

Software vergleichen

Andrea Herrmann
AndreaHerrmann3@gmx.de

25.11.2011 Fachgruppentreffen RE

Übersicht

1. Motivation
2. Stand der Forschung
3. Gap-Analyse versus Delta-Analyse
4. Grafischer Vergleich
5. Heuristischer Vergleich
6. Quantitativer Vergleich
7. Diskussion

1.) Motivation

Software auf der grünen Wiese



Anforderungs-Spezifikation
auf dem weißen Blatt

Produktauswahl oder
Weiterentwicklung/ Evolution



Anforderungen ->

Gap-Analyse: Welches Produkt passt?

Delta-Analyse: Was ändert sich?

2.) Stand der Forschung: Gap-Analyse

- RE for COTS, package selection, ...
- Literaturübersicht: Anil S. Jadhav, Rajendra M. Sonar: Evaluating and selecting software packages: A review. Information and Software Technology 51 (2009) 555–563
- “Multiple criteria decision-making problem”
- Methoden:
 - AHP
 - feature analysis
 - weighted average sum
 - expert system
 - fuzzy based approach

2.) Stand der Forschung: Delta-Analyse

- C. Ebert: „Systematisches Requirements Engineering und Management - Anforderungen ermitteln, spezifizieren, analysieren und verwalten“. dpunkt.verlag, 2008
 - Änderung <10% lokal, kaum Nachdokumentation nötig
 - 10-30% gute Analyse hinsichtlich Seiteneffekten nötig
 - >30% vollständige Nachdokumentation
- C. Rupp, D. Schüpferling, C. Pikalek: Deltarequirements – Auf den Spuren der Projektrealität, in: Informatikspektrum, Band 32, Heft 2, April 2009: Methode für Delta-Analyse
- 3 Arbeiten von Herrmann et al.
- Unterschied zwischen Gap-Analyse und Delta-Anforderungen bisher nicht Thema

