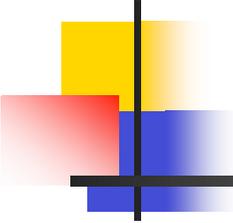


Software Engineering

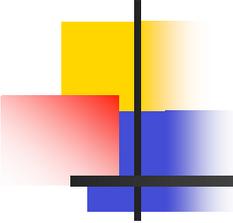
Besprechung zur Uebung 5

Verträge, Aufwand- und Risikoschätzung



Allgemeines, Formelles

- Übungen
 - Abholung vergangener Übungen: BIN 2.B.17
- Lösungen Übung 5
 - Mittelmässig bis gut gelöst. Verträge wurden meist nicht ideal gelöst.
- Form
 - Abgabe eines Dokuments (pdf)
 - Namen und Matr.Nummern auf das Deckblatt
 - Keine Umlaute oder Leerzeichen in den Dateinamen



Aufgabe 2.1: Vertragsformulierung (1)

- Ziel
 - Vertragsformulierung zu den gegebenen Schnittstellen
- Kriterien
 - Die Verträge müssen die Spezifikationen bzw. Funktionalitäten der Schnittstellen formalisieren
- Lösungen
 - Die Aufgabe war weniger gut gelöst

Aufgabe 2.1:

Vertragsformulierung (2)

```
class AnimalList {
    // Hier werden die Elemente gespeichert
    Animal[] elements = null;

    /**
     * Der Konstruktor der Liste. Es wird eine leere Liste
     * initialisiert.
     *
     * PRE -
     * POST elements <> null and size() == 0
     */
    public AnimalList() { ... }

    /**
     * Es wird ein neues Element in die Liste hinzugefügt.
     *
     * PRE -
     * POST size()@PRE + 1 == size() and
     *      elementExists(element)
     */
    public abstract void addElement(Animal element);

    /**
     * Es wird das Element an der Position „index“ der
     * Liste zurückgegeben.
     *
     * PRE 0 <= index and index < size()
     * POST result == elements[index]
     */
    public Animal getElementAt(int index) { ... }
```

```
/**
 * Diese Methode liefert die Anzahl an Elementen in der
 * Liste zurück.
 *
 * PRE -
 * POST result >= 0 and
 *      result == elements.length
 */
public int size() { ... }

/**
 * Diese Methode prüft ob das Element in der Liste
 * vorhanden ist und liefert true, wenn ja, und false,
 * wenn nein.
 *
 * PRE -
 * POST if there exists k: k >= 0 and k < size() - 1 and
 *      elements[k] == element then result == true
 *      else result == false endif
 */
public boolean elementExists(Animal element) { ... }
}
```

Verträge für die Schnittstellen

Aufgabe 2.2: Function Point Analyse (1)

- Ziel
 - Berechnung von Kosten und Durchlaufzeiten
- Kriterien
 - Zutreffende Schätzung der Eingangsvariablen
 - Kalibrierung des Modells
 - Bezugnahme auf Masszahlen über abgewickelte Projekte
- Lösungen
 - Meist nicht ideal gelöst

Der gegebene Oberflächenprototyp:

Reports abonnieren

Auswahl des Lebensraumes: Serengeti Savanne

Abonnieren

Nachricht bei Parameteränderungen

Report zu idealen Parameterwerten nach Monte-Carlo Simulation:

täglich

wöchentlich

Report mit Anzahl und Eckdaten aller getätigten Simulationsläufe:

