**Text adventure – The Little Mermaid**

|  |  |
| --- | --- |
| IPA-Daten |  |
| Firma | enersis suisse AG |
| Abteilung | Development/ IT |
| Autor | Nicole Sager |
| Ausgabedatum | 10.05.2017 |
| Version | V1.0 |
| Status | Genehmigt, zur Nutzung |

|  |  |
| --- | --- |
| Beteiligter Personenkreis | |
| In der Genehmigung | -- |
| Valid-Experte | Ramun Hofmann |
| In der Durchführung | Nicole Sager |
| Fachvorgesetzter | (Nora Kleisli) |
| Tech. Fachvorgesetzter | -- |
| Hauptexperte | Ramun Hofmann |
| Zweit-Experte | Vanessa Meister |
| Berufsbilder | Roland Dardel |

IPA 2017, Kt. Bern

**Dokumentinformationen**

Änderungskontrolle, Prüfung, Genehmigung

|  | | | |
| --- | --- | --- | --- |
| Version | Datum | Name | Beschreibung |
| Vorlage | 23.06.2013 | A. Mueller | Dokumentvorlage QV2013, Version V1.0 |
| X0.01 | 26.04.2017 | N. Sager | Teil 1: Aufgabenstellung, Projektorganisation, Organisation der IPA |
| X0.02 | 27.04.2017 | N. Sager | Teil 2: Initialisierungsphase, Ist-Analyse |
| X0.03 | 28.04.2017 | N. Sager | Teil 2: Initialisierungsphase, ISDS, Lösungssuche, Risikoanalyse, Variantenentscheid |
| X0.04 | 01.05.2017 | N. Sager | Teil 3: Initialisierungsphase, Optimierungen  Teil 2: Konzeptphase, Systemarchitektur |
| X0.05 | 02.05.2017 | N. Sager | Teil 2: Konzeptphase, Systemarchitektur, Klassendiagramm, Machbarkeitsprüfung, |
| V1.0 | 10.05.2017 | N. Sager | Abnahme des Projektes |

**Verwendete Abkürzungen**

| Abkürzung | Bedeutung |
| --- | --- |
| bzw. | beziehungsweise |
| FV | Fachvorgesetzter |
| IPA | Individuelle praktische Arbeit |
| ISDS | Informationssicherheit und Datenschutz |
| OdA | Organisation der Arbeitswelt |
| QV | Qualifikationsverfahren |
| vgl. | vergleiche |
| VZ | Verzeichnis |

**Inhaltsverzeichnis**

[1. Aufgabenstellung 2](#_Toc482180126)

[1.1 Titel der Facharbeit 2](#_Toc482180127)

[1.2 Thematik 2](#_Toc482180128)

[1.3 Ausgangslage 2](#_Toc482180129)

[1.4 Detaillierte Aufgabenstellung 2](#_Toc482180130)

[1.5 Mittel und Methoden inklusive Projektmethode 3](#_Toc482180131)

[1.6 Vorkenntnisse 3](#_Toc482180132)

[1.7 Vorarbeiten 4](#_Toc482180133)

[1.8 Neue Lerninhalte 4](#_Toc482180134)

[1.9 Arbeiten des Kandidaten im Schwerpunkt während dem 3. und 4. Lehrjahr 4](#_Toc482180135)

[2. Detaillierte Projektmethode 5](#_Toc482180136)

[2.1 Projektmethode 5](#_Toc482180137)

[2.2 Szenario 5](#_Toc482180138)

[2.3 Phasen 7](#_Toc482180139)

[2.4 Module 7](#_Toc482180140)

[2.5 Projektorganisation 9](#_Toc482180141)

[2.5.1 Projektrollen 9](#_Toc482180142)

[3. Zeitplan 10](#_Toc482180143)

[4. Organisation der IPA 12](#_Toc482180144)

[4.1 Infrastruktur 12](#_Toc482180145)

[4.2 Datensicherung der IPA 12](#_Toc482180146)

[4.3 Ordnerstruktur 13](#_Toc482180147)

[5. Firmenstandards 15](#_Toc482180148)

[6. Arbeitsjournal 16](#_Toc482180149)

[6.1 Erster halber Tag: Mittwoch, 26.04.2017 17](#_Toc482180150)

[6.2 Zweiter Tag: Donnerstag, 27.04.2017 19](#_Toc482180151)

[6.3 Dritter Tag: Freitag, 28.04.2017 21](#_Toc482180152)

[6.4 Vierter Tag: Montag, 01.05.2017 23](#_Toc482180153)

[6.5 Fünfter Tag: Dienstag, 02.05.2017 25](#_Toc482180154)

[6.6 Sechster Tag: Mittwoch, 03.05.2017 27](#_Toc482180155)

[6.7 Siebter Tag: Donnerstag, 04.05.2017 29](#_Toc482180156)

[6.8 Achter Tag: Freitag, 05.05.2017 30](#_Toc482180157)

[6.9 Neunter Tag: Montag, 06.05.2017 32](#_Toc482180158)

[6.10 Zehnter Tag: Dienstag, 07.05.2017 34](#_Toc482180159)

[7. Projektjournal 35](#_Toc482180160)

[8. Abschlussbericht 36](#_Toc482180161)

[8.1 Vergleich Ist/Soll 36](#_Toc482180162)

[8.2 Mittelbedarf 37](#_Toc482180163)

[8.3 Realisierungsbericht 37](#_Toc482180164)

[8.3.1 Ungeplante Ereignisse 37](#_Toc482180165)

[8.3.2 Erkenntnisse der Realisierung 37](#_Toc482180166)

[8.4 Testbericht 38](#_Toc482180167)

[8.5 Fazit zum IPA (Projekt) 38](#_Toc482180168)

[8.6 Persönliches Fazit 38](#_Toc482180169)

[8.7 Schlussreflexion 39](#_Toc482180170)

[9. Unterschriften Teil 1 40](#_Toc482180171)

[Teil 2: Projektdokumentation 41](#_Toc482180172)

[10. Initialisierung 42](#_Toc482180173)

[10.1 Studie; Ist-Zustand 42](#_Toc482180174)

[10.1.1 Das greenited Projekt 42](#_Toc482180175)

[10.1.2 Barchart 43](#_Toc482180176)

[10.1.3 Donutchart 47](#_Toc482180177)

[10.2 Meilensteine 50](#_Toc482180178)

[10.2.1 SMART 50](#_Toc482180179)

[10.2.2 Persönliche Vorgehensziele 50](#_Toc482180180)

[10.2.3 Projektziele 51](#_Toc482180181)

[10.3 Anforderungen 52](#_Toc482180182)

[10.3.1 Funktionale Anforderungen 52](#_Toc482180183)

[10.3.2 Nicht funktionale Anforderungen 52](#_Toc482180184)

[10.4 Informationssicherheit und Datenschutz (ISDS) 53](#_Toc482180185)

[10.5 Machbarkeitsprüfung Initialisierungsphase 53](#_Toc482180186)

[10.6 Risikoanalyse 57](#_Toc482180187)

[10.7 Risikographen 58](#_Toc482180188)

[10.7.1 Risikograph vor den Massnahmen 58](#_Toc482180189)

[10.7.2 Risikograph nach den Massnahmen 58](#_Toc482180190)

[10.7.3 Kurze Stellungnahmen zu den Risiken 58](#_Toc482180191)

[10.8 Lösungen suchen 59](#_Toc482180192)

[10.8.1 Aufsetzen der Chart Library 59](#_Toc482180193)

[10.8.2 Aufruf 59](#_Toc482180194)

[10.9 Varianten 60](#_Toc482180195)

[10.9.1 Gewichtung der Kriterien 60](#_Toc482180196)

[10.9.2 React Redux Starter Kit 61](#_Toc482180197)

[10.9.3 ES6 Karma Jasmine Webpack Boilerplate 61](#_Toc482180198)

[10.9.4 Webpack library starter 62](#_Toc482180199)

[10.10 Variantenentscheid 63](#_Toc482180200)

[10.10.1 Begründung 63](#_Toc482180201)

[11. Konzept 64](#_Toc482180202)

[11.1 Allgemein 64](#_Toc482180203)

[11.1.1 Grundlagen 64](#_Toc482180204)

[11.1.2 ISDS 64](#_Toc482180205)

[11.2 Architektur 65](#_Toc482180206)

[11.2.1 Vom Source Code zur fertigen Library 65](#_Toc482180207)

[11.2.2 Einbindung der Charts auf einer Webseite 66](#_Toc482180208)

[11.2.3 Klassendiagramm 67](#_Toc482180209)

[11.2.4 Datenstruktur 68](#_Toc482180210)

[11.3 Anwendungsfall 68](#_Toc482180211)

[11.4 Machbarkeitsprüfung Konzeptphase 69](#_Toc482180212)

[11.5 Testkonzept 70](#_Toc482180213)

[11.5.1 Grundlagen 70](#_Toc482180214)

[11.5.2 Test Szenario 70](#_Toc482180215)

[11.5.3 Testrahmen 70](#_Toc482180216)

[11.5.4 Testvorgehen 70](#_Toc482180217)

[11.5.5 Testmethode 70](#_Toc482180218)

[11.5.6 Testfälle 71](#_Toc482180219)

[11.5.7 Testziele 73](#_Toc482180220)

[12. Realisierung 74](#_Toc482180221)

[12.1 Template 74](#_Toc482180222)

[12.1.1 ECMAScript 6 74](#_Toc482180223)

[12.1.2 Babel 75](#_Toc482180224)

[12.1.3 ESLint 75](#_Toc482180225)

[12.1.4 Unit Testing 76](#_Toc482180226)

[12.1.5 Vom Source Code zur fertigen Library 77](#_Toc482180227)

[12.2 Klassendiagramm 79](#_Toc482180228)

[12.3 Entwicklung 80](#_Toc482180229)

[12.3.1 ensChart Klasse 80](#_Toc482180230)

[12.3.2 Ausgelagerte Klassen 82](#_Toc482180231)

[12.4 Testprotokoll 83](#_Toc482180232)

[12.4.1 Constructor 84](#_Toc482180233)

[12.4.2 Title 85](#_Toc482180234)

[12.4.3 Legend 86](#_Toc482180235)

[12.4.4 DurationTime 87](#_Toc482180236)

[12.4.5 Color Klasse 88](#_Toc482180237)

[12.4.6 Ease Klasse 90](#_Toc482180238)

[12.4.7 Scale Klasse 91](#_Toc482180239)

[12.4.8 Size Klasse 92](#_Toc482180240)

[12.4.9 Typ Klasse 94](#_Toc482180241)

[12.5 Vergleich der Anforderung und der Umsetzung 95](#_Toc482180242)

[12.5.1 Funktionale Anforderungen 95](#_Toc482180243)

[12.5.2 Nicht funktionale Anforderungen 95](#_Toc482180244)

[12.6 API Dokumentation vorbereiten 96](#_Toc482180245)

[12.6.1 Struktur der API Dokumentation 96](#_Toc482180246)

[13. API documentation ensChart 97](#_Toc482180247)

[General Information 97](#_Toc482180248)

[Getting started 97](#_Toc482180249)

[Initializing the Library 97](#_Toc482180250)

[The different possible parameters you can set 97](#_Toc482180251)

[Lib.title(arg) 97](#_Toc482180252)

[Lib.data(arg) 98](#_Toc482180253)

[Lib.legend(arg) 98](#_Toc482180254)

[Lib.color(arg) 98](#_Toc482180255)

[Lib.ease(arg) 98](#_Toc482180256)

[Lib.durationTime(arg) 99](#_Toc482180257)

[Lib.scale(arg) 99](#_Toc482180258)

[Lib.width(arg) 99](#_Toc482180259)

