1. Einleitung

2. Funktionsumfang

2.1. Mindestanforderungen Funktionsumfang

! Es müssen **alle** Must-Kriterien Erfüllt werden!

Folgende Klassen und deren dazugehörigen ViewModels (soweit in den Desing Klassenmodell vorhanden) müssen Vollständig Implementiert werden:

- Process
- RunningProcess
- Task
- User
- Department
- Contract
- Role
- Administrator
-

3. Tests und Reviews

3.1. Code Review

Jede Codeänderung muss bevor sie zum Projekt hinzugefügt wird einen Reviewprozess durchlaufen, wobei das Review immer von einer Person (Reviewer) durchgeführt werden muss die selbst nicht Autor des Codes ist.

Das Review wird zuerst vom Reviewer ohne Beisein des Autors durchgeführt und dann die Ergebnisse und Fragen mit dem Autor besprochen

Dieser besteht aus den Folgenden Komponente, die Zeitliche Abfolge und ineinander Verschachtelung der einzelnen Komponenten ist hierbei nicht festgelegt

- Code Style Review:
  Der Code wird vom Reviewer darauf überprüft ob er dem Styleguide des Projekts entspricht.
  Beispiel:
  ist die Variablenbenennung sinnvoll, sind die Funktionen korrekt Benannt
  ist der Sytle übereinstimmend mit dem Rest des Projekts

  Zusätzlich wird überprüft ob der Code vielleicht Formal richtig, jedoch Komplexer oder unübersichtlicher gestaltet ist als notwendig
  Beispiel:
  Lange Einzeiler sind unübersichtlicher wie gut strukturierter Code über mehrere Zeilen
  Komplexe Aufrufe aufteilen und in Variablen zwischenspeichern wenn sie öfter aufgerufen werden
- Dokumentation Review:
  Der Reviewer prüft ob der Code ausreichen Dokumentiert ist
  Dafür sollte der Reviewer ohne Erklärungen des Autors den Code in seiner grundlegen Funktion verstehen können.
- Funktionales Code Review:
  Der Code wird vom Reviewer darauf überprüft ob er theoretisch die auch das tut was der Autor damit erzielen wollte.
  Beispiele:
  Sind Fehler im Code, weshalb er so nicht das gewünschte Ergebnis liefern kann
  ist der Code so in die Code Base integrierbar, oder sind vielleicht Variablen und Funktionen von anderen Codeteilen nicht richtig übernommen worden

  Auch wird überprüft ob die Ziele des Codes mit den Anforderungen an ihn Übereinstimmen
  Beispiele:
  hat der Autor vielleicht missverstanden was sein Code erreichen soll?

3.2. Anforderungen an den Reviewer

Der Reviewer sollte den Code vor beginn des Reviews möglichst wenig gelesen haben und darf ihn nicht selbst geschrieben haben.

Als Reviewer ist die Person zu wählen die die höchstmögliche Expertise in den vom Code abgedeckten Bereich hat

Es ist zu vermeiden das sich „Review-Pärchen" bilden.
D.h. das zwei Personen konstant den Code des jeweiligen anderen Reviewen

Die Review-Last ist möglichst gleichmäßigt auf die Teammitglieder zu verteilen

3.3. Leitfaden zur Feststellung der Testart und Notwendigkeit

3.3.1. Wann sind Unit Test für den Code notwendig?

3.3.1.1. Wann sind Unit Tests nicht notwendig?

- Für „temporären" Code der nicht in die Produktion Codebase überführt werden soll
- Für Code bei dem noch viele Änderungen erwartend werden, welche erfordern das der Test sehr oft neu geschrieben werden müsste
- Für Code der mit externen Systemen Arbeitet, also Datenbankanbindungen, Netzwerkdienste, Userinputs, etc.
- Für Code der selbst keine Logik beinhaltet außer das Aufrufen von anderem bereits getestetem Code (z.B. ruft eine Funktion nur die Funktion einer anderen Klasse auf und gibt die erhaltenen Werte unverändert als ihren Rückgabewert zurück)

3.3.1.2. Wann sind Unit Test notwendig?

Wenn er keine der Kriterien aus „*Wann sind Unit Tests **nicht** notwendig?*" erfüllt und mindestens einen der nachfolgenden Punkte

- Er ist komplexer als ein paar wenige Codezeilen mit einfachen Befehlen
- Er hat eigene Logik die nicht schon durch einen Vorausgehenden Test getestet wurde.
- Er kombiniert Logik aus Code der noch nicht getestet Wurde (einfacher Code wird in diesem Code zu komplexerer Funktion kombiniert)
- Für Sicherheitsrelevante Abläufe, wie zum Beispiel der Überprüfung von Zugriffsberechtigungen

Sollte ein Unit Test nicht bestanden werden, ist eine Tiefere Code Analyse und auch differenziertes Testen der einzelne Codeabschnitte notwendig.

3.3.2. Wann sind E2E Tests notwendig?

- Für sämtliche Veränderungen von Datensätzen in der Datenbank die von der UI aus gemacht werden können
- Für die Überprüfung von Zugriffsberechtigungen