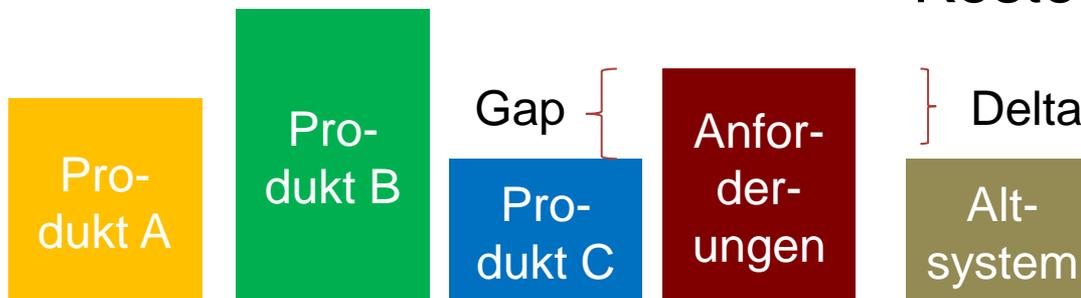
3.) Gap-Analyse versus Delta-Analyse

Gap-Analyse

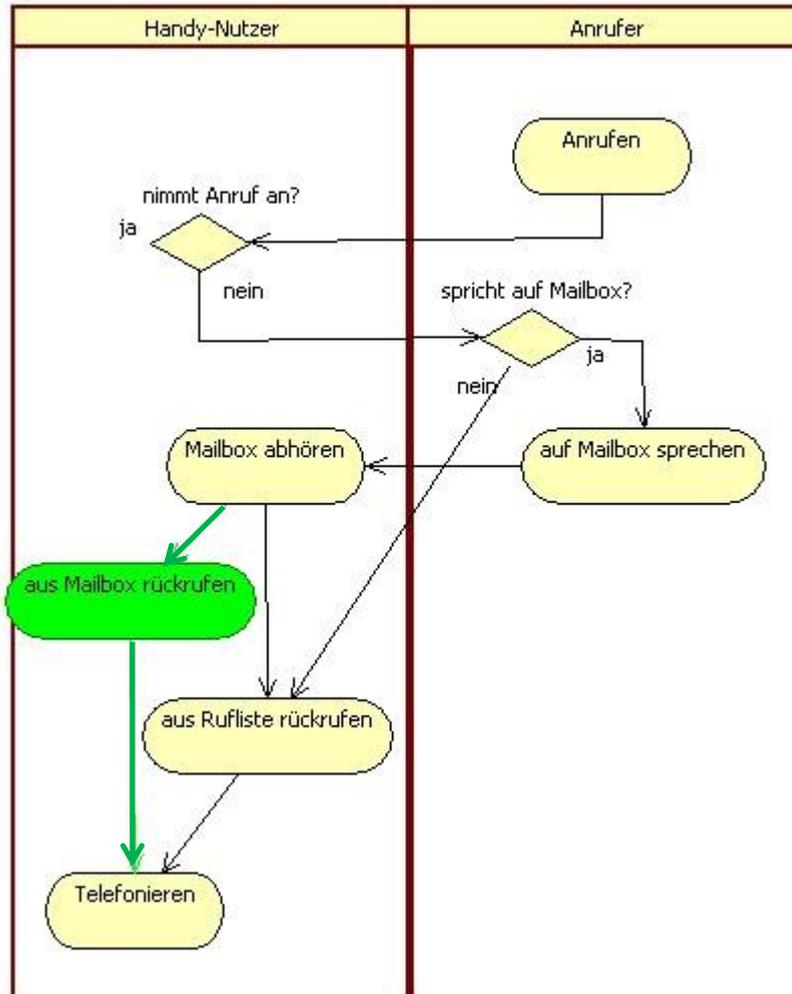
- Welches Produkt passt am besten zu den Anforderungen?
- Vergleich mehrerer Produkte mit den Anforderungen
- „similarity score“
- Nutzen-Maximierung

Delta-Analyse

- Was ändert sich gegenüber dem Altsystem?
- Vergleich der Anforderungen mit dem Altsystem
- Was muss geändert werden?
- Kosten-Minimierung



4.) Grafischer Vergleich



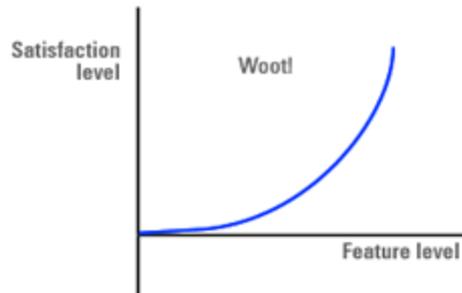
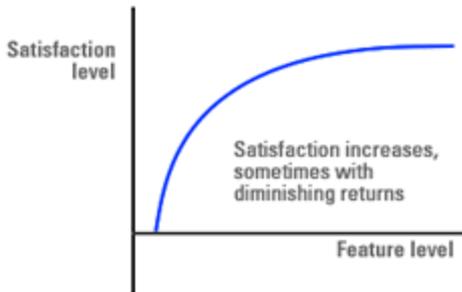
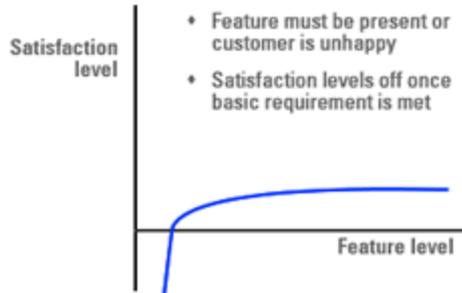
Farben:
schwarz = bleibt so,
grün = neu,
rot = ausblenden,
gelb = ändern

Quantifizierung:
•Abstandsmetrik
•Function Points

5.) Heuristischer Vergleich, z.B. mit Kano (1/2)

3 categories of requirements:

