

## Übungsblatt 3: Softwarearchitektur

### Übungsinhalt:

In dieser Übung erstellen Sie eine valide Softwarearchitektur für ihr Projekt. Im Anschluss bereiten Sie dann die Implementierung vor, indem Sie ein Projekt bei github einrichten.

### Aufgaben:

#### 1. Konzeptionelles Datenmodell erstellen

Erstellen Sie ein konzeptionelles Datenmodell für ihre Anwendung mit Hilfe von draw.io (UML Klassenmodell). Siehe dazu den Inhalt der Vorlesung „Spezifikation und Planung“.

#### 2. A-Architektur erstellen und Schnittstellen beschreiben

Erstellen Sie aus dem konzeptionellen Datenmodell und der Liste an Funktionen (= Use Cases) mit der Priorität A und C, die Sie in der letzten Übung identifiziert haben, eine A-Architektur mitsamt ihren Schnittstellen. Dokumentieren Sie die A-Architektur mit Hilfe von draw.io (UML Komponentenmodell). Beschreiben Sie die Schnittstellen der Komponenten direkt in Java-Syntax.

#### 3. Variabilitätsanalyse der A-Architektur

Führen Sie eine Variabilitätsanalyse der A-Architektur durch. Spielen Sie alle Tasks, die sie mit der Priorität B versehen haben, auf ihre Auswirkungen in Bezug auf die A-Architektur durch. Beantworten Sie dabei insbesondere die folgenden Fragen:

- Welche Stellen müssen angepasst werden?
- Wie sehr muss die Architektur dafür umgebaut werden?

Falls für eine Änderung viele Stellen angepasst werden müssen oder die Architektur im großen Stil umgebaut werden muss, dann diskutieren Sie eventuell notwendige Anpassungen an der A-Architektur, um dies zu verhindern.

Hinweis: Falls Sie keine Tasks mit der Priorität B versehen haben, dann überlegen Sie sich Funktionen, die womöglich in Zukunft in die Software integriert werden müssen.

#### 4. T-Architektur erstellen

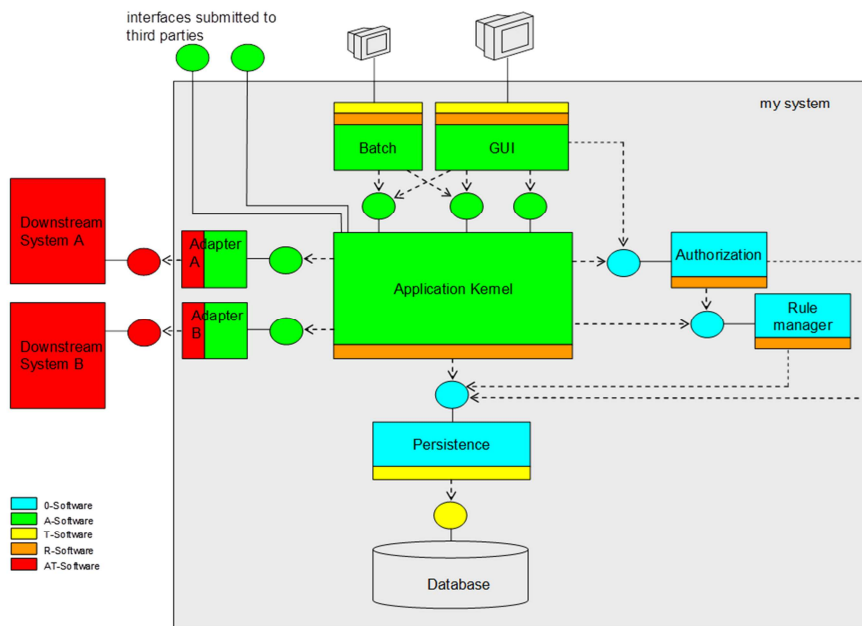


Abbildung 1: T-Musterarchitektur für Informationssysteme

Erstellen Sie aus den technologischen Vorgaben heraus eine T-Architektur auf Basis der in der Vorlesung vorgestellten Musterarchitektur für Informationssysteme. Konzentrieren Sie sich dabei insbesondere auf die verwendeten Technologien und Open-Source-Bausteine. Dokumentieren Sie die T-Architektur als UML-Komponentenmodell mit draw.io.

