

Extreme delivery

Test Strategy

Revision History

Date	Version	Author	Description

1. Scope

It defines parameters like

Who will review the document?

- Kim, Steffen & Hold B

Who will approve this document?

- Signe (Scrum Master/Dronning)
- Kim & Steffen

Testing activities carried out with timelines

- Unit test efter hvert sprint på 14 dage.
- En gang i måneden laver vi integrationstest og acceptance test.
- Unit test skal bestå hver gang der merges ind i dev og main

- Vi kører ikke TDD, da projektets størrelse er for småt og tiden til at lave det ikke er der.
- Vi har PO (**product owner** til at deltage i sprint review hver fredag, samt til daily standup hvis det bliver nødvendigt)
- Scrum Master er med til alle daily standup, sprint review og sprint retrospective
- Projektleder har ansvaret for et team
- Programleder har ansvaret for interessenthåndtering og at alle projektlederne "løber i samme retning"

2. Test Approach

It defines

- **Process of testing**
 - Skriv kode og test før der integreres
- **Testing levels**
 - Vi tester på alle niveauer i test Pyramiden
- **Roles and responsibilities of each team member**
 - Udviklere
 - Geograf (GIS)
 - Lead udvikler
 - Projektleder
 - Programleder
 - Test personer
 - UI & UX
 - Database Manager
 - Supply Chain Management
- **Types of Testing (Load testing, Security testing, Performance testing etc.)**
 - All of them

- **Testing approach & automation tool if applicable**

Structural testing -> unit test.

Og front-end testing -> brugertest

- **Adding new defects, re-testing, Defect triage, Regression Testing and test sign off**

- Vi tager modellen fra hjemme-is og forbedre og tilføjere
-

3. Test Environment

- **Define number of requirement and setup required for each environment**

- Udviklerne sidder i VS Code og skriver unit-test i test directory.
- Vi bruger GitHub Actions is a CI/CD platform til at automatisere unit test når der pushes til Dev og Main branches.

- **Define backup of test data and restore strategy**

- Vi har en original-database og laver kopier (redundans i db) af vi altid kan gå tilbage til den originale ved fejl.

4. Testing Tools

- **Automation and Test management tools needed for test execution**

- Github Actions - CI/CD tools
- JUnit python versionen

- **Figure out number of open-source as well as commercial tools required, and determine how many users are supported on it and plan accordingly**

- commercial (GitHub actions)

- open source (JUnit)

5. Release Control

- **Release management plan with appropriate version history that will make sure test execution for all modification in that release**
 - Første release kommer når alle test er bestået og derefter version styres der via GitHub. Ny release kommer automatisk så bruger skal “bare” opdatere.

6. Risk Analysis

- **List all risks that you can estimate**
 - hackerangreb og data breach
 - sygdom
 - barsel
 - direkte konkurrence
 - Prisstigninger hos cloud provider
 - regulering fra landets lovgivende forsamling (øv)
- **Give a clear plan to mitigate the risks also a contingency plan**
 - Vi kan lave red team/ blue team test for at sikre så godt som muligt imod hacker angreb.
 - Vi kan skabe et godt arbejdsmiljø og gode vilkår der forebygger sygdom. Herunder sundhedsforsikring, ergonomiske møbler, stress coach, 4-dages arbejdsuge.
 - Vi kan hyre mennesker nok, så vi ikke er sårbare ift barsel. Samt give gode barselsvilkår så vi kan tiltrække kvindelige talenter.
 - ift direkte konkurrence - Blitzscaling (mange investorer og hurtig markedsdominans)
 - Vi bruger mange forskellige cloud providers, sådan at vi ikke er sårbare overfor prisstigninger hos en enkelt af dem.
 - Når vi er store nok, skal der laves seriøst lobbyarbejde.

7. Review and Approvals

- All these activities are reviewed and sign off by the business team, project management, development team, etc.
- Summary of review changes should be traced at the beginning of the document along with approved date, name, and comment