Software Requirements Specification

**Version: Draft**

**Project KIWI**

**Team Tango**

**Revision Chart**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Primary Author(s)** | **Description of Version** | **Date Completed** |
| Draft | Andreas | Initial draft created for distribution and review comments | 10/3/2020 |
| Preliminary | TBD | Second draft incorporating initial review comments, distributed for final review | TBD |
| Final | TBD | First complete draft, which is placed under change control | TBD |
| Revision 1 | TBD | Revised draft, revised according to the change control process and maintained under change control | TBD |
| Revision 2 | TBD | Revised draft, revised according to the change control process and maintained under change control | TBD |
| etc. | TBD | TBD | TBD |

**Contents**

**1.** **Introduction**

1.1 Purpose

1.2 Scope

1.3 Definitions, Acronyms, and Abbreviations

1.4 References

**2.** **Overall Description**

2.1 Product Perspective

*2.1.1* *System Interfaces*

*2.1.2* *User Interfaces*

*2.1.3* *Hardware Interfaces*

*2.1.4* *Software Interfaces*

*2.1.5* *Communications Interfaces*

*2.1.6* *Memory Constraints*

*2.1.7* *Site Adaptation Requirements*

2.2 User Characteristics

2.3 Constraints

2.4 Assumptions and Dependencies

**3.** **Specific Requirements**

3.1 External Interface Requirements

*3.1.1* *User Interfaces*

*3.1.2* *Hardware Interfaces*

*3.1.3* *Software Interfaces*

*3.1.4* *Communications Interfaces*

3.2 Software Product Features

*3.2.1* *Feature 1*

3.3 Performance Requirements

3.4 Software System Attributes

*3.4.1* *Reliability*

*3.4.2* *Availability*

*3.4.3* *Security*

*3.4.4* *Maintainability*

*3.4.5* *Portability*

3.5 Logical Database Requirements

3.6 Other Requirements

**4.** **Appendices**

# Introduction

When it comes to developer tools, feedback time can make a drastic difference in both development speed, and for the development experience itself. Tools that can make a measurable difference here can save valuable development time. They are also easily marketable because one off cost of the tool can quickly be recouped in the form of improved development efficiency.

## Purpose

The purpose of the Kiwi project is first and foremost to create a tool that facilitates both development efficiency as well as enjoyment in the realm of frontend Javascript unit testing. The purpose of the project foster this, and while doing so, contribute to the ecosystem of the “Kakoune” editor specifically.

## Scope

The project scope is deliberately quite conservative. The first iteration of the testing tool will

only support a highly opinionated setup. As stated in the purpose, the editor plugin will only support the Kakoune code editor. Also, the tool will work specifically with the “Webpack” build tool and with the “Chai” unit testing toolkit. The core feature of the project is to run unit tests and print results next to the relevant code snippet in the editor. Additional features such as coverage reports will be added depending on how fast development of the project progresses.

## Definitions, Acronyms, and Abbreviations

Kiwi – An interactive unit test runner for the Kakoune editor

Kakoune – A Vim clone text/code editor essentially

Webpack – The most popular Javascript build tool

Chai – Unit testing helper functions

## References

https://github.com/stoand/kiwi-webpack-plugin

https://www.chaijs.com/guide/

# Overall Description

## Product Perspective

### System Interfaces

-

### User Interfaces

The Webpack plugin allows users to configure the tool and include it into their projects.

### Hardware Interfaces

*-*

### Software Interfaces

* Chrome Runner CR 1.3.5 https://www.npmjs.com/package/chrome-runner
* Webpack WP 4.0 https://webpack.js.org/concepts/
* Kakoune KA v2019.07.01 <https://github.com/mawww/kakoune>

### Communications Interfaces

-

### Memory Constraints

-

### Site Adaptation Requirements

-

## User Characteristics

The target user demographic will be developers experienced with using Kakoune as

well as writing Javascript unit tests.

## Constraints

-

## Assumptions and Dependencies

-

# Specific Requirements

## External Interface Requirements

### User Interfaces

### Hardware Interfaces

### Software Interfaces

### Communications Interfaces

## Software Product Features

Please read the spec at <https://kiwi-spec.netlify.com/>

## Performance Requirements

The tool should on average respond in under half a second for medium sized projects.

## Software System Attributes

### Reliability

The tool needs to avoid crashing or otherwise becoming unresponsive.

### Availability

-

### Security

-

### Maintainability

The code should be written in a way so that it is easily testable.

Data transformation functionality should be covered by tests.

### Portability

The tool needs to support all platforms that Kakoune supports. That is, Linux and Mac; not Windows.

## Logical Database Requirements

-

## Other Requirements

-

# Appendices

-