

Objektorientierte Analyse und Design (Übungsblatt 2)

WS 2018/2019

1 Modus für Übungsblatt 2

- Teamarbeit (Gruppenarbeit)
- Gesamtpunkte: 30P

2 Aufgabe 1 (6P)

Erstellen Sie mit Visual Paradigm ein UML (Analyse-)Klassendiagramm für folgende Beschreibung.¹ Sollte es notwendig sein, treffen Sie geeignete zusätzliche Annahmen. Als Hintergrundinformation: das Modell sollte als Basis für die Entwicklung einer Software zur Unterstützung der Hausverwaltung einer Hausverwaltungsgesellschaft verwendbar sein.

Beschreibung: Häuser besitzen eine Hausnummer und eine Adresse und bestehen aus einer oder mehreren Wohnungen. Jede Wohnung hat eine Wohnungsnummer und befindet sich in einem bestimmten Stock. Wohnungen werden von Mietern gemietet. Mieter werden charakterisiert durch ihren Vornamen, Nachnamen und das Geburtsdatum. Für die Hausverwaltung ist es wichtig zu wissen, von wann bis wann eine bestimmte Wohnung von einem Mieter gemietet wurde (die Historie der Mieter einer Wohnung soll also bekannt sein). Eine Wohnung besteht aus einem oder mehreren Räumen, wobei Räume durch eine Raumnummer beschreiben werden. Räume sind durch Türen miteinander verbunden – in diesem Zusammenhang soll gespeichert werden, welcher Raum über welche Tür mit welchem anderen Raum verbunden ist. In einem Raum können sich Messgeräte befinden (bspw. ein Messgerät zur Bestimmung des Heizungsverbrauchs). Jedes Messgerät ist genau einem Raum zugeordnet. Einmal pro Jahr werden sogenannte Ablesungen der einzelnen Messgeräte durch Mitarbeiter der Hausverwaltung durchgeführt.

¹ Details zur Lizenzvergabe von Visual Paradigm erfahren Sie in den Übungen.

Für jede Ablesung sollen der Wert, die Einheit und der durchführende Mitarbeiter gespeichert werden. Die Historie der Messwerte für ein Messgerät ist von Relevanz.

3 Aufgabe 2 (24P)

Organisation. Bestimmen Sie einen Teamleiter (Projektleiter) und sorgen Sie für eine klare *Aufgabenverteilung* im Projekt. Sinnvolle Aufgaben in Ihrem Projekt sind bspw. Projektleitung, Analyse & Design, Test, Implementierung und Usability Engineering. Klarerweise wird (mit Ausnahme des Projektleiters) nicht immer eine Person genau eine Rolle innehaben. Etablieren Sie in weiterer Folge die für Ihr Team notwendige *Infrastruktur* (Entwicklungsumgebung, Datenbank, Modellierungsumgebung, grober Projektplan, Aufteilung der Aufgaben unter den Teammitgliedern).

Inhalt. Es ist eine Software auf Basis des *Unified Process* zu erstellen. Im Rahmen des OAD Softwareprojektes werden *drei Phasen* durchlaufen (*Inception*, *Elaboration* und *Construction*) wobei jede Phase aus genau *einer Iteration* besteht (siehe auch: Abbildung 1). Das Ergebnis jeder Iteration ist eine funktionsfähige Software ("*executable release*"), deren Umfang sich abhängig von der Phase ändert (erweitert). Zu den Ergebnissen jeder Iteration gibt es im Rahmen von Abgabegesprächen Feedback von den Studienassistenten. *Achtung:* die Software aus der vorhergehenden Phase sollte in der nächsten Phase weiter verwendet werden! Zur Priorisierung der Anforderungen (Requirements) verwenden sie das Tool InnoSensR. Für jede Phase (*Inception*, *Elaboration* und *Construction*) soll es ein Release geben. Nach der Deadline zur Gruppeneinteilung erhält Ihre Gruppe detaillierte Informationen sowie eine Einladung zum Tool.

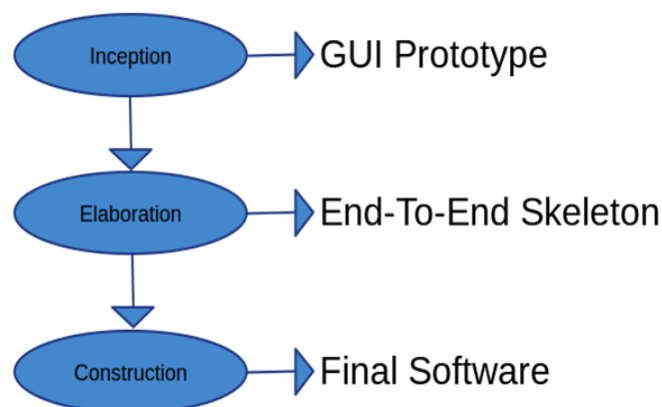


Fig. 1. OAD Anwendungsprojekt: dahinter liegender (vereinfachter) Softwareprozess basierend auf dem Unified Process.

