



# Software Engineering Übung 5

## Verträge, Aufwand- und Risikoschätzung

### 1 Information

#### 1.1 Daten

- Ausgabe: Di. 13.11.2007
- Abgabe: So. 25.11.2007
- Besprechung: Fr. 30.11.2007 12:15 Uhr, BIN 2.A.10

#### 1.2 Formales

Die Dateien, welche zu Ihrer Abgabe gehören, müssen in eine .zip-Datei gepackt werden (Diagrammen und andere Dokumente als PDF, Quellcode als Textdateien in einem separaten Unterverzeichnis). Die Abgabe erfolgt per Email an stoiber@ifi.uzh.ch.

#### 1.3 Gruppen

Die Übung ist in 3er Gruppen zu lösen. Falls die Aufgaben aufgeteilt wurden, muss klar ersichtlich sein, wer welchen Teil bearbeitet hat. Alle Gruppenmitglieder müssen über alle Teile Auskunft geben können.

### 2 Aufgabenstellung

Die Aufgaben beziehen sich auf die Fallstudie aus den Übungen 2 bis 4.

#### 2.1 Vertragsformulierung (4 Punkte)

Gegeben ist die Schnittstelle der Java-Klasse `AnimalList`, welche zur Verwaltung der Objekte der Klasse `Animal` in unserer Software eingeführt werden soll. Damit kann das bisher verwendete Java-Array `Animal[]` durch diese dynamische Datenstruktur ersetzt werden. Hinweise: Im Nachfolgenden bedeutet `{ ... }`, dass die betreffende Methode eine Implementierung hat, diese aber hier nicht gezeigt ist.

Fügen Sie die fehlenden Verträge (jeweils Pre- und Postconditions) zur spezifizierten Schnittstelle hinzu. Für die bessere Verständlichkeit soll bei der Formulierung von Gleichheit in Ausdrücken von Verträgen das doppelte Gleichheitszeichen (`==`) verwendet werden. Verwenden Sie ansonsten die Notation, wie Sie diese aus der Vorlesung kennen. Sie müssen ausser den Verträgen keine weitere Implementierung vornehmen.

```

class AnimalList {

// Hier werden die Elemente gespeichert
Animal[] animals = null;

/**
 * Der Konstruktor der Liste. Es wird eine leere Liste initialisiert.
 *
 * PRE:
 * POST:
 */
public AnimalList() { ... }

/**
 * Es wird ein neues Element in die Liste hinzugefügt.
 *
 * PRE:
 * POST:
 */
public void addElement(Animal element) { ... }

/**
 * Es wird das Element an der Position "index" der Liste zurückgegeben.
 *
 * PRE:
 * POST:
 */
public Animal getElementAt(int index) { ... }

/**
 * Diese Methode liefert die Anzahl an Elementen in der Liste zurück.
 *
 * PRE:
 * POST:
 */
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:
 */
public boolean elementExists(Animal element) { ... }

}

```

## 2.2 Aufwandschätzung, Function Points (6 Punkte)

Gegeben sei der Oberflächenprototyp für die Abonnieung von Neuigkeiten innerhalb einer Simulationsumgebung (Siehe Abb. 1).

The screenshot shows a web form titled "Reports abonnieren". It is divided into several sections:

- Auswahl des Lebensraumes:** A dropdown menu with "Serengeti Savanne" selected.
- 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:**
  - Beim Login**
- Allgemeiner Newsletter:**
  - Ich möchte den allgemeinen Newsletter von Simulation Systems Inc. erhalten**

At the bottom, there are two buttons: "Auswahl speichern" and "Fertig".

Abbildung 1: Oberflächenprototyp für die Abonnieung von Reports zu einem gewünschten Lebensraum

Die folgenden zusätzlichen Informationen zu dieser Eingabemaske sind gegeben:

- In der Liste der Auswahl der gewünschten Lebensräume wird initial der erste Eintrag angezeigt.
- Es werden die Einstellungen für die Benachrichtigung zum aktuell in der Liste ausgewählten Lebensraum, z.B. Serengeti Savanne, angezeigt. Diese sind Nachricht bei Parameteränderungen, Report zu idealen Parameterwerten nach MC-Sim. und Report mit Anzahl und Eckdaten getätigter Simulationsläufe (letztere jeweils täglich oder wöchentlich).
- Wird ein anderer Lebensraum ausgewählt, so werden die Einstellungen dieses Lebensraums angezeigt.
- Änderungen werden über die Schaltfläche *Auswahl speichern* gespeichert. Dabei werden immer nur die aktuellen Eingaben gespeichert.
- Mit der Schaltfläche *Fertig* wird *Reports abonnieren* verlassen. Änderungen die nicht gespeichert wurden, gehen dabei verloren.