[Lib.height(arg) 99](#_Toc482180260)

[Lib.type(arg) 99](#_Toc482180261)

[If something doesn’t work 100](#_Toc482180262)

[If you want to include a new Chart into the Library 100](#_Toc482180263)

[14. Literatur und Quellenverzeichnis 101](#_Toc482180264)

[15. Glossar 102](#_Toc482180265)

[16. Unterschriften für Abnahme 104](#_Toc482180266)

[17. Anhang 105](#_Toc482180267)

[17.1 Entwickler Richtlinien 105](#_Toc482180268)

[1. Einleitung 107](#_Toc482180269)

[1.1. Ziel und Zweck dieses Dokuments 107](#_Toc482180270)

[1.2. Vorschriften und Standards 107](#_Toc482180271)

[2. Allgemeine Richtlinien und Verantwortlichkeiten 107](#_Toc482180272)

[2.1. Zuständigkeiten 107](#_Toc482180273)

[2.2. Technologische Verantwortlichkeiten 108](#_Toc482180274)

[3. Projektmanagement 108](#_Toc482180275)

[3.1. Methodik 108](#_Toc482180276)

[3.2. Projektmanagement Tool 109](#_Toc482180277)

[4. Softwareumgebung 110](#_Toc482180278)

[5. Entwicklungsrichtlinien 110](#_Toc482180279)

[5.1. Allgemein 110](#_Toc482180280)

[5.2. Grids 111](#_Toc482180281)

[5.3. Browserkompatibilität 111](#_Toc482180282)

[5.4. Coding Style 111](#_Toc482180283)

[5.4.1. Allgemein 111](#_Toc482180284)

[5.4.2. C# 112](#_Toc482180285)

[5.4.3. JavaScript 113](#_Toc482180286)

[5.4.4. PL-SQL 114](#_Toc482180287)

[6. Dokumentation 116](#_Toc482180288)

[17.2 Protokolle 117](#_Toc482180289)

[17.2.1 Erster Besuchstag 117](#_Toc482180290)

[17.2.2 zweiter Besuchstag 120](#_Toc482180291)

[1 Ablauf 121](#_Toc482180292)

[2 Wichtige Punkte und Entscheidungen 121](#_Toc482180293)

[17.3 Source Code 122](#_Toc482180294)

[17.3.1 index.js 122](#_Toc482180295)

[17.3.2 color.config.js 128](#_Toc482180296)

[17.3.3 ease.config.js 129](#_Toc482180297)

[17.3.4 scale.config.js 130](#_Toc482180298)

[17.3.5 size.config.js 131](#_Toc482180299)

[17.3.6 type.config.js 131](#_Toc482180300)

[17.3.7 main.chart.js 133](#_Toc482180301)

[17.3.8 donut.chart.js 135](#_Toc482180302)

[17.3.9 pie.chart.js 135](#_Toc482180303)

[17.3.10 bar.chart.js 138](#_Toc482180304)

[17.3.11 horizontal.js 139](#_Toc482180305)

[17.3.12 vertical.js 140](#_Toc482180306)

[17.3.13 index.html 141](#_Toc482180307)

[17.3.14 library.spec.js 142](#_Toc482180308)

[17.3.15 color.spec.js 145](#_Toc482180309)

[17.3.16 ease.spec.js 148](#_Toc482180310)

[17.3.17 scale.spec.js 149](#_Toc482180311)

[17.3.18 size.spec.js 150](#_Toc482180312)

[17.3.19 type.spec.js 154](#_Toc482180313)

[17.3.20 ensChart.js & ensChart.js.map 155](#_Toc482180314)

**Abbildungsverzeichnis**

[Abbildung 1: Detaillierte Projektmethode 5](file:///D:\nicol\Dropbox%20(enersis)\55%20Shared%20Services\Office%20IT\03_Projekte\Projekt_IPA_SNI\IPA_Nicole_2017\02_Dokumentation\11_Tag\Entwicklung%20einer%20Chart%20Library%20zur%20internen%20Verwendung_V1.0.docx#_Toc482180315)

[Abbildung 2: Szenarioleitfaden 6](#_Toc482180316)

[Abbildung 3: Module die für das Projekt verwendet wurden 8](#_Toc482180317)

[Abbildung 4: Projektorganisation 9](file:///D:\nicol\Dropbox%20(enersis)\55%20Shared%20Services\Office%20IT\03_Projekte\Projekt_IPA_SNI\IPA_Nicole_2017\02_Dokumentation\11_Tag\Entwicklung%20einer%20Chart%20Library%20zur%20internen%20Verwendung_V1.0.docx#_Toc482180318)

[Abbildung 5: Zeitplan 10](#_Toc482180319)

[Abbildung 6: Legende Zeitplan 11](#_Toc482180320)

[Abbildung 7: Arbeitsplatz 12](#_Toc482180321)

[Abbildung 8: Ordnerstruktur 13](file:///D:\nicol\Dropbox%20(enersis)\55%20Shared%20Services\Office%20IT\03_Projekte\Projekt_IPA_SNI\IPA_Nicole_2017\02_Dokumentation\11_Tag\Entwicklung%20einer%20Chart%20Library%20zur%20internen%20Verwendung_V1.0.docx#_Toc482180322)

[Abbildung 9: Ordner 02\_Dokumentation 14](file:///D:\nicol\Dropbox%20(enersis)\55%20Shared%20Services\Office%20IT\03_Projekte\Projekt_IPA_SNI\IPA_Nicole_2017\02_Dokumentation\11_Tag\Entwicklung%20einer%20Chart%20Library%20zur%20internen%20Verwendung_V1.0.docx#_Toc482180323)

[Abbildung 10: Inhalt des ersten Tagesordners 14](#_Toc482180324)

[Abbildung 11: Unterschrift Teil 1 40](#_Toc482180325)

[Abbildung 12: greenited Projekt - So funktioniert es 42](#_Toc482180326)

[Abbildung 13: greenited Projekt - Webseite 43](#_Toc482180327)

[Abbildung 14: greenited Projekt - Stacked Bar Chart 43](file:///D:\nicol\Dropbox%20(enersis)\55%20Shared%20Services\Office%20IT\03_Projekte\Projekt_IPA_SNI\IPA_Nicole_2017\02_Dokumentation\11_Tag\Entwicklung%20einer%20Chart%20Library%20zur%20internen%20Verwendung_V1.0.docx#_Toc482180328)

[Abbildung 15: HTML DIV-Element Barchart 44](#_Toc482180329)

[Abbildung 16: Donutchart «Energiemix regional» 47](file:///D:\nicol\Dropbox%20(enersis)\55%20Shared%20Services\Office%20IT\03_Projekte\Projekt_IPA_SNI\IPA_Nicole_2017\02_Dokumentation\11_Tag\Entwicklung%20einer%20Chart%20Library%20zur%20internen%20Verwendung_V1.0.docx#_Toc482180330)

[Abbildung 17: HTML-DIV-Element Donutchart 47](#_Toc482180331)

[Abbildung 18: Risikograph 58](file:///D:\nicol\Dropbox%20(enersis)\55%20Shared%20Services\Office%20IT\03_Projekte\Projekt_IPA_SNI\IPA_Nicole_2017\02_Dokumentation\11_Tag\Entwicklung%20einer%20Chart%20Library%20zur%20internen%20Verwendung_V1.0.docx#_Toc482180332)

[Abbildung 19: Risikograph 58](file:///D:\nicol\Dropbox%20(enersis)\55%20Shared%20Services\Office%20IT\03_Projekte\Projekt_IPA_SNI\IPA_Nicole_2017\02_Dokumentation\11_Tag\Entwicklung%20einer%20Chart%20Library%20zur%20internen%20Verwendung_V1.0.docx#_Toc482180333)

[Abbildung 20: Bewertungsmöglichkeiten 60](#_Toc482180334)

[Abbildung 21: Einbindung der Charts auf einer Webseite 66](#_Toc482180335)

[Abbildung 22: Klassendiagramm 67](#_Toc482180336)

[Abbildung 23: Datenstruktur 68](file:///D:\nicol\Dropbox%20(enersis)\55%20Shared%20Services\Office%20IT\03_Projekte\Projekt_IPA_SNI\IPA_Nicole_2017\02_Dokumentation\11_Tag\Entwicklung%20einer%20Chart%20Library%20zur%20internen%20Verwendung_V1.0.docx#_Toc482180337)

[Abbildung 24: Anwendungsfälle 69](#_Toc482180338)

[Abbildung 25: ecma international 74](file:///D:\nicol\Dropbox%20(enersis)\55%20Shared%20Services\Office%20IT\03_Projekte\Projekt_IPA_SNI\IPA_Nicole_2017\02_Dokumentation\11_Tag\Entwicklung%20einer%20Chart%20Library%20zur%20internen%20Verwendung_V1.0.docx#_Toc482180339)

[Abbildung 26: BABEL 75](file:///D:\nicol\Dropbox%20(enersis)\55%20Shared%20Services\Office%20IT\03_Projekte\Projekt_IPA_SNI\IPA_Nicole_2017\02_Dokumentation\11_Tag\Entwicklung%20einer%20Chart%20Library%20zur%20internen%20Verwendung_V1.0.docx#_Toc482180340)

[Abbildung 27: Babel Beispiel 75](#_Toc482180341)

[Abbildung 28: ESLint 75](file:///D:\nicol\Dropbox%20(enersis)\55%20Shared%20Services\Office%20IT\03_Projekte\Projekt_IPA_SNI\IPA_Nicole_2017\02_Dokumentation\11_Tag\Entwicklung%20einer%20Chart%20Library%20zur%20internen%20Verwendung_V1.0.docx#_Toc482180342)

[Abbildung 29: mocha und chai 76](file:///D:\nicol\Dropbox%20(enersis)\55%20Shared%20Services\Office%20IT\03_Projekte\Projekt_IPA_SNI\IPA_Nicole_2017\02_Dokumentation\11_Tag\Entwicklung%20einer%20Chart%20Library%20zur%20internen%20Verwendung_V1.0.docx#_Toc482180343)

[Abbildung 30: Testing des Titels 76](file:///D:\nicol\Dropbox%20(enersis)\55%20Shared%20Services\Office%20IT\03_Projekte\Projekt_IPA_SNI\IPA_Nicole_2017\02_Dokumentation\11_Tag\Entwicklung%20einer%20Chart%20Library%20zur%20internen%20Verwendung_V1.0.docx#_Toc482180344)

[Abbildung 31: Klassendiagramm 79](#_Toc482180345)

[Abbildung 32: testing the Constructor 84](#_Toc482180346)

[Abbildung 33: Testing the Title 85](#_Toc482180347)

[Abbildung 34: Testing the Legend 86](#_Toc482180348)

[Abbildung 35: Testing the DurationTime 87](#_Toc482180349)

[Abbildung 36: Testing the Color Class 88](#_Toc482180350)

[Abbildung 37: Testing the Color Class 89](#_Toc482180351)

[Abbildung 38: Testing the Ease Class 90](#_Toc482180352)

[Abbildung 39: Testing the Scale Class 91](#_Toc482180353)

[Abbildung 40: Testing the Size Class (width) 92](#_Toc482180354)

[Abbildung 41: Testing the Size Class (height) 93](#_Toc482180355)

[Abbildung 42: Testing Type Class 94](#_Toc482180356)

[Abbildung 43: Unterschrift für Abnahme 104](#_Toc482180357)

**Tabellenverzeichnis**

[Tabelle 1: Projektrollen 9](#_Toc482180358)

[Tabelle 2: Inhalte der Ordnerstruktur 13](#_Toc482180359)

[Tabelle 3: Projektjournal 35](#_Toc482180360)

