Klausurprüfung (Fachtheorie) aus Programmieren und Software Engineering

# P.A.S.S.T. Projekt Aufwandsschätzung Software Solution für Teams

Für ein Softwarehaus soll eine Applikation realisiert werden, welche die dort agierenden Teams und Personen dabei unterstützen soll, die Aufwände für zukünftige Kundenprojekte durch mehrere Mitarbeiter rasch und in hoher Qualität zu schätzen.

Sie werden mit der Umsetzung zweier Prototypen betraut.

1. Entwickeln sie ein vollständiges Domänenmodell für die Datenhaltung und die Implementierung der fachlichen Logik laut nachfolgender Auflistung der abzubildenden Geschäftsfälle. Diese Implementierung stellt **KEINE** Datenspeicherung (zB JDBC/JPA) bzw. Eingabe (via Web-Interface) bereit.
   1. Der Fokus liegt auf korrekter Implementierung der Geschäftsregeln und die Sicherstellung der Umsetzung durch umfassende Testabdeckung.
   2. Des weiteren liegt der Fokus auf sauberem objekt-orientiertem Design und der allfälligen Verwendung von Patterns.
2. Entwickeln sie eine Konzept-Webapplikation zur Abbildung einer Teilmenge von vereinfachten Geschäftsfällen um die Mechanik und Darstellung der Datenerfassung und –ausgabe zu konzeptionieren.
   1. Der Fokus liegt auf einem technischen Durchstich durch alle Schichten, Domain, Persistence (zB JPA), Service und Presentation

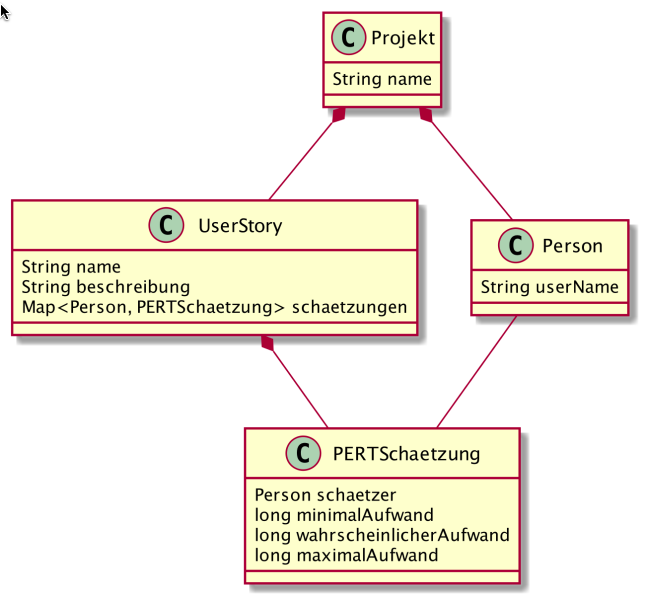
## ***Rahmenbedingungen***

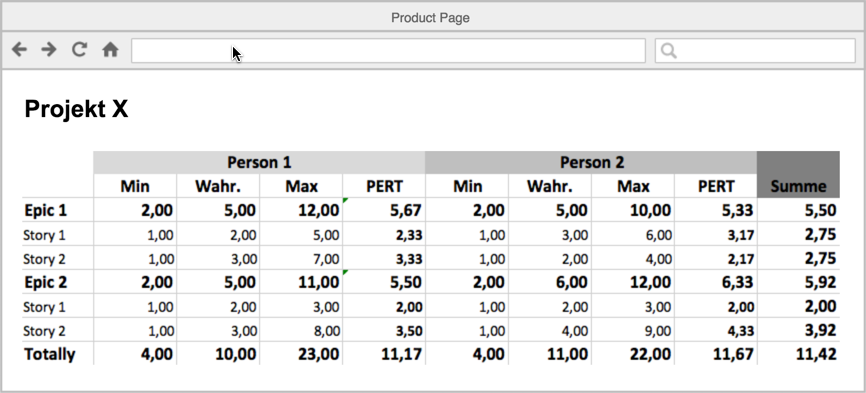
* Die Applikation ist als Webanwendung zu implementieren.
* Die Benutzerschnittstelle ist als HTML Anwendung für aktuelle Browser zu implementieren.
* Für die gestalterische Aufbereitung der Anwendung ist ein Team von Grafikern beauftragt – Ihre Implementierung soll sich vorerst ausschließlich darum kümmern, dass die Gestaltung der Anwendung einfach möglich ist (Stichwort: Trennung von „Was wird dargestellt“ von „Wie wird es dargestellt“).
* Die Implementierung der Benutzerschnittstelle soll alle beschriebenen Geschäftsfällen ermöglichen (soll sich aber auf die Funktionalität beschränken).
* Alle relevanten Daten der unten angeführten Geschäftsfälle sind in einer relationalen Datenbank zu speichern.
* Achten Sie auf eine hohe Qualität bei der Implementierung!