täglich

wöchentlich

Benachrichtigung

Über Mail an die Email-Adresse: stoiber@ifi.uzh.ch

Beim Login

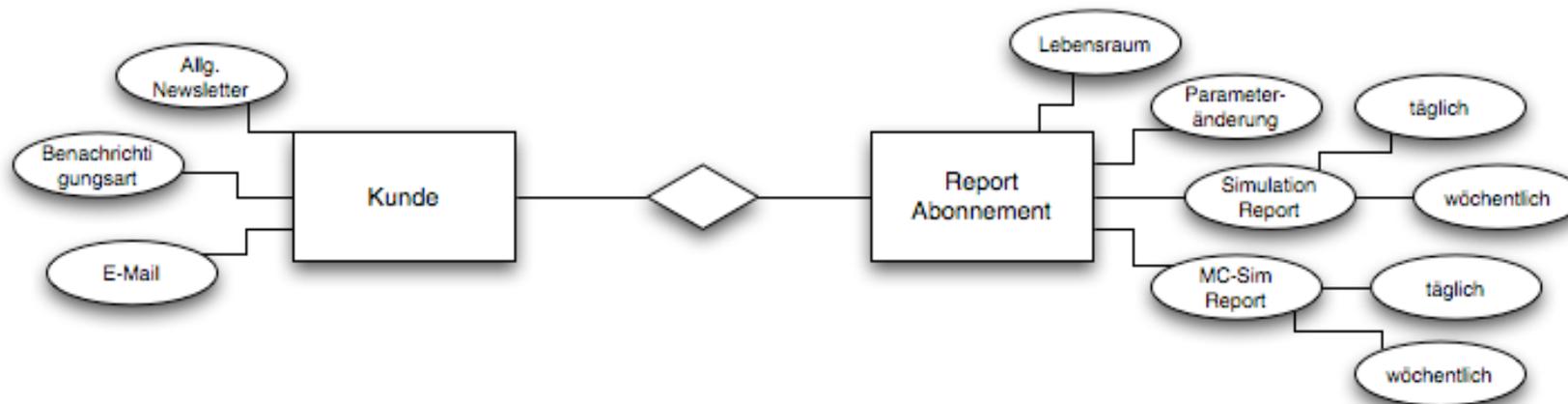
Allgemeiner Newsletter

Ich möchte den allgemeinen Newsletter von Simulation Systems Inc. erhalten

Auswahl speichern Fertig

Aufgabe 2.2: Function Point Analyse (2)

- Eine beispielhafte Datenmodellierung für das GUI:



- Dateneingaben

- 2 Datenbestände
- 9 Datenelemente
- --> Komplexität: mittel

Anzahl bearbeiteter Datenbestände	Anzahl unterscheidbarer Datenelemente in der Eingabe		
	1-4	5-15	>15
0-1	einfach	einfach	mittel
2	einfach	mittel	komplex
>2	mittel	komplex	komplex

Aufgabe 2.2: Function Point Analyse (3)

- Anfrage ("Abonnieren" wird durch Lebensraum aktualisiert)
 - 1 Datenbestand, 5 Datenelemente, Komplexität einfach
- Ext. Schnittstelle bzw. Int. Datenbestand
 - Annahme treffen
 - Entweder externe Schnittstelle
 - Annahme: einfach, Anbindung zu einer Datenbank
 - Oder interne Datenbestände
 - Annahme: einfach, 2 Datenbestände

Element	Schwierigkeitsgrad			Summe
	einfach	mittel	komplex	
Dateneingaben	_____ x 3 = _____	_____ x 4 = _____	_____ x 6 = _____	_____
Datenausgaben	_____ x 4 = _____	_____ x 5 = _____	_____ x 7 = _____	_____
Anfragen	_____ x 3 = _____	_____ x 4 = _____	_____ x 6 = _____	_____
Ext. Schnittstellen	_____ x 5 = _____	_____ x 7 = _____	_____ x10 = _____	_____
Int. Datenbestände	_____ x 7 = _____	_____ x10 = _____	_____ x15 = _____	_____
Function Point Rohwert (UFP)				_____

Einsetzen in die Tabelle:

-	1 x 4	-
-	-	-
1 x 3	-	-
1 x 5	-	-
(2 x 7)	-	-

Summe: UFP = 12

Aufgabe 2.2: Function Point Analyse (4)

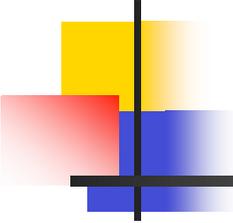
- Gesamt-Einflussfaktor (TDI)
 - Summe = 44

Nr.	Faktor	Wert	Einzusetzen sind Werte zwischen 0 und 5
1	Datenkommunikation		
2	Verteilte Funktionen		0 nicht vorhanden, kein Einfluss
3	Leistungsanforderungen		1 unbedeutender Einfluss
4	Belastung der Hardware		2 mäßiger Einfluss
5	Verlangte Transaktionsrate		3 durchschnittlicher Einfluss
6	Online-Dateneingabe		4 erheblicher Einfluss
7	Effiziente Benutzerschnittstelle		5 starker Einfluss
8	Online-Datenänderungen		
9	Komplexe Verarbeitungen		
10	Wiederverwendbarkeit		
11	Einfache Installation		
12	Einfache Benutzbarkeit		
13	Installation an mehreren Orten		
14	Änder- und Erweiterbarkeit		
Summe der Faktoren (TDI)			

Berechnung der Adjusted Function Points:

$$\text{VAF} = 0.65 + 0.01 * \text{TDI} = \underline{1.09} \text{ (value adjustment factor)}$$

$$\text{FP} = \text{UFP} * \text{VAF} = 12 * 1.09 = \underline{\mathbf{13.08}}$$



Aufgabe 2.3: COCOMO II (1)

- Ziel
 - Wie bei der FP-Analyse ebenfalls die algorithmische Aufwandschätzung
- Kriterien
 - Schätzung ausgehend von der Produktgrösse
- Lösungen
 - Etwas besser gelöst als die vorherigen Aufgaben