[Tabelle 4: Mittelbedarf 37](#_Toc482180361)

[Tabelle 5: Funktionen Barchart 46](#_Toc482180362)

[Tabelle 6: Funktionen Donutchart 49](#_Toc482180363)

[Tabelle 7: Funktionen des Barcharts 49](#_Toc482180364)

[Tabelle 8: SMART 50](#_Toc482180365)

[Tabelle 9: Risikoanalyse 57](#_Toc482180366)

[Tabelle 10: Risikobeschreibung 58](#_Toc482180367)

[Tabelle 11: Risikobeschreibung 58](#_Toc482180368)

[Tabelle 12: Kriterien 60](#_Toc482180369)

[Tabelle 13: React Redux Starter Kit 61](#_Toc482180370)

[Tabelle 14: ES6 Karma Jasmine Webpack Boilerplate 62](#_Toc482180371)

[Tabelle 15: Webpack library starter 62](#_Toc482180372)

[Tabelle 16: Variantenentscheid 63](#_Toc482180373)

[Tabelle 17: Vom Source Code zur fertigen Library 65](file:///D:\nicol\Dropbox%20(enersis)\55%20Shared%20Services\Office%20IT\03_Projekte\Projekt_IPA_SNI\IPA_Nicole_2017\02_Dokumentation\11_Tag\Entwicklung%20einer%20Chart%20Library%20zur%20internen%20Verwendung_V1.0.docx#_Toc482180374)

[Tabelle 18: Testfälle 72](#_Toc482180375)

[Tabelle 19: Hauptbausteine von Chai 77](#_Toc482180376)

[Tabelle 20: Vom Source Code zur fertigen Library 77](file:///D:\nicol\Dropbox%20(enersis)\55%20Shared%20Services\Office%20IT\03_Projekte\Projekt_IPA_SNI\IPA_Nicole_2017\02_Dokumentation\11_Tag\Entwicklung%20einer%20Chart%20Library%20zur%20internen%20Verwendung_V1.0.docx#_Toc482180377)

[Tabelle 21: Was ist webpack 78](#_Toc482180378)

[Tabelle 22: Glossar 103](#_Toc482180379)

**Kurzfassung des IPA-Berichtes**

Ausgangssituation

Umsetzung

Ergebnis

2017

Text adventure – the little mermaid

AutoriN: Nicole Sager

Teil 1: Ablauf und Umfeld

# Aufgabenstellung

Dieses Kapitel beschreibt die Aufgabenstellung gemäss Originaleingabe des Fachvorgesetzten auf PKORG.

## Titel der Facharbeit

Text adventure – The Little Mermaid

## Thematik

Ein simples Text adventure soll erstellt werden. Dieses soll von der Originalgeschichte der kleinen Meerjungfrau (geschrieben von Hans Christian Andersen) inspiriert werden. Für die Umsetzung soll Python verwendet werden.

## Ausgangslage

Um sich für die IPA entsprechend vorzubereiten wird eine Probe-IPA durchgeführt. Im Gegensatz zu der „richtigen“ IPA, welche 10 Tage dauert, wird diese erste IPA 5 Tage dauern. Python wurde vorgegeben da dies die Programmiersprache ist welche im normalen Arbeitsumfeld verwendet wird.

## Detaillierte Aufgabenstellung

Gefordert wird ein Textbasiertes Spiel. Es werden mindestens 3 verschiedene Auswahlmöglichkeiten gefordert welche jeweils eine Auswirkung auf das ende des Spiels haben müssen. Die visuellen Eigenschaften des Projektes stehen nicht im Vordergrund. Die ausgegebene Sprache des Spiels ist Englisch.

Das Projekt soll mit Unit-test getestet werden.

## Mittel und Methoden inklusive Projektmethode

HERMES 5.1 IPA soll als Projektmethode verwendet werden.

Zu benutzten ist das von der Firma zur Verfügung gestellte Macbook Pro.

Das Programm soll mit Python realisiert werden.

## Vorkenntnisse

Eine Probe IPA, eine offizielle IPA. Ein 1-wöchentliches training bezüglich Python.

## Vorarbeiten

Die teilweise Wiederverwendung bereits erstellter Diagramme.

## Neue Lerninhalte

Ein eigenes Projekt mit Python umzusetzen.

## Arbeiten des Kandidaten im Schwerpunkt während dem 3. und 4. Lehrjahr

Die Lernende absolviert den vierjährigen Bildungsgang der Informatikmittelschule in der Fachrichtung Applikationsentwicklung und kaufmännische Berufsmaturität.

Auf die dreijährige schulische Vollzeitausbildung am bwd folgt für die Lernende ein Praktikumsjahr im Betrieb der enersis suisse AG und am 02.08.2017 startete die lernende beim Betrieb 89grad GmbH. Die Ausbildungszeit "on-the-job" ist daher relativ kurz.

# Detaillierte Projektmethode

In diesem Teil der Arbeit werden die einzelnen Phasen und Module dokumentiert.

Im ersten Abschnitt wird die vordefinierte Projektmethode genauer erklärt. Wie in Kapitel 1.5 Projektmethode beschrieben, wird Hermes 5.1 IPA angewendet.

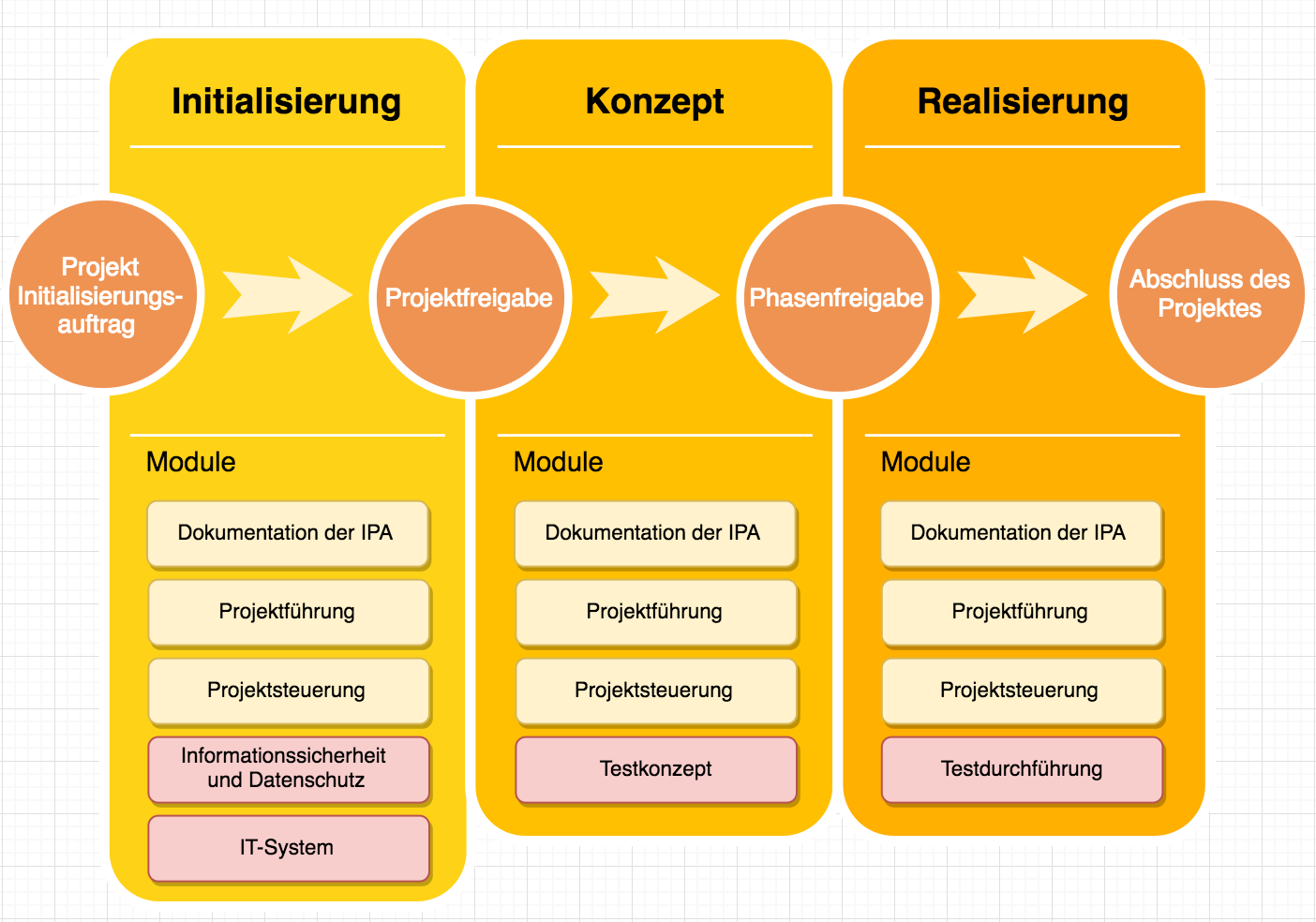


Abbildung : Detaillierte Projektmethode

Das ebengezeigte Diagramm Zeigt in welchen Phasen welche Module durchgeführt werden.

## Projektmethode

Hermes 5.1 IPA ist für die Durchführung von IPAs entwickelt worden. Es ist eine stark vereinfachte Form von Hermes 5.1.[[1]](#footnote-1)

## Phasen

Das Phasenmodell gliedert den Lebenszyklus des Projekts und schafft die Voraussetzung für das gemeinsame Verständnis der Projektbeteiligten zum Projektablauf. Das Phasenmodell bildet unter anderem ebenfalls die Grundlage für die finanzielle Steuerung des Projekts.

Bei Phasenfreigabe werden die Mittel (finanziell, personell, infrastrukturell) für die jeweils anschliessende Phase durch den Auftraggeber freigegeben.

Die vier Phasen einer Hermesplanung sind die Initialisierungsphase, die Konzeptphase, die Realisierungsphase und zum Schluss die Einführungsphase

Es ist zu beachten das bereits im Vorfeld festgelegt wurde, dass es keine eigentliche Phasenfreigabe geben wird, da dies zu zeitintensiv wäre. Ausserdem wird in dieser Probe IPA keine Einführungsphase durchlaufen.

## Module

Folgende Module wurden in dieser IPA verwendet:

|  |  |
| --- | --- |
| **Modul** | **Beschreibung** |
| **Dokumentation der IPA** | * Der Verlauf der IPA wird dokumentiert und stets aktuell gehalten * Das Arbeitsjournal wird täglich geschrieben |
| **Projektführung** | * Das Projekt planen, führen und in den definierten Rahmenbedingungen von Zeit und Kosten mit dem geforderten Ergebnis zum Ziel bringen * Die Interessen der Stakeholder kennen, die Kommunikation führen und Entscheide sicherstellen * Risiken managen, Probleme bewältigen und Erfahrungen berücksichtigen * Qualitätssicherung führen gemäss IPA Richtlinien und Firmenrichtlinien |
| **Projektsteuerung** | * Das Projekt initialisieren, kontinuierlich steuern und mit den übergeordneten Zielen und Vorgaben der Stammorganisation in Übereinstimmung halten * Anliegen der Stakeholder berücksichtigen und integrieren, Risiken managen und Entscheide treffen * Das Projekt abschliessen |
| **Informationssicherheit und Datenschutz** | • Anforderungen der Sicherheit und des Datenschutzes ermitteln, Risiken bewerten und Massnahmen zur Erfüllung der Anforderungen konzipieren und umsetzen   * Das ISDS-Konzept erstellen und die Ergebnisse laufend dokumentieren * Datenschutz gemäss Firmenrichtlinien umsetzten |
| **IT-System** | * Das IT-System realisieren und dokumentieren * Die Systemanforderungen verfeinern, die Systemarchitektur erarbeiten und die Machbarkeit überprüfen * Die Detailspezifikation erarbeiten sowie das System und die Integration realisieren |
| **Testkonzept** | * Konzeption des Testings * Szenarien erstellen * Vorgehen definieren wie Strategie und Methodiken * Tests vorbereiten (Szenario) |
| **Testdurchführung** | * Tests durchführen und dokumentieren * Testfazit erstellen * Erkenntnisse ableiten |

Abbildung 2: Module die für das Projekt verwendet wurden

## Szenario

Das Szenario legt ein grundsätzliches Schema für die Inhalte der Phasen fest. Bei dieser Probe IPA wurde das Szenario „IT-Individualanwendung“ ausgewählt.

|  |  |
| --- | --- |
| **Szenario** | **Beschreibung** |
| **IT-Individualanwendung** | Für die spezifischen Bedürfnisse eines Fachbereichs einer IT-Anwendung entwickeln und technisch und organisatorisch integrieren |

## Projektorganisation

Die Projektorganisation stellt die hierarchische Struktur des Projektes dar.

