der Produktlinie** (Produktwissen)
- 2. Identifikation der Architekturelementtypen** (Produktwissen)
- 3. Identifikation der Architekturelementen** (Produktwissen)
- 4. Identifikation der Flexibilitätsklassen** (Produktwissen)
- 5. Identifikation der Rahmenbedingungen** (Produktwissen)
- 6. Identifikation der Entwicklungsmeilensteine** (Prozesswissen)
- 7. Identifikation der Entwicklungsaktivitäten** (Prozesswissen)
- 8. Erhebung der Informationsbedarfe** (Prozesswissen)
- 9. Festlegung der Anforderungsarten** (Prozesswissen-Wissen)
- 10. Identifikation konzeptioneller Beziehungen** (BestPractice-Wissen)
- 11. Generierung des Erhebungsleitfadens** (BestPractice-Wissen)

Fallstudien-Beobachtungen zum Tailoring

- Schritte 1-7 brauchen bei einem realistischen, konfigurierbaren Informationssystem ca. 4 PT
- Tailoring kann von Personen innerhalb der Organisation selbst durchgeführt werden
 - bedarf aber Erläuterung
 - bedarf i.d.R. zwei Durchläufe
- **Gesamtaufwand < 10 PT**

Zusammenfassung

- In zahlreichen konfigurierbaren Systemen lassen sich wichtige Anforderungen **nicht immer explizit antizipieren** (offene Variabilität)
- Den bestmöglichen „Fit“ zwischen Anforderungen und wirtschaftlich vertretbaren Möglichkeiten zu erreichen **dauert entweder zu lange oder funktioniert überhaupt nicht**
- Derzeit keine Ansätze, welche Anforderungsingenieure während der Erhebung von nicht-explizit antizipierten Anforderungen unterstützen
- Idee: **Tailoring präziser Erhebungsschecklisten** durch Extraktion von Prozess- und Produktwissen über eine Produktlinie
- Erhebung kann dadurch **signifikant effektiver** durchgeführt werden
- Tailoring selbst geht **verhältnismäßig effizient** durch geeignete Werkzeugunterstützung

Mögliche Anschlussarbeiten

- Verbesserung des Tailoring-Tools
 - Anbindung an Architekturanalyse-Tools (automatisiert Schritt 2-3)
 - Verbesserung der Usability

- Toolunterstützung für die Erhebung
 - Verknüpfung mit Wiederverwendungsdatenbanken (Service Repositories) zum schnellen Auffinden existierender Realisierungen
 - Automatische Prüfung von Rahmenbedingungen (wenn Menge zu groß wird um manuell handhaben zu können)

Vielen Dank für Ihre Aufmerksamkeit!

Fragen?