

Software Lifecycle Management

Maintenance Monitor

We are a small hydro-power electricity supplier near Vienna. Our customers expect electricity around the clock with a service level agreement of 99.95%. This means that the maximum outage of [21 minutes and 54 seconds](#) (monthly in sum) is tolerated. It is easy to derive that service times are very important to us. Huge monitors were installed that should show

- a green monitor in case everything looks fine and
- a red monitor in case of problems.

All monitors show the same content: a maximized website.

Project

A team of max 3 members should implement a REST-based server in Java (use [Spring Boot](#)). The service should be able to manage a centrally stored message and hereby capable to:

- reset the message
- set it to a specific message
- deliver the message to the clients

using REST.

Create a web frontend which is capable to query the message every 5 seconds.

Requirements

Use GitHub or Azure DevOps for the project and follow the correct DevOps procedure. Use a Kanban board to manage your User Stories and use a git branching model (preferable gitflow) to manage your code. Track your development process by updating your Kanban board and write at least one unit tests for every requirement. A Continuous Integration pipeline that produces the finished software artifact should be implemented as well.

Document

- the whole process
- the user stories
- the repository URL
- the usage of the software

in a PDF file with screenshots and explanatory text. Submit the code (including the .git folder and ALM files) as a zip file (please put the PDF inside the zip file).

You can use external resources as long as you mark them: “ // taken from: <URL> ”

Points

- Documentation of the process: 15%
- Requirement definitions (User Stories): 15%
- Correct status / Linking / Branching (Kanban, Git): 15%
- Implementation: 15%
- Testing: 15%
- Pipeline (Continuous Integration and Maven): 15%
- Artefacts (Continuous Delivery): 10%

All elements must be present in the documentation.

References