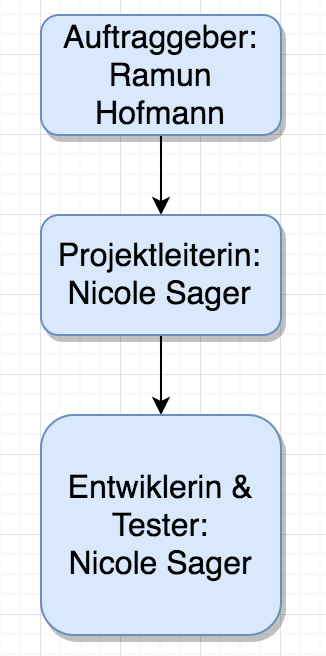


Abbildung 4: Projektorganisation

### Projektrollen

Die IPA Projektrollen werden in der folgenden Tabelle kurz erläutert:

|  |  |
| --- | --- |
| Rolle | Rollenbeschreibung |
| Auftraggeber | Erteilt den Auftrag mit den gewünschten Anforderungen. |
| Projektleiter | Ist für die Planung und Steuerung des Projektes verantwortlich |
| Entwickler | Setzt das System um. |

Tabelle 1: Projektrollen

# Zeitplan!!!

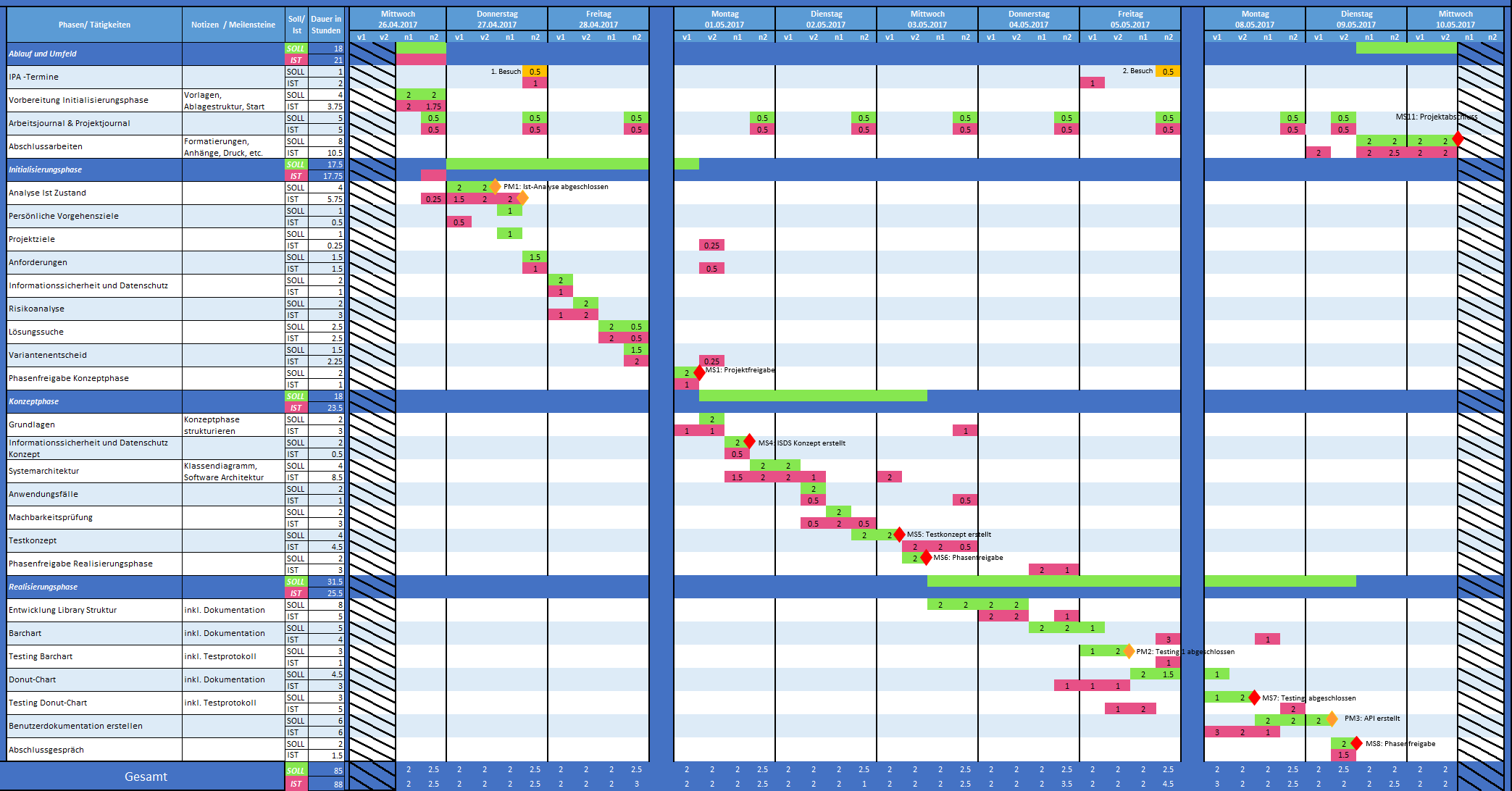


Abbildung 4: Zeitplan

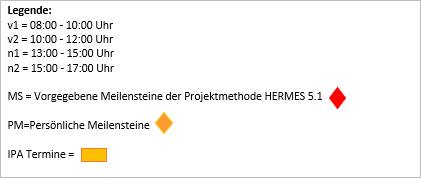


Abbildung 5: Legende Zeitplan

Die Aufgezeigten roten Meilensteine sind die von Hermes 5.1 IPA vorgegebenen Meilensteine. Hierbei wurden bewusst die jeweiligen Phasenfreigaben ausgelassen.

# Organisation der Probe IPA !!!

Abbildung 6: Arbeitsplatz

Der Arbeitsplatz besteht aus einem Macbook Pro einem verfügbaren Stuhl/Tisch.

Zusätzlich stehen verschiedene Büromaterialien zur Verfügung.

## Infrastruktur

Wie bereits im Kapitel 1 vermerkt erfolgt die Realisierung im Rahmen der lokalen Infrastruktur.

* Einen eigenen PC
* Entwicklungsumgebung: Visual Studio Code- als Texteditor

Zusätzlich werden verwendet:

* Draw.io – für das Erstellen Diverser Diagramme

## Datensicherung der IPA

Die Sicherung erfolgt einmaltäglich auf Github sowie während des normalen Arbeitens auf der Lokalen Maschine. Eine Umstellung der zentralen Ablage ist geplant, dies kann jedoch für die IPA nicht mehr berücksichtigt werden.

Die Dokumente, die während dieser Arbeit erstellt werden, werden laufend gespeichert. Zudem erfolgt eine Zweitspeicherung einmal pro Tag auf einem USB-Stick.

## Ordnerstruktur



Abbildung : Ordnerstruktur

Die vorherige Abbildung zeigt die für die Projektorganisation erstellte Ordnerstruktur auf.

In der folgenden Liste wird kurz erläutert, welche Inhalt die jeweiligen Ordner besitzen bzw. besitzen werden.

|  |  |  |
| --- | --- | --- |
| **Nr.** | **Name** | **Inhalte** |
| **00** | Unterlagen Pkorg | Alle Vorlagen, die auf der Pkorg Seite zur Verfügung stehen, sowie eine PDF Version der Kandidatenseiten und dem dazugehörigen Detailbeschrieb werden darin abgespeichert. |
| **01** | Vorlagen und Vorarbeiten | Die bereits als Vorarbeit erstellten Dokumente & die als Übung durchgeführten IPAs welche als Hilfestellung verwendet werden |
| **02** | Dokumentation | Enthält die Ordnerstruktur für die einzelnen Tage |
| **07** | Bilder | Die in diesem Dokument verwendeten Bilder |
| **98** | Sonstiges | Alles Dokumente, welche keiner der Hauptkategorien zugeordnet werden können |
| **99** | SrcCode[[2]](#footnote-2) | Der für das Projekt erstellte Quelltext |

Tabelle 2: Inhalte der Ordnerstruktur

In einem der einzelnen Tage ist die Dokumentation und der Zeitplan enthalten.

# Firmenstandards

Die Firma 89grad besitzt momentan keine relevanten Firmenstandards.

# Arbeitsjournal

Gemäss Art. 5 Absatz 2 der Wegleitung über die individuelle praktische Arbeit (IPA) an Lehrabschlussprüfungen des BBT vom 27. August 2001 gilt:

*„Die zu prüfende Person führt ein Arbeitsjournal. Sie dokumentiert darin täglich das Vorgehen, den Stand der Prüfungsarbeit, sämtliche fremde Hilfestellungen (auch das Internet ist eine Hilfestellung) und besondere Vorkommnisse wie z.B. Änderungen der Aufgabenstellung, Arbeitsunterbrüche, organisatorische Probleme, Abweichungen von der Soll-Planung.“*

Das Arbeitsjournal zur IPA ist zwingend zu führen und den Experten und Fachvorgesetzten vorzulegen. Das Arbeitsjournal ist täglich sinngemäss und korrekt auszufüllen.

Das Arbeitsjournal dient der Nachvollziehbarkeit der von den Lernenden ausgeführten Arbeiten und wird als Teil der IPA in die Bewertung mit einbezogen.

Die Tätigkeit war für heute geplant.

Die Tätigkeit war für einen späteren Tag geplant.

Die Tätigkeit war für einen bereits vergangenen Tag geplant.

