



Auftrag Lösungsarchitektur M2

1 Einleitung

Dieses Dokument beschreibt abschliessend die Resultate, die am Meilenstein M2 abgegeben werden müssen. Dazu enthält es die Kriterien, nach denen die Resultate bewertet werden.

Laden Sie die Artefakte vor der Präsentation auf Moodle in den entsprechenden Abgabe-Ordner hoch. Die genauen Abgabemodalitäten sind auf Moodle beim jeweiligen Meilenstein beschrieben.

2 Abgaben

Die folgenden Resultate müssen am Meilenstein 2 abgegeben werden.

2.1 Quellcode

Der aktuelle Stand des Quellcodes Ihres Projektes muss abgegeben werden.

Für die Abgabe des Quellcodes legen Sie eine einzelne ZIP-Datei mit folgenden Inhalten im vorgesehenen Abgabe-Ordner von Moodle ab: kompletter Quellcode samt aller Konfigurationsdateien, die nötig sind, um das Programm auszuführen; generierte API-Dokumentation als HTML (Javadoc o.ä.).

2.2 Technischer Bericht I

Der Technische Bericht I beschreibt neben den Resultaten der Anforderungsanalyse auch die von Ihnen geplante Architektur und erste Designentscheide. Er ist als PDF-Dokument ebenfalls auf Moodle hochzuladen.

3 Aufbau des Technischen Berichts I

Es ist ein technischer Projektbericht (ca. 20 Seiten) mit den wichtigsten, bisher erarbeiteten Resultaten des Projekts zu erstellen.

3.1 Inhaltsstruktur

- Use-Case-Modell
- Zusätzliche Anforderungen
- Domänenmodell
- Softwarearchitektur
- Design-Artefakte
- Implementation
- Projektmanagement
- Glossar

3.2 Use-Case-Modell

Alle identifizierten Use Cases werden in einer Übersicht in einem UML-Use-Case-Diagramm mit den dazugehörigen Akteuren dargestellt. Dabei soll das UML-Use-Case-Diagramm auch die Abgrenzung gegenüber externen Systemen darstellen (Systemkontext).



Der oder die fachlich wichtigsten Use Cases müssen ausführlich («fully dressed») ausformuliert werden.

Weitere wichtige Use Cases werden normal («casual») ausformuliert, während der Rest der Use Cases noch kurz («brief») beschrieben wird.

Für das Standardszenario des fachlich wichtigsten und vollständig ausformulierten Use-Case wird ein System-Sequenzdiagramm (SSD) erstellt.

Für die wichtigsten Use-Cases werden UI-Sketches erstellt und allenfalls bei einem komplexen UI die Navigationsmöglichkeiten (Dialogablauf) in einem Diagramm dargestellt.

3.3 Zusätzliche Anforderungen

Hier werden weitere funktionale und vor allem die nicht funktionalen Anforderungen (Randbedingungen und Qualitätsanforderungen) angegeben. Zusätzlich werden wichtige Regeln des Problemgebietes (z.B. Geschäfts- oder Spielregeln) und sonstige wichtige Informationen (z.B. weitere Vorgaben und Randbedingungen für die Entwicklung der Applikation) aufgelistet.

3.4 Domänenmodell

Es ist ein Domänenmodell zu entwickeln, das die (fachlichen) Konzepte der wichtigsten Use-Cases darstellt. Das Domänenmodell wird mit einem (konzeptuellen) UML-Klassendiagramm dargestellt.

3.5 Softwarearchitektur

Darstellung und Beschreibung der gewählten Softwarearchitektur mit Begründung des Entscheides.

Die Softwarearchitektur ist mindestens aus der Sicht der Schichtung (logische Architektur) und Subsysteme/Module mit einem UML-Paketdiagramm darzustellen. Spezieller Fokus hat dabei die Trennung von UI und Domänenlogik sowie der Einsatz von Frameworks.

Falls es eine verteilte Anwendung ist, müssen das Verteilungsmodell (Client/Server, Peer-to-Peer etc.) und die gewählte Kommunikationsart beschrieben und begründet werden.

3.6 Design-Artefakte

Dokumentieren Sie das bestehende Design mit einem Design-Klassendiagramm (DCD) und geeigneten Interaktionsdiagrammen. Zeigen Sie für mindestens 4 Systemoperationen ein entsprechendes Interaktionsdiagramm. Dazu sind wichtige, bisher gefällte Design-Entscheide zu dokumentieren.

3.7 Implementation

Beschreiben Sie hier kurz, wie die gewählte Softwarearchitektur verifiziert wurde. Die gewählte Softwarearchitektur muss mit einer partiellen und unvollständigen Implementation (Prototyp) der wichtigsten Use Cases verifiziert sein.

Eine kurze Erklärung der Teststrategie: Welche Tests werden auf welchen Teststufen gemacht mit einer Begründung (Unit-, Integration-, und Systemtests).



3.8 Projektmanagement

Die Planung der bisherigen Iterationen und der zugehörige Aufwand sind erfasst und dargestellt. Pro Iteration wird der tatsächliche Aufwand und die erreichten Resultate mit dem geplanten Aufwand und den gesteckten Zielen verglichen und die allfällig getroffenen Massnahmen festgehalten. Zum Schluss gehört auch die detaillierte Planung der nächsten Iterationen inklusive Aufwandsschätzung zur Abgabe, wie auch die aufdatierte Risikoliste.

3.9 Glossar

Ein Glossar, wo die wichtigsten Begriffe des Problemgebiets (s. Domänenmodell) erklärt und definiert werden, ist zu erstellen. Dieses Glossar wird im Laufe des Projektes laufend ergänzt und aktualisiert.

4 Fachliche Bewertung (30%)

Alle Kapitel des Technischen Berichts I werden fachlich separat bewertet gemäss den folgenden Kriterien:

- Vollständigkeit gemäss der obigen Beschreibung
- Formale Korrektheit gemäss dem Theorieunterricht in SWEN1
- Verständlichkeit und Ausführlichkeit
- Angemessenheit gegenüber Ihrem Projekt

Der Quellcode wird gemäss folgenden Aspekten bewertet:

- Qualität des Programmcodes gemäss Clean-Code Kriterien.
- Wie umfangreich werden die kritischen Aspekte Ihrer gewählten Architektur abgedeckt.

Die Bewertung der 8 Kapitel des Technischen Berichts I und die 2 Aspekte des Quellcodes werden gleich gewichtet und ergeben die Note des Meilenstein 2.