Aufgabe 2.3: COCOMO II (2)

a) Wahl des Projektleiters

- betroffene Skalierungsfaktoren
 - Präzedenz (Vertrautheit mit dem Produkt)
 - IT4U MA: *gering*, AppSysIT MA: *hoch*
 - Zusammenarbeit
 - IT4U MA: *hoch*, AppSysIT MA: *gering*
 - (eventuell noch Risiko-Umgang)
 - IT4U MA: -, AppSysIT MA: *evtl hoch*
- betroffene Kostenfaktoren
 - Personnel Continuity
 - Durch externen MA könnte es zu Unstimmigkeiten im Personal kommen...
 - Application Experience
 - Externe Bewerberin hat mehr Erfahrung im Anwendungsbereich ...
 - etc...

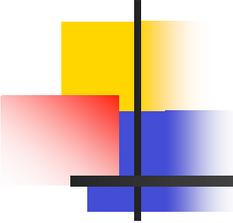
Faktor	Sehr gering	Gering	Nominal	Hoch	Sehr hoch	Extra hoch
Präzedenz	4,05	3,24	2,43	1,62	0,81	0
Flexibilität	6,07	4,86	3,64	2,43	1,21	0
Risiko-Umgang	4,22	3,38	2,53	1,69	0,84	0
Zusammenarbeit	4,94	3,95	2,97	1,98	0,99	0
Prozessreife	4,54	3,64	2,73	1,82	0,91	0

Factor	Very low	Low	Nominal	High	Very High	Extra high
Reliability required	0.75	0.88	1.00	1.15	1.39	
Database size		0.93	1.00	1.09	1.19	
Product complexity	0.75	0.88	1.00	1.15	1.30	1.66
Reuse required		0.91	1.00	1.14	1.29	1.49
Documentation required	0.89	0.95	1.00	1.06	1.13	
Execution time constraint			1.00	1.11	1.31	1.67
Storage constraint			1.00	1.06	1.21	1.57
Platform volatility		0.87	1.00	1.15	1.30	
Analyst capability	1.50	1.22	1.00	0.83	0.67	
Programmer capability	1.37	1.16	1.00	0.87	0.74	
Personnel continuity (turnover)	1.24	1.10	1.00	0.92	0.84	
Application experience	1.22	1.10	1.00	0.89	0.81	
Platform experience	1.25	1.12	1.00	0.88	0.81	
Language and tool experience	1.22	1.10	1.00	0.91	0.84	
Use of software tools	1.24	1.12	1.00	0.86	0.72	
Team co-location and communications support	1.25	1.10	1.00	0.92	0.84	0.78
Required development schedule	1.29	1.10	1.00	1.00	1.00	

Tabellen aus dem Skript

Achtung:
Faktor != Rechenwert
aus der Tabelle

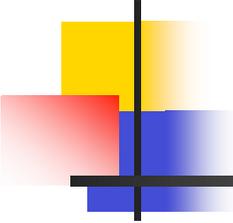
Skalierungsfaktor hoch --> niedriger Rechenwert !



Aufgabe 2.3: COCOMO II (3)

b) Outsourcing (ITIndia) oder in-house Entwicklung

- betroffene Skalierungsfaktoren
 - Zusammenarbeit
 - Intern: *hoch*, ITIndia: *gering*
 - Flexibilität
 - Intern: *hoch*, ITIndia: *gering*
- betroffene Kostenfaktoren
 - Programmer capability
 - Team in Indien hat mehr Erfahrung in der Programmierung; Faktor für ITIndia höher
 - Platform experience
 - Team in Indien kennt die Programmiersprache gut und auch Programmierumgebungen, etc.; Faktor ist für ITIndia höher
 - Team Co-Location
 - Das Indische Team ist schwieriger zu erreichen; Faktor ist für IT4U höher
 - etc...



Aufgabe 2.3: COCOMO II (4)

c) Beurteilung des Skalierungsfaktors *Flexibilität*

- Eigenschaften des Projektes die diesen Skalierungsfaktor betreffen
 - Termindruck
 - Hoher Termindruck macht weniger flexibel
 - Einflussnahme des Kunden
 - Je mehr Entscheidungen der Kunde an der Entwicklung trifft, desto weniger Spielraum bleibt den Entwicklern
 - Es ist mehr Flexibilität nötig um auf laufende Kundenwünsche einzugehen
 - Grösse des Teams
 - Je grösser das Team, desto weniger flexibel wird es (die Anzahl der Kommunikationskanäle steigt überproportional ...)
 - etc.