## Erster Tag: Montag, 14.08.2017

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Beteiligte  Personen | SOLL  (Std) | IST  (Std) |
| Vorbereitung Initialisierungsphase | Nicole Sager | 4 | 4.5 |
| Arbeitsjournal | Nicole Sager | 0.5 | 0.5 |
| Analyse Ist Zustand | Nicole Sager | 1 | 0.25 |
| Persönliche Vorgehensziele | Nicole Sager | 0.5 | 0.25 |
| Projektziele | Nicole Sager | 0.5 | 0.25 |
| Anforderungen | Nicole Sager | 1 | 1 |
| Informationssicherheit und Datenschutz | Nicole Sager | 0.5 | 0.25 |
| Machbarkeitsprüfung | Nicole Sager | 0 | 1.5 |
| **Total**: | - | 8 | 8.5 |
| Tagesablauf |  |  |  |
| Heute habe ich das Projekt gestartet. Es waren vor allem Vorbereitungen für die Initialisierungsphase und bereits einzelne Bereiche der Initialisierungsphase. Unter anderem habe ich die Ordnerstruktur erstellt und die allgemeinen Informationen ausgefüllt & die bestätigte Aufgabenstellung übernommen.  Der vorbereitete Zeitplan habe ich noch ergänzt die Kapitel Detaillierte Projektmethode, Organisation der Probe IPA & Firmenstandards habe ich abgeschlossen.  Ich hatte sogar noch etwas Zeit, um erste Arbeiten an der Machbarkeitsprüfung umzusetzen. | | | |
| Hilfestellungen |  |  |  |
| www.hermes.admin.ch(14.08.2017)  Leitfaden\_Hermes5\_1.pdf  Hermes 5.1 – Projektmanagement für alle Projekte, REFERENZHANDBUCH (Eidgenössisches Finanzdepartement EFD, Informatiksteuerungsorgan des Bundes ISB, 2014)  https://de.wikipedia.org/wiki/SMART\_(Projektmanagement)(26.04.2017) | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Reflexion |  |  |  |
| **Was lief gut?**  Ich bin mit der Dokumentation gut vorangekommen und sogar leicht vor meinem Zeitplan.  Es war eine sehr interessante Erfahrung so oft um Hilfe gefragt zu werden, aber da ich nun mal die einzige mit etwas Erfahrung bin war das wohl vorhersehbar.  Auch ist es schön das sich meine bisherige Arbeit schlussendlich doch auszahlt.  **Was lief weniger gut?**  Da ich bereits wusste was auf mich zukommt eigentlich nichts.  **Meine Erkenntnisse von heute:**  Je öfter man eine IPA macht desto einfacher wird es. | | | |
| Nächste Schritte |  |  |  |
| Morgen möchte ich die Initialisierungsphase abschliessen und mit der Konzeptphase beginnen. | | | |

# Abschlussbericht

Der Abschlussbericht dient der Zusammenfassung des gesamten durchgeführten Projektes.

## Vergleich Ist/Soll

In der folgenden Auflistung wird überprüft, ob die gestellten Anforderungen erfüllt worden sind. Hierbei gilt: ✓ als erfüllt und 🗶 als nicht erfüllt oder nicht vollständig

* Die Chart Library soll folgende zwei Diagrammtypen zur Verfügung stellen:
  + Donut-Chart
  + Barchart

Wie zu sehen ist, wurden beinahe alle Punkte erfüllt. Einzig und allein das Barchart konnte aufgrund aufgetretener Probleme nicht vollständig abgeschlossen werden.

## X Mittelbedarf

Die Mittel, welche verwendet wurden sind nachfolgend aufgelistet. Anzumerken ist, das keinerlei zusätzliche Mittel besorgt werden mussten.

|  |  |
| --- | --- |
| Mittel | Bedarf für |
| Macbook Pro | Ist die von der Firma zur Verfügung gestellte Arbeitsgerät. |
| Draw.io | Zur professionellen Visualisierung von Diagrammen |
| Office 365 | Zur allgemeinen Bearbeitung der Dokumentation |
| Visual Sudio Code | Zur Bearbeitung und Erstellung des Souce Codes und dessen Testing. |
| Schreibzeug und Papier | Wurde für allgemeine Notizen, zum Strukturieren und Planen des Projektes verwendet |
| Harvest | Wird für die allgemeine Zeiterfassung verwendet |

Tabelle 4: Mittelbedarf

## Realisierungsbericht

### Ungeplante Ereignisse

Es gab zu Beginn einige Probleme mit dem Erstellen der Unit Tests. Es stellte sich heraus, dass einige Konfigurationen bezüglich des Source Codes falsch gewesen waren. Das Problem hierbei war, dass das Tool Mocha, welches für das Unit Testing verwendet wurde, direkt auf den Source Code zugegriffen hat. Da dieser jedoch in ECMAScript 6 geschrieben wurde, gab es den Fehler unexpected Token. Dadurch, dass Mocha auf die von Babel formatierte Version, welche in ECMAScript 5 geschrieben ist, zugriff konnte dieser Fehler behoben werden.

Zudem hatte die Projektleiterin mit dem ihr unbekannten ECMAScript 6 und dessen klassenorientierte Schreibweise keinerlei Erfahrung wodurch die Umsetzung erschwert wurde.

### Erkenntnisse der Realisierung

Anderweitige grössere Probleme Traten nicht auf. Jedoch war vor allem die Umsetzung des Testing und der Umgang mit dem für die Projektleiterin noch unbekannten ECMAScript 6, sehr zeitintensiv.

## Testbericht

Die Unit Tests verliefen allesamt gut. Es ist zwar nicht so, dass alles auf Anhieb funktioniert hat, jedoch wurde dies auch nicht erwartet. Durch die stetig kontrollierten Tests wurden alle Fehler und möglichen Fehler schnell erkannt und umgehend behoben.

Die Tests wurden solange wiederholt, bis alle Testdurchläufe erfolgreich verliefen. Dies wurde bei allen implementierten Testfällen erreicht.

## Fazit zum IPA (Projekt)

## Persönliches Fazit

Da es sich um ein persönliches Fazit handelt, wurde dieses Unterkapitel in der Ich-Person verfasst.

Ich persönlich bin sehr zufrieden mit dem Endergebnis meiner IPA.

## Schlussreflexion

Für mich persönlich war das Dokumentieren am schwierigsten. Da ich mir weder Wörter noch Buchstaben oder Zahlen bildlich vorstellen kann, war das formulieren der Sätze besonders schwer. Was ich jedoch bei der Erstellung der API Dokumentation feststellte, war, dass es für mich einfacher war auf Englisch zu schreiben als auf Deutsch.

Trotz dieses persönlichen Mankos habe ich alles versucht, um eine so aussagekräftige Dokumentation wie irgend möglich zu erstellen. Am Ende der Arbeit habe ich gelernt was es wirklich heisst, ein Projektleiter zu sein und wie schwer es sein kann, das grosse Ganze nicht aus den Augen zu verlieren. Dies ist ein Punkt, den ich besonders für die nächsten Arbeiten mitnehmen möchte. Dass es wichtig ist zuerst ein gutes Konzept und anschliessend eine solide Grundstruktur zu erstellen da man sich sonst in den Details verheddert.

# Unterschriften Teil 1

Die lernende Person bestätigt mit ihrer Unterschrift diese IPA aus Eigenleistung erbracht und nach den Vorgaben der Prüfungskommission Informatik Kanton Bern erstellt zu haben. Die Angaben im Arbeitsjournal entsprechen dem geleisteten Arbeitsaufwand.

Da die Phasenfreigabe während dieser ProbeIPA nicht durchgeführt werden gibt es entsprechend auch keine Unterschrift.

Teil 2: Projektdokumentation

# Initialisierung

„Die Initialisierung schafft eine definierte Ausgangslage für das Projekt/IPA und stellt sicher, dass die Projektziele mit PkOrg übereinstimmen. Die Projektgrundlagen und der Projektauftrag werden erarbeitet. Es wird ein Variantenentscheid getroffen oder es wird begründet warum es keinen gibt. Der Variantenentscheid wird schlussendlich vom Lernenden erarbeitet, geprüft und eingeführt.“[[3]](#footnote-3)

## Studie; Ist-Zustand

## X Meilensteine

Nach Hermes 5.1 IPA wurden bereits sieben Meilensteine vorgegeben. (Im Normalfall wären es acht jedoch wird in dieser Probe IPA die Einführungsphase nicht durchgeführt)

Diese wären wie Folgt:

1. Projekt Initialisierungs-auftrag
   1. ISDS Konzepterstellt
2. Projektfreigabe
   1. Testkonzept
3. Phasenfreigabe
   1. Test abgeschlossen
4. Abschluss des Projektes

### X SMART

Die persönlichen Ziele und die Projektziele wurden mithilfe des SMART-Prinzips erstellt.

|  |  |  |
| --- | --- | --- |
| **Buchstabe** | **Bedeutung** | **Beschreibung** |
| **S** | Spezifisch | Ziele müssen eindeutig definiert sein (nicht vage, sondern so präzise wie möglich). |
| **M** | Messbar | Ziele müssen messbar sein (Messbarkeitskriterien). |
| **A** | Ansprechend | Die Ziele müssen für die Person ansprechend bzw. erstrebenswert sein, zum Teil auch "achievable" oder "attainable", also erreichbar, dann steht das 'R' für 'relevant', ursprünglich "assignable", also - einem bestimmten Verantwortlichen - zuweisbar. |
| **R** | Realistisch | Das gesteckte Ziel muss möglich und realisierbar sein. |
| **T** | Terminiert | Das Ziel muss mit einem fixen Datum festgelegt werden können. |

Tabelle 8: SMART[[4]](#footnote-4)

### X Persönliche Vorgehensziele

In diesem Abschnitt werden die persönlichen Ziele der Absolventin definiert. Dabei ist zu beachten, dass die Ziele unabhängig vom Variantenentscheid sowie der Realisierung gestellt sind.

Die Ziele wurden nach dem SMART Prinzip, welches im vorherigen Kapitel erklärt wurde, erstellt.

1. Bis spätestens 14.08.2017 ist die Ist-Analyse abgeschlossen.
2. Bis spätestens 17.08.2017 ist der Realisierungsteil abgeschlossen und die Ergebnisse wurden dokumentiert.
3. Bis spätestens 18.05.2017 ist die Dokumentation, einschliesslich der zusätzlichen Dokumente wie Arbeitsjournal, Projektjournal, Handbuch etc. abgeschlossen.

### X Projektziele

In diesem Abschnitt werden die allgemeinen Ziele definiert. Dabei ist zu beachten, dass die Ziele unabhängig vom Variantenentscheid sowie der Realisierung gestellt sind.

Die Ziele wurden nach dem SMART Prinzip, welches im Kapitel 9.2 erklärt wurde, erstellt.

1. Das ISDS-Konzept muss bis spätestens dem 15.08.2017 fertiggestellt werden.
2. Das Test-Konzept muss bis spätestens dem 16.08.2017 fertiggestellt werden.
3. Bis spätestens 18.08.2017 ist das Testing abgeschlossen und die Ergebnisse wurden dokumentiert.
4. Bis spätestens 18.08.2017 ist die Dokumentation, einschliesslich der zusätzlichen Dokumente wie Arbeitsjournal abgeschlossen und die ausgedruckten Exemplare wurden abgegeben und eine digitale Version wurde via Mail verschickt.

## Anforderungen

Die Anforderungen in einem Projekt sind die definierten Eigenschaften eines Produktes welche vom Auftraggeber definiert wurden. Es ist die Aufgabe des Auftragnehmers diese zu analysieren, spezifizieren und gegeben Falls zu erweitern, letzteres jedoch nur mit Absprache.

### Funktionale Anforderungen

Funktionale Anforderungen beschreiben WAS das System bzw. Produkt leisten soll. Die funktionalen Anforderungen beschreiben somit die gewünschten Funktionalitäten des Systems, aber auch dessen Daten oder Verhalten.

* Es sollen min. 3 verschiedene Eingabeparameter vom Benutzerverlangt werden welche eine Auswirkung auf den Verlauf des Spiels haben
* Es sollen Pausen integriert werden welche den lese Fluss erleichtern sollen
* Das Spiel Soll mit Eingabe des eigenen Namens Personalisiert werden

### Nicht funktionale Anforderungen

Nicht funktionale Anforderungen beschreiben WIE GUT das System etwas leisten soll und decken im Prinzip die qualitativen Aspekte und Bedingungen des Systems oder Funktionalitäten ab (z.B. Erweiterbarkeit).

* Das Produkt soll mittels whitebox-testing getestet werden
  + Jede dieser Inputparameter soll via Unittesting getestet werden

## Informationssicherheit und Datenschutz (ISDS)

Da dieses Projekt keinerlei sensitive Daten enthält, wird von einer zusätzlichen Sicherung, wie z.B. mittels Eingaben von Anmeldeinformationen, zur Verwendung des Codes abgesehen.