- **Must have/ basic:** e.g. State of the art-> system must have this feature, but it is no differentiating characteristic
- **More the better/ performance:** the better this requirement is fulfilled, the more satisfied the customer
- **Delighters/ excitement:** unexpected features, which delight the customer a lot



5.) Heuristischer Vergleich (2/2)

Beispiel-Heuristik für Gap-Analyse -> Entscheidung:

- Produkt, das nicht alle **Basic-Anforderungen** erfüllt, kommt nicht auf die Short List.
- Die **Performance**-Anforderungen bestimmen den Similarity Score.
- **Delighter**-Anforderungen geben bei gleichem Score den Ausschlag.

6.) Quantitativer Vergleich: gewichtete Summe

		<i>Ist-Features</i>						
		$\sum_i x_i w_i$						
		1	2	3	4	5	<i>Erfüllung x_i</i>	
							<i>x mal Gewicht w_i</i>	
Soll-Feature I	10	1					1	10
II	10		0,5		0,5		1	10
III	2						0	0
IV	1					2	2	2
V	1				1		1	1
VI	0,1			5			5	0,5
	24	1	0,5	5	1,5	2	23,5	98 <i>Gesamterfüllungsgrad</i>
							$\sum_i x_i w_i$	2 <i>Gap/ Delta</i>

QFD: concept planning and product design

Concept selection matrix

Für Auswahl zwischen alternativen technischen Realisierungen der Anforderungen

	Importance rating	Product concept A	Product concept B	Product concept C
Tech. requ. 1	4	20	12	12
Tech. requ. 2	4	20	12	20
Tech. requ. 3	6	18	30	$\Delta 6$
Tech. requ. 4	5	25	25	15
...				
Total		114	144	100

Bewertung: 1-3-5, mal importance

SAAM = Software Architecture Analysis Method

	Architektur 1					Architektur 2					
	Prio Szen.	unterst. Szen.	betr Komp	Komponente1	Komp. 2 ...	Komp. k1	unterst. Szen.	betr Komp	Komponente1	Komp 2 ...	Komp. K2
Szenario 1	2	1	3	1	1	1	1	1	1		
Szenario 2	1	1	1		1		1	2		1	1
Szenario 3	1	1	1			1	1	1			1
...											
Szenario n	1	1	1		1		0	0			
				1	3	3			1	1	1
Punkte		5	3	2,3			4	1,7	1		
		1	-1	-1			-1	1	1		

Gewichtetes Mittel der Anzahl der Szenarien pro Komponente

Gewichtetes Mittel der pro Szenario betroffenen Komponenten

Summe unterstützte Szenarien

Excel-Unterstützung erstellt von Andrea Herrmann, auf der Basis von:

Clements P., Bass L., Kazman R., Abowd G.: "Predicting Software Quality by Architectural-Level Evaluation",
 Proceeding of the Fifth International Conference on Software Quality, Oct 1995

Kazman R, Abowd G, Bass L, Clements P: "Scenario-based analysis of software architecture", IEEE Software 13(6), Nov 1996, 47-55

6.) Gap-Analyse: Handy-Beispiel

Soll-Features	Gewicht w_i	Ist-Features (Siemens S35)					Erfüllung x_i	x mal Gewicht w_i		
		1	2	3	4	5				
I - Anrufen mit manueller Eingabe	10	1					1	10		
II - Anrufen aus Adressbuch	5		1				1	5		
III - angerufen werden	10			1			1	10		
IV - Rückruf mit einem Klick	10				1		1	10		
V - Akkustand-Anzeige	5					1	1	5		
VI - E-Mails abrufen	1						0	0		
VII - Benutzerfreundlichkeit	5						0,7	3,5		
	46	1	1	1	1	1	0,7	43,5	95	Gesamterfüllungsgrad
									5	Gap