## ***Abzubildende Geschäftsfälle und –regeln***

1. ***Prototyp Domänenmodell***
2. Das Modell soll mehrere unabhängige Projekte behandeln können. Die Verwaltung der Projekte ist NICHT zu implementieren.
3. Ein Projekt soll eine 2-stufige Hierarchie von Projektplanungselementen, bestehend aus Epic 🡪 Userstory, behandeln können.
4. Ein Projekt soll mehrere Personen, die Schätzungen für einzelne Projektplanungselemente abgeben sollen, handhaben können.
5. Ein Projektplanungselement soll mehrere von Kommentaren handhaben können.
6. Ein Kommentar soll einer Person zugeordnet werden können, ein Erstellungsdatum und einen Status (Erstellt, Beantwortet).
7. Jedes Projektplanungselement soll beauskunften können, ob es noch unbeantwortete Kommentare gibt.
8. Ein Kommentar soll mehrere Subkommentare handhaben können.
9. Für Schätzungen sollen initial folgende Implementierungsvarianten bereitgestellt werden:
   1. Zwei-Werte-Schätzung: Enthält minimalen und maximalen Aufwand. Der Schätzwert ergibt sich aus dem arithmetischen Mittelwert.
   2. PERT-Schätzung: Enthält minimalen, wahrscheinlichsten und maximalen Aufwand. Der Schätzwert ergibt sich aus der Formel  
      (1 \* min. + 4 \* wahrscheinlichster + 1\* max.) / 6
10. Die Schätzwerte sollen auf Projekt- oder Epic-Ebene für Subhierarchien aufaggregiert werden.
11. Eine Userstory soll mehrere personenbezogene Schätzungen handhaben können, die arithmetisch gemittelt werden sollen.
12. Jedes Projektplanungselement soll beauskunften können, ob alle dem Projekt zugeordneten Personen Schätzungen abgegeben haben.
13. ***Prototyp Webapplikation***

Dem Prototyp der Webapplikation liegt folgendes vereinfachtes Domänenmodell zugrunde.



**

Schematische Darstellung

1. Die Verwaltung von Projekten, Personen und Epics ist NICHT zu implementieren, stattdessen nutzen sie ein Default Projekt sowie eine Anzahl von Default Personen und Epics.
2. Implementieren sie das vereinfachte Domänenmodell samt Persistence Layer und entsprechendem Service Layer. Stellen sie eine ausreichende Unit- und Integrationstestabdeckung sicher.
3. Implementieren sie eine Eingabemaske zur Erfassung der Schätzwerte für eine Projektstruktur für eine auszuwählende Person.
4. Implementieren sie eine Ausgabeseite für die Schätzungen eines Projektes, welche der schematischen Darstellung folgt.
   1. Personen die noch keine Schätzungen abgegeben haben, sind mit 0-Werten darzustellen.

## ***Realisierung***

Die Aufgabenstellung kann mittels:

* Java Enterprise Edition (Spring)
* ASP.NET (C#) oder
* JavaScript (NodeJS)

realisiert werden.

## ***Bewertung***

Die Bewertung berücksichtigt sowohl welche Geschäftsfälle implementiert wurden als auch die Qualität der Implementierung.

Die Punkte werden je nach Qualität folgender Beurteilungskriterien (BK) vergeben.

BK1 Die in der Geschäftsfallbeschreibung geforderte Funktionalität ist vorhanden ~**50% der zu erreichenden Punkte**

BK2 Die geforderte Funktionalität ist durch Tests sicher gestellt (Testüberdeckung, Gut-Fälle sind abgedeckt, mögliche Fehlerfälle sind abgedeckt) ~**25% der zu erreichenden Punkte**

BK3 Die Prinzipien der objektorientierten Implementierung wurden umgesetzt (z.B. single responsibility, don’t repeat yourself, naming of classes and methods, encapsulation, coupling) ~**20% der zu erreichenden Punkte**

BK4 Stil und Lesbarkeit des Codes (z.B. einheitliche Formatierung, Namengebung, kein überflüssiger Code) ~**5% der zu erreichenden Punkte**

Unterschieden wird je Geschäftsfall wie folgt:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Geschäftsfall** | **BK1** | **BK2** | **BK3** | **BK4** | **Summe** |
| Domänenmodell 2 – Hierarchie | 10 | 5 | 4 | 1 | 20 |
| Domänenmodell 3 – Personen | 5 | 3 | 2 | 1 | 11 |
| Domänenmodell 4 – Kommentare | 5 | 3 | 2 | 1 | 11 |
| Domänenmodell 5 - Kommentarattribute | 5 | 3 | 2 | 1 | 11 |
| Domänenmodell 6 - unbeantwortete Komm. | 4 | 2 | 1 | 1 | 8 |
| Domänenmodell 7 – Kommentarhierarchie | 4 | 2 | 1 | 1 | 8 |
| Domänenmodell 8 - Schätzvarianten | 6 | 3 | 2 | 1 | 12 |
| Domänenmodell 9 - Aggregation | 4 | 2 | 1 | 1 | 8 |
| Domänenmodell 10 – personenb.Schätzungen | 4 | 2 | 1 | 1 | 8 |
| Domänenmodell 11 – fehlende Schätzungen | 4 | 2 | 1 | 1 | 8 |
| **Zwischensumme Aufgabenstellung 1** |  |  |  |  | **105** |
| Webapplikation 2 – Basislayer (D/Pers/S) | 30 | 15 | 4 | 5 | 54 |
| Webapplikation 3 - Eingabe | 15 | 5 | 3 | 5 | 28 |
| Webapplikation 4 - Ausgabe | 15 | 5 | 3 | 5 | 28 |
| **Zwischensumme Aufgabenstellung 2** |  |  |  |  | **110** |
| **Summe** |  |  |  |  | **225** |

Die verbleibenden Punkte werden anhand des Gesamteindrucks der Abgabe vergeben. Kriterien hierfür sind z.B.:

* die gewählte Architektur
* die Organisation (Struktur) des Projektes
* die Klarheit der Lösung im Gesamten
* die strukturelle Qualität der resultierenden HTML Seiten

Noten-Punkte-Aufschüsselung:

Nicht Genügend 0-50%, Genügend >50%-62,5%, Befriedigend >62,5%-75%, Gut >75%-87,5%, Sehr Gut >87,5%-100%

Erlaubte Hilfsmitteln: leeres Projektfile mit allen Abhängigkeiten, eBooks zur DB Programmierung und API Dokumentationen