Hochschule für Telekommunikation Leipzig

Software Engineering Specification-UX Design

Kaab Rustam

Supervisor: **Professorin Dr. Ing. Sabine Radomski**

A report submitted in fulfilment of the requirements for the module
Software Engineering Specification-UX Design

in the Degree Program of

International Masters of Engineering in ICT

Topic: *UX Design of "Embold" in the light of User Centred Design (UCD)*

January 29, 2020

***"Design is not just what it looks like and feels like. Design is how it works"*- Steve Jobs**

Contents

1 Analysis **3**

1.1 Context of Use . . . 3

1.2 User Group Profiles . . . 4

1.3 Task Model . . . 4

1.4 Scenarios . . . 4

1.5 Persona . . . 4

2 User Requirements **5**

2.1 User Needs . . . 5

2.2 User Story . . . 5

2.3 User Requirement . . . 5

3 Product Design **6**

3.1 Information Architecture . . . 6

3.2 Style Guide - Comparison . . . 6

4 Evaluate the Design **7**

4.1 Test Cases . . . 7

4.2 Test tools & Metrics . . . 7

4.3 Usability Test Report . . . 7

5 Conclusion **8**

5.1 Summary . . . 8

5.2 Recomendation for Improvements . . . 8

5.3 Lesson Learned . . . 8

Bibliography **8**

List of Figures

Chapter 1

Analysis

1.1 Context of Use

Where there is code, there are bugs for sure. There are various tools and methods to test & s the code before it goes into the real time environment. It is very fruitful if code analysis is done along with the traditional testing of the code, as it assists the enterprises to find the potential risks, vulnerabilities & other weaknesses in the source code. If a vulnerability remains undetected it can cause a severe problem if its a multi-tiered system, as it will become a big deal to in both the aspects i.e time and budget. A well organised code analysis will prevent the enterprises from such troubles well before the time. The traditional testing methods are not resilient enough to detect all the vulnerabilities and the threats in a multi-tiered or multiple tech infrastructures. If enterprises only rely on such traditional methods they are potentially on a risk to face:

- Business Disruptions
- High Maintenance Cost
- Failure to meet standard
- Poor Performance
- Integrity on stake

and many more issues. Every missed vulnerability is potent enough to demolish the enterprise. Code analysis software are used to fix the dugs when they can be fixed cheaply, as its a static analysis method hence a code is analysed before it goes into the execution mode.

Dynamic code analysis is another code analysis method but it mainly has focus on the output rather than detecting the issues in a source code. So for that purpose static code analysis is used. Static code analysis tools/softwares use many complex programming principles to detect the complexity, evaluate the size and detection of potent vulnerabilities inside the source code.

There are plethora of the tools used for static code analysis and market has a variety of such tools available. I will focus on “Embold”. It is a proprietary software application used to analyze, diagnose & sustain the source code. Embold works right in the development process and connects all the stake holders around the code. It comes with both on premise and cloud solution. The purpose of Embold mainly is to get a quality solution and remove the issues in the code before

they become the roadblocks; it is also used to determine the code smells in order to remove the vulnerabilities and get an error free, perfectly functional and standard code.

It is not only designed to help the developers, but it helps significantly to project managers and architects as well. It helps with variety of characteristics like quality metrics, design issues, duplication in code, hotspot/bottlenecks etc. It uses AI techniques to examine the repositories and come up with all the parameters.

1.2 User Group Profiles

The users can be classified in three various categories depending on their role in the SDLC.

- Developers
- Architect
- Project Managers

Each of the role mentioned above has its own task and responsibilities and targets to achieve, let's see how the static code analysis works for each of them.

Developers:

1.3 Task Model

1.4 Scenarios

1.5 Persona

Chapter 2

User Requirements

2.1 User Needs

2.2 User Story

2.3 User Requirement

Chapter 3

Product Design

3.1 Information Architecture

3.2 Style Guide - Comparison

Chapter 4

Evaluate the Design

4.1 Test Cases

4.2 Test tools & Metrics

4.3 Usability Test Report

Chapter 5

Conclusion

5.1 Summary

5.2 Recomendation for Improvements

5.3 Lesson Learned

Bibliography

[1] Ux-design. https://www.interaction-design.org/literature/topics/ux-design.