Aufgaben für die Phase Inception. Für die im Rahmen von OAD zu erstel-

lende Anwendung (Umgebung) *T-REC* [1] ist eine objektorientierte Analyse durchzuführen. Es sind sowohl die *ursprünglichen Anforderungen* (= *basic properties* [1]) zu berücksichtigen als auch die eigenen Ideen einfließen zu lassen. Erstellen Sie die für die Anwendung aus Ihrer Sicht *wichtigsten Use Cases* (ohne diese würde die Anwendung keinen Sinn machen) und ein entsprechendes *Use Case Diagramm*. Stellen Sie fest, welche Klassen es in der Anwendungsdomäne gibt, welche Eigenschaften diese besitzen und wie sie zueinander in Beziehung stehen. Stellen Sie in weiterer Folge die so identifizierten Informationen als *UML (Analyse-)Klassendiagramm* in *Visual Paradigm* dar. Bedingungen (Constraints), die Sie im Klassendiagramm nicht beschreiben können, stellen Sie als *textuelle Ergänzungen* zum Klassendiagramm dar. Beachten Sie, dass es hier um die Analyse (also um das *was?*) und nicht um das *wie?* geht, d.h., modellieren Sie Klassen aus der Anwendungsdomäne und nicht Klassen, die aus technischer Sicht zusätzlich zur Lösung des Problems notwendig sind!

Entwickeln Sie für die identifizierten Use Cases einen *UI Prototypen*, der klar die Use Cases selbst und deren Zusammenhänge darstellt. Abhängig von den Zielsetzungen Ihres Projektteams kann der Prototyp entweder als *Web Anwendung* oder als *Java Swing Anwendung* realisiert werden. Die identifizierten Use Cases müssen bzgl. ihrer Implementierung *priorisiert* werden. Verwenden Sie hierzu das Tool InnoSensr! Jedes Gruppenmitglied muss die Evaluierung bezüglich der Priorisierung individuell/einzeln (und unabhängig von den anderen Kollegen) vornehmen. WICHTIG: Der Status von Requirements muss von Ihnen bis zum Ende der Abgabe aktuell gehalten werden. Dies wird beim Abgabegespräch überprüft. Mehr Informationen hierzu finden Sie im Beschreibungsdokument des Tools InnoSensr das Sie nach Deadline der Gruppenanmeldung von uns erhalten.

Sämtliche Aktivitäten sind basierend der Zeiterfassungsvorlage aus dem TeachCenter zu protokollieren und beim Abgabegespräch an die Studienassistenten auszuhändigen - WICHTIG: Sie müssen keinen Namen bei der Zeitprotokollierung angeben - Ihre Gruppennummer ist ausreichend!

Bei offenen Fragen wenden Sie sich an die OAD Newsgroup bzw. nutzen Sie am Anfang jeder Übungseinheit die Gelegenheit, offene Fragen zu klären.

4 Abgabe

- *Abgabe 1*: Lösungen zu Aufgabe 1 (*UML Analyse-Klassendiagramm* in *Visual Paradigm*).
- *Abgabe 2*: Lösungen zu Aufgabe 2 (*Business Case*, die für *T-REC* wichtigsten *Use Cases* (inkl. *Priorisierung* und entsprechender Beschreibung der

Szenarien)², ein entsprechendes *Use Case Diagramm* in Visual Paradigm, ein *UML Analyse-Klassendiagramm* in Visual Paradigm, grober *Projektplan*, *Screenshots* und Erklärungen zur Funktionsweise Ihres UI Prototypen; im Wesentlichen sollten die Screenshots die grafische Oberfläche zu den von Ihnen definierten Use Cases darstellen).

- *Deadline für die Abgabe*: 09.11.2018 um 23:59 Uhr.

Abgabe 1 und Abgabe 2 sollen in einem integrierten .pdf gespeichert werden. Die Abgabe erfolgt über das TeachCenter. Beim Abgabegespräch muss von jedem Team der ausführbare Prototyp mitgebracht und vorgeführt werden.

Achtung: Das Abgabesystem wird zur Deadline geschlossen. Zu späte Abgaben werden nicht akzeptiert. Jedes Übungsblatt muss mit min. 30% abgeschlossen werden, um die Lehrveranstaltung positiv zu absolvieren. Bedenken Sie die Möglichkeit, dass wir Abgaben auf Plagiate überprüfen.

Viel Erfolg!

References

- [1] Alexander Felfernig and Martin Stettinger. T-REC – Intelligent Tourist Recommendation. *Domain Description for the course on Object-oriented Analysis and Design (slides)*, 2018.

² Ein Template steht im TeachCenter zur Verfügung.