Wie bereits im Kapitel 1 und im Kapitel 4 vermerkt erfolgt die Dokumentenablage während des normalen workflows auf der Lokalen Maschine eine weitere Sicherung erfolgt jeweils einmal täglich auf Github.

## Machbarkeitsprüfung Initialisierungsphase

Die folgende Machbarkeitsprüfung wurde von dem Dokument «14.\_checkliste\_machbarkeitspruefung» übernommen und teilweise angepasst. Dieses Dokument wird von der Webseite der Finanzdirektion des Kantons Bern [[5]](#footnote-5)zur Verfügung gestellt.

Die Machbarkeitsprüfung wird zuerst hier in der Initialisierungsphase umgesetzt und anschliessend noch einmal in der Konzeptphase aufgenommen. Die Machbarkeitsprüfung wird zur Sicherstellung eines realistischen Projektziels, sprich ob mit den zur Verfügung gestellten Ressourcen etc. das Projekt in der geforderten Qualität umgesetzt werden kann.

Durch die Durchführung einer Machbarkeitsprüfung wird das Risiko vermeidbarer Aufwände aufgrund Nichtdurchführbarkeit enorm gesenkt, da man bereits vor der eigentlichen Umsetzung mögliche Risikofaktoren erkennt und dadurch Projektanforderungen anpasst oder im schlimmsten Fall das Projekt abbrechen kann.

Für dieses Projekt werden folgende Aspekte geprüft und bewertet:

1. Technische und fachliche Machbarkeit
2. Ressourcen und Verfügbarkeit
3. Wirtschaftliche Machbarkeit
4. Zeitliche Umsetzung
5. **Technische und fachliche Machbarkeit**

Die Grundlage der technischen und fachlichen Machbarkeitsprüfung bilden die Anforderungen.

Sind die technischen Anforderungen überhaupt erfüllbar? Ja  Nein

*Die separaten Teile des Projektes wurden in verschiedenen Übungen schon verwendet bzw. umgesetzt. Zudem wurden ähnliche Projekte bereits von anderen Entwicklern in einer ähnlichen Form umgesetzt.*

Gibt es Alternativen und was sind die jeweiligen Voraussetzungen? Ja  Nein

*Da dies ein einfaches Übungsprojekt ist wird kein reelles Problem gelöst dadurch ist die Frage nach einer alternativen unsinnig.*

Sind die fachlichen Annahmen und Anforderungen realistisch? Ja  Nein

*Da ähnliche Projekte bereits von zahlreichen anderen Entwicklern durchgeführt wurden, wird angenommen, dass es realistisch ist.*

Sind alle fachlichen Anforderungen der Nutzer berücksichtigt? Ja  Nein

*Der jeweilige Benutzer muss lediglich python3 auf dem Rechner installiert haben und der jeweilige Benutzer muss englisch Verstehen.*

Wo liegen die grössten technischen beziehungsweise fachlichen Risiken?

*Die Entwicklerin besitzt erst sehr wenig Erfahrung mit Python vor allem was das Testing angeht.*

1. **Ressourcen und Verfügbarkeit**

Wie viele Ressourcen stehen für die Umsetzung des gesamten Projektes zur Verfügung?

*Es steht ein Mitarbeiter für die Dauer von 5 Arbeitstagen zu 100% zur Verfügung.*

Wo liegen die grössten Risiken bezüglich der Ressourcen?

*Das Ausfallen der Arbeitskraft durch Krankheit oder Unfall. Von jeglichen anderen Verantwortungen wurde der Arbeitnehmer jedoch freigestellt.*

Welche sind die wichtigsten unterstützenden Massnahmen?

*Die Arbeitskraft schaut eigenständig auf die eigene Gesundheit, erkennt allfällige Symptome und gibt allenfalls nach Feststellung von Krankheitssymptomen umgehend dem Auftraggeber und dem Hauptexperten Bescheid, zudem wird bei einem Krankheitsfall ein Arztzeugnis verlangt.*

Gibt es bei der Verfügbarkeit des Produktes allfällige Risiken? Ja  Nein

Gibt es bei der Verfügbarkeit des Systems auf dem das Produkt entwickelt wird allfällige Risiken? Ja  Nein

*Es besteht immer das Risiko von technischem Versagen, sprich, dass der Rechner auf dem entwickelt wird, kaputtgeht. Aus diesem Grund werden als Massnahme die Dokumente auf Github gespeichert und täglich ein Backup erstellt.*

1. **Wirtschaftliche Machbarkeit**

Wie fällt die Wirtschaftlichkeitsbetrachtung (Kosten/Nutzen) aus (unter welchen Annahmen)?

*Das Kosten/Nutzen für die Firma 89grad ist schlecht da das produzierte Projekt keine einnahmen erbringen wird.*

Welche finanziellen Ressourcen werden wann benötigt? Ist dies machbar? Was gilt es zu beachten?

*Während 10 Arbeitstagen wird die Projektleiterin 100% nur an diesem Projekt arbeiten. Diese Ressourcen sind fix und sind nicht verhandelbar. Da der Zeitrahmen und damit die Kosten fix sind können die Kosten für das Projekt in diesem Rahmen nicht steigen.*

Welches sind die grössten Kosten- und Terminrisiken?

*Die grössten Kosten sind die zwei Arbeitswochen an Ressourcen, die 5Tage für das Projekt freigestellt werden müssen.*

*Das grösste Terminrisiko ist der Abgabetermin.*

1. **Zeitliche Umsetzung**

Wie viele Ressourcen stehen für die Umsetzung des gesamten Projektes zur Verfügung?

*Es steht ein Mitarbeiter für die Dauer von 5 Arbeitstagen zu 100% zur Verfügung.*

Wo liegen die grössten Risiken bezüglich der Zeitlichen Umsetzung?

*Da der Zeitplan sehr knapp ist kann bei jeglicher Verzögerung oder beim Auftauchen unerwarteter Probleme nicht mehr versichert werden, dass das Projekt abgeschlossen werden kann.*

Welche sind die wichtigsten unterstützenden Massnahmen?

*Das Einhalten des Zeitplanes und schnelle Reaktionszeit bei unerwarteten Problemen bzw. einer offenen Kommunikation mit dem Auftraggeber.*

Gibt es bei der Verfügbarkeit des Systems auf dem das Produkt entwickelt wird allfällige Risiken? Ja  Nein

*Bei einem Projekt bei welchem die Zeitbegrenzung nicht verhandelbar ist besteht immer ein Risiko.*

**Fazit:** Ist das Projektziel realistisch? Ja Nein

Wenn sich Zweifel an der Machbarkeit ergeben gilt es, zwischen den drei folgenden Möglichkeiten zu entscheiden:

* Durchführung einer detaillierten Machbarkeitsstudie
* Projektabbruch
* Projektdurchführung trotz hohen Risikos

**Empfehlung**

Es wird empfohlen das Projekt durchzuführen.

## Risikoanalyse

Anschliessend ist eine Tabelle bezüglich der Risiken abgebildet, welche im schlimmsten Fall auftreten könnten.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nr. | Risikobeschreibung | Auswirkung | Vor Massnahme | | | | Massnahmen/Erklärung | Nach Massnahme | | | |
| W | S | Risiko | Handlungsweise | **W** | **S** | **Risiko** | **Handlungsweise** |
|  | Verpassen des Abgabetermins | -Bewertung der Arbeit mit 0.5 Note Abzug  -Führt zum Nichtbestehen der Arbeit. | W3 | S4 | hoch | Risikominderung | -Erstellung und Einhaltung eines realistischen Zeitplans  -Einrichtung einer Terminerinnerung im Outlook  -Fertigstellungstermin vor dem Abgabetermin setzen. | W2 | S4 | mittel | Risikoakzeptanz |
|  | Umfang und Komplexität | -Die Arbeit kann nicht fertiggestellt werden. | W5 | S3 | hoch | Risikominderung | -Erstellung und Einhaltung eines realistischen Zeitplans  -Möglichst früher Fertigstellungstermin | W2 | S3 | gering | Risikoakzeptanz |
|  | Krankheitsfall | -Die Arbeit gerät massiv in den Rückstand.  -Verlorene Zeit muss nachgeholt werden oder die Arbeit kann nicht fertiggestellt werden. | W3 | S2 | mittel | Risikominderung | -Symptome frühzeitig erkennen.  -Bei eintreten umgehend ein Arztzeugnis besorgen, den Experten und den Auftraggeber informieren. | W2 | S2 | gering | Risikoakzeptanz |
|  | Begehung formaler Fehler | Abwertung der Endnote | W3 | S2 | mittel | Risikominderung | -Review der Arbeit durch Dritten mit Kenntnis der Formalitäten (z.B. Kollegen Familienmitglieder, etc.) | W2 | S2 | gering | Risikoakzeptanz |
|  | Verlust der Arbeit | -Die Arbeit kann nicht fertiggestellt werden.  -Teilbereiche müssen nochmals gemacht werden. | W3 | S3 | mittel | Risikominderung | -Versionisierung der Arbeit  -Regelmässiges Erstellen von Backups (täglich). | W1 | S3 | gering | Risikoakzeptanz |
|  | Nichtbemerken von fehlerhaftem Code | -Ein nicht funktionsfähiges Programm | W4 | S2 | Mittel | Risikominderung | -Ein gutausgearbeitetes Testing  -Wenn Fehler bemerkt werden jedoch für dessen Behebung nicht genügend Zeit vorhanden ist werden diese Dokumentiert | W3 | S1 | gering | Risikoakzeptanz |
|  | Ausfallen der Infrastruktur | -Blockiert die Arbeiten. Somit gerät die Arbeit in den Rückstand | W3 | S2 | gering | Risikominderung | -Tägliche Sicherung aller Dokumente erstellen  -Ersatzgeräte bereithalten  -Infrastruktur genügend testen | W1 | S2 | gering | Risikoakzeptanz |
|  | Höhere Gewalt | Die Arbeit kann im schlimmsten Fall nicht fertig gestellt werden. | W1 | S3 | gering | Risikoakzeptanz | -Massnahmen können bei Höherer Gewalt nicht getroffen werden.  -Da die Eintrittswahrscheinlichkeit jedoch sehr gering ist, wird das Risiko akzeptiert |  | | | |

Diese Risiken werden allgemein gehalten und sind nicht auf die spezifische Umsetzung, sprich das Programmieren, bezogen.

Tabelle 9: Risikoanalyse

|  |
| --- |
| **Schadensausmaß:**  S1 = führt zu keiner Abwertung  S2 = geringe Abwertung bis 1.0 Notenpunkte  S3 = hohe Abwertung über 1,0 Notenpunkte  S4 = führt zu Nichtbestehen    **Eintrittswahrscheinlichkeit:**  W1 = unvorstellbar  W2 = unwahrscheinlich  W3 = eher vorstellbar  W4 = vorstellbar  W5 = Eintreffen hoch |

## Risikographen

Folgend sind die beiden Risikographen abgebildet. Die Nummern stehen in Bezug zu denen in der vorherigen Risikoanalyse definierten Risiken (Kapitel 9.6).