Soll-Features	Gewicht w_i	Ist-Features (Windows Phone htc)						Erfüllung x_i	x mal Gewicht w_i		
		1	2	3	4	5	6				
I - Anrufen mit manueller Eingabe	10	1						1	10		
II - Anrufen aus Adressbuch	5		2					2	10		
III - angerufen werden	10			1				1	10		
IV - Rückruf mit einem Klick	10				1			1	10		
V - Akkustand-Anzeige	5					0,8		0,8	4		
VI - E-Mails abrufen	1						1	1	1		
VII - Benutzerfreundlichkeit	5							0,9	4,5		
	46	1	2	1	1	0,8	1	0,9	49,5	108	Gesamterfüllungsgrad
										-8	Gap

6.) Delta-Analyse: Handy-Beispiel

Soll-Features	Gewicht w_i	Ist-Features (Siemens S35)					Erfüllung x_i	Delta	Delta mal Gewicht w_i	
		1	2	3	4	5				
I - Anrufen mit manueller Eingabe	5	1					1	0	0	
II - Anrufen aus Adressbuch	10		1				1	0	0	
III - angerufen werden	5			1			1	0	0	
IV - Rückruf mit einem Klick	10				1		1	0	0	
V - Akkustand-Anzeige	10					1	1	0	0	
VI - E-Mails abrufen	10						0	1	10	
VII - Benutzerfreundlichkeit	10					0,7	0,7	0,3	3	
	60	1	1	1	1	1	0,7		13	Gesamt-Delta

$\Delta = \text{abs}(x-1)$

Soll-Features	Gewicht w_i	Ist-Features (Windows Phone htc)						Erfüllung x_i	Delta	Delta mal Gewicht w_i	
		1	2	3	4	5	6				
I - Anrufen mit manueller Eingabe	5	1						1	0	0	
II - Anrufen aus Adressbuch	10		2					2	1	10	
III - angerufen werden	5			1				1	0	0	
IV - Rückruf mit einem Klick	10				1			1	0	0	
V - Akkustand-Anzeige	10					0,8		0,8	0,2	2	
VI - E-Mails abrufen	10						1	1	0	0	
VII - Benutzerfreundlichkeit	10						0,9	0,9	0,1	1	
	60	1	2	1	1	0,8	1	0,9		13	Gesamt-Delta

6.) Gap versus Delta

Soll-Features	Gewicht w_i	Ist-Features (Siemens S35)					Erfüllung x_i	x mal Gewicht w_i			
		1	2	3	4	5					
I - Anrufen mit manueller Eingabe	10	1					1	10			
II - Anrufen aus Adressbuch	5		1				1	5			
III - angerufen werden	10			1			1	10			
IV - Rückruf mit einem Klick	10				1		1	10			
V - Akkustand-Anzeige	5					1	1	5			
VI - E-Mails abrufen	1						0	0			
VII - Benutzerfreundlichkeit	5						0,7	3,5			
	46	1	1	1	1	1	0,7	43,5	95	Gesamterfüllungsgrad	
									5	Gap	

Gewicht: nutzenbasiert
versus kostenbasiert

Erfüllungsgrad: nutzenbasiert
versus kostenbasiert

Soll-Features	Gewicht w_i	Ist-Features (Siemens S35)					Erfüllung x_i	Delta = $\text{abs}(x-1)$			
		1	2	3	4	5		Delta	Delta mal Gewicht w_i		
I - Anrufen mit manueller Eingabe	5	1					1	0	0		
II - Anrufen aus Adressbuch	10		1				1	0	0		
III - angerufen werden	5			1			1	0	0		
IV - Rückruf mit einem Klick	10				1		1	0	0		
V - Akkustand-Anzeige	10					1	1	0	0		
VI - E-Mails abrufen	10						0	1	10		
VII - Benutzerfreundlichkeit	10						0,7	0,3	3		
	60	1	1	1	1	1	0,7			13	Gesamt-Delta

6.) Quantitativer Vergleich

Matrizen

oder Listen?