a. Ermitteln Sie die Adjusted Function Points gemäss der Vorgehensweise aus der Vorlesung. Beschreiben Sie dabei Ihre Vorgehensweise und dokumentieren Sie die Annahmen, welche Sie treffen. Die Berechnung der Function Points für den Oberflächenprototyp betrifft nur Dateneingaben und Anfragen, nicht die Datenausgaben. Nehmen Sie für die Anfragen, für die externen Schnittstellen und die internen Datenbestände jeweils, falls vorhanden, einen einfachen Schwierigkeitsgrad an. Für die Gewichtung der Anfragen können die Werte der Tabelle auf Folie 22, Kapitel 16 aus der Vorlesung analog verwendet werden. Bei der Berechnung des Gesamteinflussfaktors werden die Effizienz der Benutzerschnittstelle, die Wiederverwendbarkeit, die einfache Benutzbarkeit und die Erweiterbarkeit als sehr hoch bewertet. Komplexe Verarbeitungen und Installationen an mehreren Orten haben keinen Einfluss. Für die restlichen Faktoren wird ein durchschnittlicher Einfluss angenommen. Falls Sie noch weitere Annahmen treffen, dokumentieren und begründen Sie diese.

b. Was brauchen Sie neben den Adjusted Function Points noch, um eine Schätzung des Aufwandes für den Teil unter a.) zu machen? Begründen Sie die Aussagen kurz.

### 2.3 Aufwandschätzung, COCOMO2 (6 Punkte)

Die Firma IT4U will sich um den Auftrag, das System für Systems Simulation Inc. zu entwickeln, bewerben. Bevor man ein Angebot macht, wird intern eine Aufwandschätzung vorgenommen. Verschiedene Möglichkeiten werden verglichen und ihre Auswirkungen auf den Aufwand werden berechnet. Als algorithmisches Schätzverfahren wird COCOMO2 verwendet.

a. Das Projektteam besteht aus Mitarbeitern, welche bereits gemeinsame Projekte bearbeitet haben. Projekte die dem SavannahSim-Projekt ähnlich sind, gab es allerdings noch keine. Für den Posten des Projektleiters gibt es zwei Bewerbungen. Die erste Bewerbung ist von einem Mitarbeiter der schon lange bei IT4U arbeitet. Er macht sich grosse Hoffnungen den Job zu erhalten, denn er möchte Erfahrungen im Leiten von Projekten sammeln. Falls er den Posten nicht bekommt, wird er als Teammitglied beim Projekt dabei sein.

Die andere Bewerbung kommt von einer Mitarbeiterin von ApplicationSystemsIT, einer Firma welche sich zum Teil auf Software im Bereich von Simulationssystemen spezialisiert hat. Sie hat Erfahrung in der Entwicklung von Simulationen für reale Lebensräume.

Welche Skalierungsfaktoren und Kostenfaktoren sind von der Wahl des Projektleiters betroffen? Diskutieren Sie die Auswirkungen beider Wahlmöglichkeiten auf diese Faktoren.

b. Da bisher in der Praxis schon viele Simulationssysteme in der Programmiersprache C geschrieben wurden, soll, zu Zwecken besserer Wiederverwendbarkeit vorhandener Software und Kompatibilität zu vorhandenen Schnittstellen, die Software in C entwickelt werden. Es gibt einige Programmierer in der Firma, denen sehr gute Fähigkeiten im Programmieren mit C zugesprochen werden. Es müssten aber noch weitere Mitarbeiter geschult werden, um ein Projekt dieser Grösse bewältigen zu können. Alternativ könnte die gesamte Programmierung an das indische Software-Unternehmen ITIndia outgesourct werden. ITIndia stehen viele in der Sprache C erfahrene Programmierer zur Verfügung.

Welche Skalierungsfaktoren und Kostenfaktoren sind von der Entscheidung, wo die Programmierung stattfinden soll, betroffen? Diskutieren Sie die Auswirkungen beider Möglichkeiten auf diese Faktoren.

c. Sie müssen den Skalierungsfaktor *Flexibilität* beurteilen. Nennen Sie zwei Eigenschaften des Projekts, welche dabei eine Rolle spielen und beurteilen Sie diese anhand des gegebenen Projektes. Falls Sie dazu Annahmen treffen müssen, begründen Sie diese.

**d.** Um wieviel Prozent müsste der Berechnungsfaktor KSLOC (Kilo Source Lines Of Code) verringert werden können, damit sich Wiederverwendung, welche den Wert des Kostenfaktors *Reuse Required* von Nominal auf Very High ändert, lohnt? Die restlichen Kostenfaktoren bleiben gleich und die Skalierungsfaktoren werden alle mit Nominal bewertet. Zur Lösung soll auch der Rechnungsweg dargestellt werden.

**e.** Was ist Voraussetzung, damit die Berechnung mit COCOMO2 zuverlässige Werte liefert? Begründen Sie Ihre Aussage kurz.

#### 2.4 Risikoschätzung (4 Punkte)

Im Skript Software Engineering zur Vorlesung finden Sie im Kapitel 15 die zehn häufigsten Risiken bei Software-Projekten.

**a.** Finden Sie 4 weitere Risiken für Ihr Softwareprojekt aus Übung 2 (keines der Risiken aus dem Skript).

**b.** Nehmen Sie eine Risikobewertung für die vier weiteren gefundenen Risiken vor, indem Sie das Risiko, sowie die Schadenshöhe je auf einer Skala zwischen 1 - 10 bewerten.

**c.** Stellen Sie passende Massnahmen zusammen, um die Risiken zu mindern oder zu eliminieren.