### Risikograph vor den Massnahmen

|  |  |
| --- | --- |
| Nr. | **Risikobeschreibung** |
| 1 | Verpassen des Abgabetermins |
| 2 | Umfang und Komplexität |
| 3 | Krankheitsfall |
| 4 | Begehung formaler Fehler |
| 5 | Verlust der Arbeit |
| 6 | Nichtbemerken von fehlerhaftem Code |
| 7 | Ausfallen der Infrastruktur |
| 8 | Höhere Gewalt |

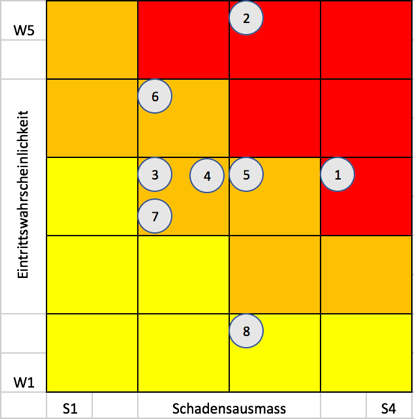


Abbildung 18: Risikograph

Tabelle 10: Risikobeschreibung

|  |  |
| --- | --- |
| Nr. | **Risikobeschreibung** |
| 1 | Verpassen des Abgabetermins |
| 2 | Umfang und Komplexität |
| 3 | Krankheitsfall |
| 4 | Begehung formaler Fehler |
| 5 | Verlust der Arbeit |
| 6 | Nichtbemerken von fehlerhaftem Code |
| 7 | Ausfallen der Infrastruktur |

### Risikograph nach den Massnahmen

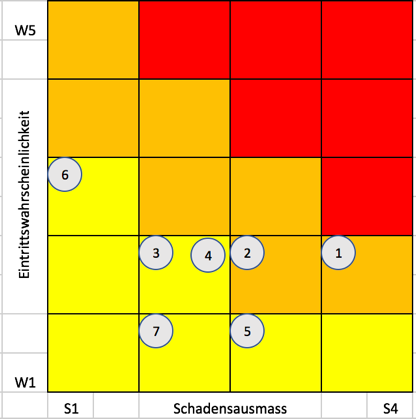


Abbildung 19: Risikograph

Tabelle 11: Risikobeschreibung

### Kurze Stellungnahmen zu den Risiken

Wie hier zu sehen ist, sind durch die getroffenen Massnahmen alle Risiken aus dem roten-Bereich herausgefallen. Während der Arbeit werden die gesamten genannten Massnahmen ausgeführt und das verbleibende Risiko wird akzeptiert.

## Lösungen suchen

Die Lösungssuche ist ein strukturierter und kreativer Prozess, in welchem nach möglichen Ansätzen gesucht wird, welche die Umsetzung der Systemziele unterstützen.

Der Entscheid über den Lösungsvorschlag (Variantenentscheid) schliesst die Lösungssuche ab.

### Benutzung via Terminal

Der Benutzer des Programmes Startet die Applikation direkt auf seinem lokalen Terminal. Insofern der Benutzer bereits Python3 installiert hat muss nichts weiteres installiert werden und das Spiel wäre sofort lauffähig.

Bei dieser Variante leidet jedoch der Visuelle Aspekt. Um dem etwas entgegen zu wirken könnte man etwas ASCII-Art verwenden. ASCII-Art ist wenn man ein Bild nur mithilfe von Schriftzeichen darstellt.

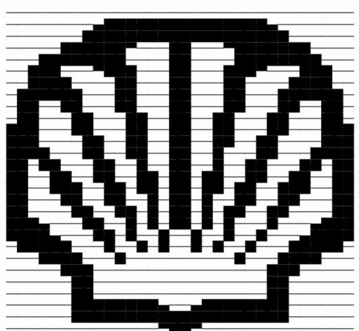


Abbildung : ASCII Art Shell[[6]](#footnote-6)

### Benutzung via Webseite

Der Benutzer kann die Applikation über eine Webseite zugreifen. Dies würde die Visuellen Aspekte verbessert, jedoch würde der Programmierumfang wesentlich erhöht werden und das Testing wird erschwert.

### Eingabe der „Choices“ via Integer

Die user-experience ist eventuell nicht so gut, da sich der Benutzer nicht persönlich angesprochen fühlt. Das Testing wird erleichtert und die Eingabe wäre sicherer.

### Eingabe der „Choices“ via Chars

Die Eingabe wirkt persönlicher. Jedoch kann es evtl. Probleme geben, wenn z.B. ein Wort falsch geschrieben wird, gross/kleinschreibe Fehler. Diese möglichen Probleme können jedoch mithilfe von einfachen Wörtern,

## Varianten

Im vorherigen Kapitel wurden jeweils 2 Varianten für 2 Verschiedene Aspekte des Projektes vorgestellt. Um nun Objektiv bewerten zu können welche der jeweiligen Varianten die beste Wahl ist, wird ein Variantenentscheid durchgeführt.

### Kriterien

Um die verschieden Varianten bewerten zu können werden Kriterien verwendet. Diese erschliessen sich aus den Anforderungen und unteranderem aus den möglichen Risiken welche bei der Machbarkeitsprüfung zum Vorschein kamen.

* Es muss schnell umsetzbar Sein
* Es muss gut Test bar sein
* Es soll benutzerfreundlich sein
* Es soll ansprechbar aussehen

Wie sie eventuell bemerkt haben Sind die zwei ersten Kriterien MUSS Kriterien und die zwei letzteren SOLL Kriterien. Die MUSS-Kriterien wurden direkt aus der Aufgabenstellung abgeleitet oder ergaben sich aus der Machbarkeitsprüfung. Die SOLL-Kriterien hingegen sind nicht zwingend währen jedoch gut.

Dies hat zur Folge das die MUSS Kriterien doppelt gewichtet werden.

### Gütestufen

Die Verschiedenen Varianten werden entsprechend den Kriterien Bewertet diese besitzen vier verschiedene Gütestufen. Diese sind wie folgt.

1. Kriterium nicht erfüllt
2. Kriterium etwas erfüllt
3. Kriterium erfüllt
4. Kriterium übertroffen

Da sowohl die Kriterien als auch die Gütestufen festgelegt wurden kann nun mit der Bewertung begonnen werden.

### Benutzung via Terminal

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Kriterium | Gewichtung | Bewertung | Resultat |
| 1 | Es muss schnell umsetzbar Sein | 2 | 4 | 8 |
| 2 | Es muss gut Test bar sein | 2 | 3 | 6 |
| 3 | Es soll benutzerfreundlich sein | 1 | 2 | 2 |
| 4 | Es soll ansprechbar aussehen | 1 | 1 | 1 |
| Total | | | 10 | 17 |

### Benutzung via Webseite

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Kriterium | Gewichtung | Bewertung | Resultat |
| 1 | Es muss schnell umsetzbar Sein | 2 | 1 | 2 |
| 2 | Es muss gut Test bar sein | 2 | 2 | 4 |
| 3 | Es soll benutzerfreundlich sein | 1 | 3 | 3 |
| 4 | Es soll ansprechbar aussehen | 1 | 4 | 4 |
| Total | | | 10 | 13 |

### Eingabe der „Choices“ via Integer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Kriterium | Gewichtung | Bewertung | Resultat |
| 1 | Es muss schnell umsetzbar Sein | 2 | 4 | 8 |
| 2 | Es muss gut Test bar sein | 2 | 4 | 8 |
| 3 | Es soll benutzerfreundlich sein | 1 | 3 | 3 |
| 4 | Es soll ansprechbar aussehen | 1 | 1 | 1 |
| Total | | | 12 | 20 |

### Eingabe der „Choices“ via Strings

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Kriterium | Gewichtung | Bewertung | Resultat |
| 1 | Es muss schnell umsetzbar Sein | 2 | 1 | 2 |
| 2 | Es muss gut Test bar sein | 2 | 3 | 6 |
| 3 | Es soll benutzerfreundlich sein | 1 | 3 | 3 |
| 4 | Es soll ansprechbar aussehen | 1 | 4 | 4 |
| Total | | | 11 | 15 |

### Vergleich

Jetzt werden die jeweiligen Bewertungen mit den Gewichtungen berechnet und anschliessend zusammengezählt.

|  |  |  |
| --- | --- | --- |
| Bereich | Variante | Total der Resultate |
| Benutzung via | Terminal | 17 |
| Webseite | 13 |
| Eingabe der „Choices“ via | Integer | 20 |
| String | 15 |

### Begründung

Wie der durchgeführte Variantenentscheid zeigt, ist es für dieses Projekt wichtiger, dass die Umsetzung möglichst simpel gehalten wird. Ein Beispiel dafür wann Ästhetische Aspekte eventuell wichtiger sind währe beim anzeigen allgemeiner Informationen mithilfe eines Charts.

Aus diesem Grund wird das Projekt Benutz via Terminal und die Eingabe-Parameter werden sich in Bezug auf Choices auf die Eingabe von Integer beschränken.

# Konzept

In der Konzepterarbeitung werden die Grundlagen für die Realisierung und Einführung eines Informatiksystems entwickelt.

Das Konzept wird schrittweise mit folgender Gliederung entwickelt.

* Wiederholung der bereits bekannten Grundlagen
* Einer allgemeinen Struktur des Spieles
* Eine detaillierte Struktur des Spieles
* Das Testkonzept mit den einzelnen Testfällen

## Allgemein

Nachfolgend werden nochmals die allgemeinen Voraussetzungen, Anforderungen erläutert.

### Grundlagen

In der Firma 89grad ist es üblich einige Probe IPAs als Vorbereitung für die offizielle IPA durchzuführen. Aus diesem Grund wurde der Lernenden aufgetragen eine Verkürzte IPA

**89grad Status-Quo**

* **Momentan ist noch kein Text Adventure vorhanden**

**Probe IPA-Ansatz**

* Es sollen min. 3 verschiedene Eingabeparameter vom Benutzerverlangt werden welche eine Auswirkung auf den Verlauf des Spiels haben
* Es sollen Pausen integriert werden welche den lese Fluss erleichtern sollen
* Das Spiel Soll mit Eingabe des eigenen Namens Personalisiert werden
* Das Produkt soll mittels whitebox-testing getestet werden
  + Jede dieser Inputparameter soll via Unittesting getestet werden

### Informationssicherheit und Datenschutz (ISDS)

ISDS wurde schon unter Punkt 9.4 Informationssicherheit und Datenschutz (ISDS) beschrieben. An dieser Stelle erfolgt nochmals die Aufzählung der wichtigsten Punkte.

* Da dieses Projekt keinerlei sensitive Daten enthält, wird von einer zusätzlichen Sicherung abgesehen.
* Die Dokumentenablage während des normalen Workflows auf der Lokalen Maschine eine weitere Sicherung erfolgt jeweils einmal täglich auf Github.

## Der Anfang der Geschichte

Die Geschichte um welche es sich in diesem Game handelt ist „Die kleine Meerjungfrau“ von Hans Christian Andersen. Hans Christian Andersen kommt aus Dänemark und publizierte diese buch im Jahr 1837.

Die Geschichte handelt von einer jungen Meerjungfrau. Sie ist die Tochter des Meerkönigs, besitzt fünf ältere Schwestern, hat eine wunderschöne Stimme und kann unglaublich gut Singen und sie wünscht sich nichts mehr als die Oberfläche zu besuchen.

Als sie an ihrem 16ten Geburtstag endlich an die Oberfläche darf sieht sie einen Menschenprinzen und verliebt sich schlussendlich. Als sein Schiff in einem Sturm untergeht rettet sie ihn. Voller liebe und Hoffnungslosigkeit besucht sie die „Seehexe“ welche im Gegenzug für ihre Stimme ihre Flossen in beine verwandelt. Jedoch muss sie den Prinzen dazu bringen sie zu Lieben und zu heiraten ansonsten wird sie am Tag nachdem er eine andere geheiratet hat sterben und zu Schaum zerfallen. Zusätzlich wird jeder schritt den sie an Land macht sich anfühlen als würde sie auf Messern stehen.

Dies war nur der erste Teil der Geschichte falls Sie die gesamte gerne lesen würden hänge ich diese sowohl auf Deutsch als auch auf Englisch im Anhang an.