Gewicht

		1	2	3	4	5
I	10	1				
II	10		0,5		0,5	
III	2					
IV	1					2
V	1				1	
VI	0,1			5		

	Gewicht	Produkt A	Produkt B	Produkt C
I	10	1	1	0,5
II	10	1	1	1
III	2	0	2	0,5
IV	1	2	1	0
V	1	1	2	1
VI	0,1	5	1	1



6.) Vom Vergleich zur Entscheidung

- Gap: Produktauswahl: Welches Produkt hat den höchsten Punktestand bzw. Punkte/ Kosten?
- Delta: Kostenschätzung + Nutzen -> Priorisierung (nach Nutzen/ Kosten oder Kategorien) -> Releaseplanung -> Delta-Analyse verfeinern

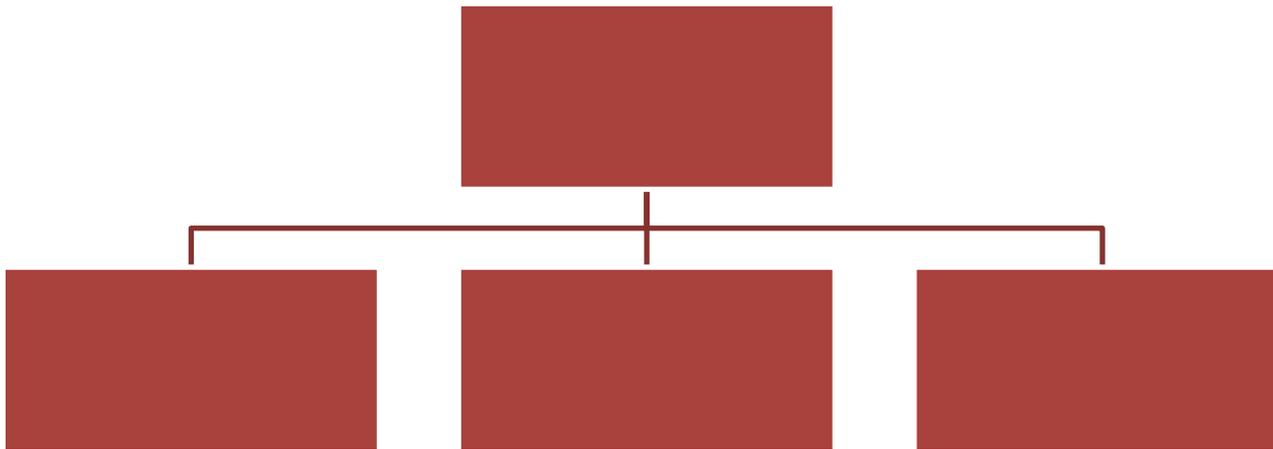
7.) Diskussion (1/2)

- **SAAM und TQM:** beide nutzenorientiert (Gap?), aber Mapping von Anforderungen auf Architektur (Delta)
- **Gap-Analyse** fragt „Welchen Anteil des Nutzens, den ich mir wünsche, werde ich bekommen?“
 - Gewichte entsprechend Nutzen
 - Erfüllungsgrad 80% = 80% so nützlich als wäre Anforderung vollständig erfüllt.
- **Delta-Analyse** fragt: „Welchen Anteil des Systems habe ich schon und wie viel muss ich noch mit wie viel Aufwand ändern oder ergänzen?“
 - Gewicht entsprechend Anteil am Gesamten (Anzahl Code-Zeilen? Kosten/Erstellungsaufwand?)
 - Erfüllungsgrad 80% = 20% des Aufwands noch zu erbringen

7.) Diskussion (2/2)

Anwendbarkeit auf **komplexere Systeme**?

- Aufwand n^2
- Hierarchische Darstellung?, aber:



Stopping rule??

Kein Gap



Kein Gap?

Vielen Dank für Ihre Aufmerksamkeit!

Fragen? Anmerkungen?