#### 5. github Projekt

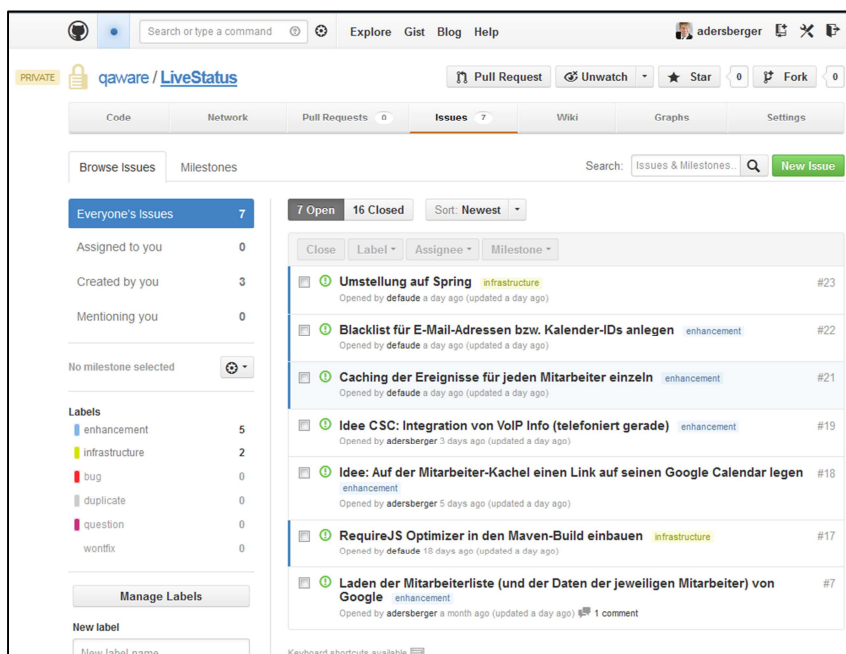


Abbildung 2: github

Als zentrale Projekt-Infrastruktur für die Projekte nutzen wir im weiteren Verlauf github. Github stellt ein Versionskontrollsystem, ein Ticketsystem und einen Wiki zur Verfügung. Ziel dieser Aufgabe ist die initiale Einrichtung des Bereichs auf github für ihr Projekt. Gehen Sie dazu wie folgt vor:

1. Erstellen Sie auf github (<https://github.com>) ein Repository für ihr Projekt und posten Sie den Namen bzw. die URL des Projekts auf Yammer.
2. Überlegen Sie sich eine vernünftige Verzeichnisstruktur für ihr Projekt. Übertragen Sie dorthin alle bisher erstellten Dokumente. Sehen Sie dabei auch einen Bereich für ihren Quellcode vor.
3. Erstellen Sie eine Startseite für ihr Projekt (README.md auf oberster Ebene im Repository). Hinterlegen Sie dort die aus ihrer Sicht relevanten Informationen zum Projekt. Dies ist mindestens das Produkt-Statement. Hinterlegen Sie eventuell weitere Informationen im Wiki-Bereich.
4. Legen Sie im Bereich „Issues“ Meilensteine entsprechend ihrer Projektplanung an.
5. Legen Sie für jede Aufgaben aus ihrer Stückliste einen entsprechenden Issue an und ordnen Sie den Issue entsprechend der Planung einem Meilenstein zu.
6. Legen Sie ein Label „current increment“ an und markieren Sie alle Issues, die Sie im kommenden Sprint bearbeiten wollen mit diesem Label.
7. Optional: Legen Sie pro Thema oder Funktion ebenfalls Labels an und ordnen Sie die Issues entsprechend zu. Machen Sie dies aber nur dann, wenn es der Übersichtlichkeit dient.

Hinweis: Die folgende Website bietet eine kurze interaktive Einführung in das Versionskontrollsystem git (<http://try.github.io/levels/1/challenges/1>). Recherchieren Sie bei Bedarf weitere Quellen im Internet, mit deren Hilfe Sie sich in git einarbeiten können.

**Viel Spaß!**