



Universität
Zürich^{UZH}

Institut für Informatik



Software Engineering Übung 5

Verträge, Aufwand- und Risikoschätzung

1 Informationen

1.1 Daten

- Ausgabe Di 15.11.2011
- Abgabe So 27.11.2011 bis 23:59 Uhr
- Besprechung am Di 06.12.2011 um 11:50 Uhr

1.2 Formales

Die Lösungen sollen als PDF Datei mit dem Namen **Ex[n]_[NameA_NameB_NameC].pdf** abgegeben werden, wobei [n] die Nummer der Übung ist und [NameA_NameB_NameC] die Nachnamen der Gruppenmitglieder sind. Die PDF Datei sollte ausserdem ebenfalls Ihre Namen und Matrikelnummern beinhalten.

Mailen sie Ihre Lösungen vor dem Abgabetermin an todoran@ifi.uzh.ch und nhoby@access.uzh.ch. Der Betreff der E-mail sollte mit **[SE EX HS11]** beginnen. Falls Sie zusätzliche Abgabematerialien (z.B. Source Code) haben, mailen Sie bitte ein Archiv (.zip-File), welches alle Dateien, einschliesslich dem PDF, enthält. Benennen sie das Archiv anhand der oben erwähnten Konventionen. Als Abgabedateien sind nur PDF-Dokumente und Source Code Dateien erlaubt, keine Bilder.

Die Übungen sollen in 3er Gruppen gelöst werden. Jedes Gruppenmitglied muss über alle Teile der Lösungen Auskunft geben können. Verspätete Abgaben werden korrigiert, aber nicht bewertet.

2 Aufgabenstellung

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

2.1 Vertragsformulierung (4 Punkte)

Ein mögliches Einsatzgebiet der *Vector* Klasse in Java ist die Implementierung einer mehrdimensionalen, verwandten Datenstruktur: Einer Matrix. Matrizen werden überall dort verwendet, wo mehrdimensionale Darstellungen von Daten benötigt werden, sehr intensiv zum Beispiel im Gebiet der Computergrafik. Ein Beispiel einer *Matrix* Klasse besitzt die unten aufgeführten Methoden. Zu Beginn sind alle Elemente in der Matrix *null*.

Hinweis: 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 (Folienkapitel 5.6). Sie müssen ausser den Verträgen keine weitere Implementierung vornehmen.

```
public class Matrix {

    // size of the matrix
    protected int height, width;
    // vector of row vectors
    protected Vector rows;

    /**
     * The constructor specifies the width and height of the matrix
     *
     * @pre
     * @post
     *
     * @return matrix with h rows and w columns
     */
    public Matrix(int h, int w) {...}

    /**
     * Manipulates individual elements of the matrix: tells what object is
     * at a specific location
     *
     * @pre
     * @post
     *
     * @return the object at (row,col)
     */
    public Object get(int row, int col) {...}

    /**
     * Manipulates individual elements of the matrix: changes a specific
     * location to a value; the elements of the Matrix are initially null,
     * but can be reset with this method
     *
     * @pre
     * @post
     *
     */
}
```

```

* @return
*/
public void set(int row, int col, Object value) {...}

/**
* Inserts a new row (r) of null elements
*
* @pre
* @post
*
* @return
*/
public void addRow(int r) {...}

/**
* Inserts a new column (c) of null elements
*
* @pre
* @post
*
* @return
*/
public void addCol(int c) {...}

/**
* Removes a row of the matrix
*
* @pre
* @post
*
* @return a Vector of removed values
*/
public Vector removeRow(int r) {...}

/**
* Removes a column of the matrix
*
* @pre
* @post
*
* @return a Vector of removed values
*/
public Vector removeCol(int c) {...}

/**
* Gives the number of columns in the matrix
*
* @pre
* @post
*

```

```

* @return number of columns in the matrix
*/
public int width() {...}

}

```

2.2 Aufwandschätzung, Function Points (6 Punkte)

Gegeben sei der Oberflächenprototyp für das Ticketverkaufs-System der *iCommute'n'Smile* (siehe Abb. 1).

The image shows a web form titled "Arbeitnehmerticket kaufen". It contains the following elements:

- Firma:** A text field containing "Software Inc."
- Name:** An empty text input field.
- Rabatt:** A slider control set to 50%.
- Gültigkeitsdauer:** Three radio button options: 3, 6, and 12 Monate.
- Buttons:** Two buttons at the bottom: "Kaufen" and "Abbrechen".

Abbildung 1: Oberflächenprototyp für den Kauf eines Angestelltentickets

Die folgenden zusätzlichen Informationen zu dieser Eingabemaske sind gegeben:

- Alle Felder sind zu Beginn leer.
- Wenn ein Arbeitnehmerticket gekauft wird überprüft das System, ob sich die Gültigkeitsdauer des gewünschten Tickets mit einem bereits ausgestellten Ticket überschneidet.
- Zu einem Arbeitgeber werden alle Tickets gespeichert, die dieser bereits für seine Angestellten gekauft hat.

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 Eingaben 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 einfache Benutzbarkeit und die Erweiterbarkeit als sehr hoch bewertet, die Wiederverwendbarkeit als hoch. Komplexe Verarbeitungen und Belastung der Hardware haben einen unbedeutenden 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 Programs4People will sich für den Auftrag, das System für die iCommute'n'Smile 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 VVA Projekt ähnlich sind, gab es allerdings noch keine. Für den Posten der Projektleitung gibt es zwei Bewerbungen. Die erste Bewerbung ist von einer Mitarbeiterin, die schon lange bei Programs4People arbeitet. Sie macht sich grosse Hoffnungen, den Job zu erhalten, denn sie möchte Erfahrungen im Leiten von Projekten sammeln. Falls sie den Posten nicht bekommt, wird sie als Teammitglied beim Projekt dabei sein.

Die andere Bewerbung kommt von einem Mitarbeiter von TravelSystemsIT, einer Firma welche sich zum Teil auf Software im Verkehrs-Bereich spezialisiert hat. Er hat Erfahrung in der Entwicklung von ähnlichen Systemen.

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 Ticketverkaufs-Systeme mit dem J2EE Framework implementiert wurden, soll, zu Zwecken besserer Wiederverwendbarkeit vorhandener Software und Kompatibilität zu vorhandenen Schnittstellen, die Software mit diesem Framework entwickelt werden. Es gibt einen Programmierer in der Firma, dem sehr gute Fähigkeiten im Programmieren mit Java und J2EE zugesprochen werden. Es müssten aber noch weitere Mitarbeiter geschult werden, um das VVA Projekt bewältigen zu können. Alternativ könnte die gesamte Programmierung an das indische Software-Unternehmen ITIndia outgesourct werden. ITIndia stehen viele mit J2EE 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 *Risiko-Umgang* 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 Wiederverwendbarkeit, 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 14 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.