Aufgabe 2.3: COCOMO II (5)

d) Berechnungsfaktor KSLOC (kilo source lines of code)

- Szenario: Änderung des Kostenfaktors *Reuse Required* von *Nominal* auf *Very High*
- Berechnung: Änderung von KSLOC damit sich die Wiederverwendung lohnt

SF... Skalierungsfaktoren
EM... Kostenfaktoren
B... Wachstumsfaktor

$$\text{Aufwand} = 2,45 \cdot \text{KSLOC} \cdot \prod_{i=1}^{B-17} \text{EM}_i \quad (\text{aus einer Abgabe})$$

Wachstumsfaktor:

$$B = 1,01 + 0,01 \sum_{i=1}^5 \text{SF}_i$$

Summe der Skalierungsfaktoren (mit Nominalwerte):

$$\sum_{i=1}^5 \text{SF}_i = 2,43 + 3,64 + 2,53 + 2,97 + 2,73 = 13,3$$

$$B = 1,01 + 0,01 * 13,3 = 1,153$$

EM_N=1.0 (nominal)

EM_{VH}=1.29 (very high)

$$\text{Aufwand} = 2,45 \cdot \text{KSLOC} \cdot \prod_{i=1}^{B-16} \text{EM}_i \cdot \text{EM}_N = 2,45 \cdot (x \cdot \text{KSLOC}) \cdot \prod_{i=1}^{B-16} \text{EM}_i \cdot \text{EM}_{VH}$$

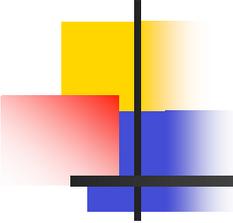
$$2,45 \cdot \text{KSLOC} \cdot \prod_{i=1}^{B-16} \text{EM}_i \cdot 1 = 2,45 \cdot (x \cdot \text{KSLOC}) \cdot \prod_{i=1}^{B-16} \text{EM}_i \cdot 1,29 \quad | : (2,45) \cdot \prod_{i=1}^{B-16} \text{EM}_i$$

$$\text{KSLOC} = x \cdot \text{KSLOC} \cdot 1,29 \quad | : \text{KSLOC}$$

$$1 = x \cdot 1,29$$

$$(1/1,29) = x \rightarrow x = 0,8$$

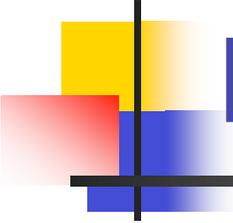
Ergebnis: Verringerung von KSLOC um mindestens 20%, damit sich die Wiederverwendung lohnt.



Aufgabe 2.3: COCOMO II (6)

e) Voraussetzung für zuverlässige Werte

- Gegenstand
 - einfache Anwendungssoftware
- Umfeld
 - stabile Umgebung
- Gleichungen müssen “unternehmensspezifisch kalibriert” werden
 - Einfluss von *Erfahrungswerten* aus vergangenen Projekten
 - Berechnungen müssen auf die konkreten Bedingungen *angepasst* werden



Aufgabe 2.4: Risikoschätzung (1)

- Ziel
 - Risikoabschätzung für das konkrete Projekt
 - *Vorbeugung* gegen Bedrohungen des sachlichen oder wirtschaftlichen Werts des Projektes

- Kriterien
 - Relevanz/Plausibilität der Risiken für das Projekt
 - Keine Risiken aus dem Skript
 - Das Risiko ergibt sich aus Schadenshöhe * Eintrittswahrscheinlichkeit

Aufgabe 2.4:

Risikoschätzung (2)

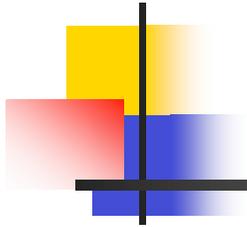
Beispiele

Allgemein

- Unmotivierte Entwickler
 - $E = 2, S = 4, \text{Risiko} = E * S = 8$. Massnahmen: Arbeitsbedingungen verbessern, Firmenfeier, Abläufe/Prozesse verbessern, Anreize für gute Leistungen schaffen, ...
- Falscher Technologieeinsatz
 - $E = 1, S = 7, \text{Risiko} = 7$. Massnahmen: Experimentelle Prototypen entwickeln, Experten mit einbeziehen, ...

Konkret in diesem Projekt

- Nicht alle Kunden zufrieden (viele und geographisch verteilte Kunden)
 - $E = 4, S = 7, R = 28$. Massnahmen: Regelmässiges Feedback, ...
- Probleme mit Übersetzung / Sprachen
 - $E = 3, S = 3, R = 9$. Massnahmen: vorab schon nach Übersetzern suchen, ...
- Zahlungsfähigkeit der Auftraggeber
 - $E = 3, S = 9, R = 27$. Massnahmen: durch Verträge absichern, ...
- etc...



Danke für die Aufmerksamkeit.