



Software Engineering Übung 5

Verträge, Aufwand- und Risikoschätzung

1 Informationen

1.1 Daten

- Ausgabe: Di. 11.11.2008
- Abgabe: So. 23.11.2008 bis 23:59 Uhr
- Besprechung: Di. 02.12.2008 12:15 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 jeanneret@ifi.uzh.ch. Der Betreff der E-mail sollte mit **[SE EX HS08]** 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.

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)

Gegeben ist die Java-Klasse `Queue`, welche genutzt wird, um Elemente (z.B. E-Mails) zu speichern, bevor sie weiter verarbeitet werden.

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 sowie Klasseninvarianten) 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.

```
/**
 * A collection designed for holding elements prior to processing.
 * Elements are usually (but not necessarily) ordered in a FIFO
 * (first-in-first-out) manner.
 *
 * This queue implementation has a limited capacity and does not accept null
 * elements.
 *
 * @inv
 */
public class Queue<E> {

    /**
     * The number of elements in this queue.
     */
    private int size;

    /**
     * The maximal capacity of this queue.
     */
    private int capacity;

    /**
     * Checks whether an additional element can be enqueued.
     *
     * @return true if the queue is full
     *
     * @pre
     * @post
     */
    public boolean isFull() {...}

    /**
     * Checks whether this queue contains no element.
     *
     * @return true if the queue is empty
     *
     * @pre
     * @post
     */
    public boolean isEmpty() {...}

    /**
     * Retrieves and removes the head of this queue.
     *
     * @return the head element of the queue
     */
}
```

```

    * @throws NoSuchElementException
    *         if the queue is empty
    *
    * @pre
    * @post
    */
public E dequeue() {...}

/**
 * Inserts the specified element into this queue if it is possible to do so.
 *
 * @param element
 *         the element to be enqueued
 *
 * @throws NullPointerException
 *         if element is null
 * @throws IllegalStateException
 *         if the queue is full
 *
 * @pre
 * @post
 */
public void enqueue(E element) {...}
}

```

2.2 Aufwandschätzung, Function Points (6 Punkte)

Gegeben sei der Oberflächenprototyp für die Einschreibung, womit sich die Studierenden für Workshops eintragen können. (Siehe Abb. 1).

Die folgenden zusätzlichen Informationen zu dieser Eingabemaske sind gegeben:

- In den Listen der Auswahl der Prioritäten wird initial der erste Eintrag angezeigt (die Einträge sind alphabetisch sortiert).
- Das Feld für die Semesterangabe ist zu Beginn leer.
- In der Liste der Auswahl des letzten Kurses wird initial der erste Eintrag angezeigt.
- Änderungen werden über die Schaltfläche *Speichern* gespeichert, und *Workshop Auswahl* wird verlassen.
- Mit der Schaltfläche *Abbrechen* wird *Workshop Auswahl* 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

Workshop Auswahl

Mit diesem Formular können Sie ihren Workshop auswählen.

1. Priorität:

2. Priorität:

3. Priorität:

4. Priorität:

Geben Sie Ihr Semester an:

Wählen Sie Ihren letzten Kurs aus:

Abbildung 1: Oberflächenprototyp für die Einschreibung durch Studierende

22, Kapitel 16 aus der Vorlesung analog verwendet werden. Bei der Berechnung des Gesamt-Einflussfaktors 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 uns 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 Workshop-Management 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 Akademie-Bereich spezialisiert hat. Sie hat Erfahrung in der Entwicklung von ähnlichen akademischen 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 viele akademische 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 einige Programmierer in der Firma, denen sehr gute Fähigkeiten in Java zugesprochen werden, die aber kaum über Kenntnisse in J2EE verfügen. Es müssten 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 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 *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.