## Architektur

## Aktivitätsdiagramm

Das Aktivitätsdiagramm soll den Groben Ablauf der Geschichte Zeigen.

Die Verschiedenen Entscheidungen sowie Enden werden grob beschrieben der Gesamte Ablauf mit dem Fertigen Text befindet sich in der Realisierungsphase.



|  |  |
| --- | --- |
| Elemente des Aktivitätsdiagramm |  |
|  |  |
|  |  |
|  |  |
|  |  |

## Anwendungsfall

Bisher wurden die Standard Charts spezifisch für jeden Kunden erstellt. Mit einer Chart Library werden die Standard Charts nicht mehr spezifisch für den Kunden entwickelt. Dadurch wird das sogenannte «look and feel» der Projekte einheitlich und die Corporate Identity wird ausgeprägter.

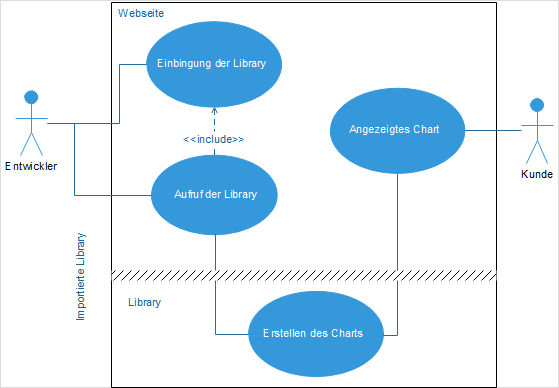


Abbildung 13: Anwendungsfälle

Bei diesem Anwendungsfall geht es im Wesentlichen darum wie der Entwickler ein Chart erstellt. Dafür muss er als erstes die Library in seinem Sourcecode einbinden. Anschliessend muss er die Library aufrufen und die korrekten Parameter bzw. Properties übergeben.

Daraufhin wird in der Chart Library das Chart erstellt. Das erstellte Chart wird nun auf der Webseite angezeigt, auf welcher der Kunde dieses sehen kann.

## Machbarkeitsprüfung Konzeptphase

1. Wirtschaftliche Machbarkeit
2. Technische Machbarkeit
3. Zeitliche Umsetzung
4. Ressourcen und Verfügbarkeit

**Empfehlung**

Im Vergleich zu der Initialisierungsphase hat sich an der Machbarkeit des Projektes nichts geändert. Aus diesem Grund wird weiterhin empfohlen das Projekt durchzuführen.

## Testkonzept

### Grundlagen

Das Testing besteht im Allgemeinen aus zwei Bereichen. Dem Frontend und dem Backend.

Das Frontend wird manuell kontrolliert und für das Backend wird Unit Testing verwendet.

### Test Szenario

Die Tests werden während des Programmierens gleichzeitig mit dem eigentlichen Code implementiert.

### Testrahmen

Die Tests werden lokal auf dem Arbeitsrechner der enersis ausgeführt.

Es bedarf keiner speziell gesicherten Testumgebung, weil die übermittelten Daten zum momentanen Zeitpunkt nicht produktionsrelevant sind.

### Testvorgehen

Die Unit Tests werden mithilfe von Mocca und Chai implementiert simultan mit dem eigentlichen Code implementiert.

Die Durchführung der Testsfälle geschieht automatisch. Dies geschieht durch die Eingabe des Befehls: npm run test.

### Testmethode

**Unit-Test**[[7]](#footnote-7): Mittels Unit-Test (auch Modul- oder Komponententest genannt) werden einzelne Komponenten einer Anwendung geprüft. Unit-Tests werden oft im Rahmen der Softwareentwicklung vorgenommen.

### Testfälle

|  |  |  |
| --- | --- | --- |
| **Nr.** | **Aktion** | **Erwartetes Ergebnis** |
| **1TF** | Aufruf des Barcharts. Nur mit Angabe des Typs. | Fehlermeldung |
| **2TF** | Aufruf des Barcharts. Mit Angabe des Typs und Daten mit falscher Datenstruktur. | Fehlermeldung |
| **3TF** | Aufruf des Barcharts. Mit Angabe des Typs und Daten mit korrekter Datenstruktur. | Anzeigen eines Barcharts |
| **4TF** | Aufruf eines Charts. Mit Angabe des eines falschen Typs. | Fehlermeldung |
| **5TF** | Properties des Titels : «Test Titel» (String). | Titel wurde im entsprechenden Property gespeichert. |
| **6TF** | Properties des Titels : 9 (Zahlenwert) | Fehlermeldung |
| **7TF** | Daten mit korrekter Datenstruktur | Daten wurde im entsprechenden Property gespeichert. |
| **8 TF** | Daten mit falscher Datenstruktur | Fehlermeldung |
| **9 TF** | Als Zielelement wird «#DIVId» (String) übergeben. | Wenn ein entsprechendes HTML Element gefunden wurde, wird das Element in dem entsprechenden Property gespeichert. |
| **10 TF** | Als Zielelement wird «.DIVClass» (String) übergeben. | Fehlermeldung |
| **11 TF** | Als Zielelement wird «#DIVId» (String) übergeben. | Wenn kein entsprechendes HTML Element gefunden wurde, bekommt man eine Fehlermeldung. |
| **12 TF** | Als Zielelement wird der Zahlenwert 9 übergeben. | Fehlermeldung |
| **13 TF** | Propertie des Color: [«#fff»] (String in einem Array) | Color wurde im entsprechenden Property gespeichert. |
| **14 TF** | Propertie des Color: «#fff» (String) | Fehlermeldung |
| **15 TF** | Propertie des Color: [«#fff», «#000», «#abcabc»](String in einem Array) | Color wurde im entsprechenden Property gespeichert. |
| **16 TF** | Propertie des height: 10 (Zahlenwert) | height wurde im entsprechenden Property gespeichert. |
| **17 TF** | Propertie des height: «test» (String) | Fehlermeldung |
| **18 TF** | Propertie des width: 10 (Zahlenwert) | width wurde im entsprechenden Property gespeichert. |
| **19 TF** | Propertie des width: «test» (String) | Fehlermeldung |
| **20 TF** | Propertie des TransitionTime: 100 (Zahlenwert) | TransitionTime wurde im entsprechenden Property gespeichert. |
| **21 TF** | Propertie des TransitionTime: «test» (String) | Fehlermeldung |
| **22 TF** | Propertie des TransitionType: sinus | TransitionType wurde im entsprechenden Property gespeichert. |
| **23 TF** | Propertie des TransitionTime: «test» (String) | Fehlermeldung |
| **24 TF** | Propertie des Legend: false | Legend wurde im entsprechenden Property gespeichert. |
| **25 TF** | Propertie des Legend: «test» (String) | Fehlermeldung |
| **26 TF** | Propertie des Scale: 10 (Zahlenwert) | Scale wurde im entsprechenden Property gespeichert. |
| **27 TF** | Propertie des Scale: «test» (String) | Fehlermeldung |
| **28 TF** | Propertie des Radius: false(Boolean) | Legend wurde im entsprechenden Property gespeichert. |
| **29 TF** | Propertie des Legend: «test» (String) | Fehlermeldung |

Tabelle 18: Testfälle

### Testziele

Alle Eingabeparameter wurden getestet. Und es ist sichergestellt, dass Veränderungen während der Laufzeit möglich sind.

# Realisierung

Die Phase Realisierung dient zur Erstellung des Systems und schafft die Voraussetzungen für die nachfolgende API.

## Template

In diesem Projekt wird das Wort Template zur Repräsentierung einer von einem anderen Entwickler aufgesetzten Projektstruktur verwendet.

Diese Projektstruktur enthält bereits die benötigten Libraries und mit diesem Template wird sichergestellt, dass mit diesen allgemeinen Arbeiten keine Zeit verschwendet wird. Das Projektstruktur – Template stellt ausserdem sicher, dass alle erforderlichen Dateien und Programme für die Kompilierung und zur Laufzeit bereitstehen.

In den folgenden Unterkapiteln wird auf die einzelnen Komponenten des Templates eingegangen.

### C:\Users\nicol\AppData\Local\Microsoft\Windows\INetCacheContent.Word\Ecma_RVB-003.jpgECMAScript 6

ECMAScript 6 ist die neuste offizielle Version von ECMAScript[[8]](#footnote-8), der offizielle Name lautet ECMAScript 2015 und kurz ES6. ECMAScript ist ein Standard der Programmiersprache JavaScript.

Abbildung 25: ecma international

Andere Standards wären z.B. JScript oder ActionScript.

Entwickler benutzen ECMAScript vor allem zur clientseitigen Webentwicklung, jedoch wird es auch immer häufiger zur serverseitigen Programmierung mit Node.js verwendet.

**Neue Features in ES 6**

In der neuen ECMAScript Version wurden einige neue Features hinzugefügt, unter anderem wären das:

* Klassen
  + Getter/ Setter Methoden
* Module
* Generatoren
* Typed Arrays
* Destructuring Assignments

Die neue ECMAScript Version wird jedoch noch nicht von allen Browsern fehlerfrei unterstützt. Deswegen wird für die Verwendung von ECMAScript 6 Code ein Compiler benötigt, welcher den ES6 Code in ES5 Code kompiliert.

### C:\Users\nicol\AppData\Local\Microsoft\Windows\INetCacheContent.Word\babel-1.pngBabel

In diesem Projekt wird Babel[[9]](#footnote-9) als Compiler verwendet. Babel kompiliert ECMAScript 6 in ECMAScript 5.

In der nachfolgenden Abbildung sehen sie ein Beispiel, welches von der offiziellen Seite übernommen wurde.

Abbildung 26: BABEL



Abbildung 16: Babel Beispiel

Wie anhand des Beispiels erkennbar ist, transformiert Babel nur die Syntax, währendem die eigentliche Funktionalität bestehen bleibt.

Ausserdem wird die kompilierte Syntax gemapt, so dass bei der Fehlersuche ersichtlich ist, in welcher Zeile sich der Originalcode befindet.

### C:\Users\nicol\AppData\Local\Microsoft\Windows\INetCacheContent.Word\EsLint.pngESLint

ESLint[[10]](#footnote-10) ist ein Tool, welches bei der allgemeinen Code Formatierung und Problemsuche hilft. EsLint besitzt diverse Regelungen, welche alle separat an- und ausgeschaltet werden können. Diese Hinweise tragen zu einem besseren Code und damit zur Verhinderung von Fehlern bei.

Nach der Definition von Wikipedia[[11]](#footnote-11): «Lint ist ein Computerprogramm zur statischen Code-Analyse der Quelltexte von Computerprogrammen. Das Verb linten für   
das Durchführen der statischen Code-Analyse hat sich davon   
unabhängig als allgemeine Bezeichnung etabliert.»

Abbildung 28: ESLint

Hier noch ein paar übersetzte Angaben von der offiziellen Webseite bezüglich der Benutzung von ESLint[[12]](#footnote-12).

* Jede Regel kann: abgestellt, nur eine Warnmeldung oder eine Fehlermeldung ausgeben.
* Zusätzliche Regeln können während der Laufzeit hinzugefügt werden.
* ESLint fördert keinen bestimmten Codierungsstil

### Unit Testing

Nach der Definition von Wikipedia wird ein Modultest (auch von englisch unit test als Unittest oder als Komponententest bezeichnet) in der Softwareentwicklung angewendet, um die funktionalen Einzelteile (Units) von Computerprogrammen zu testen, d.h., sie auf korrekte Funktionalität zu prüfen.

#### C:\Users\nicol\AppData\Local\Microsoft\Windows\INetCacheContent.Word\mocha-chaijs.pngMocha und Chai

Mocha and Chai[[13]](#footnote-13) sind zwei der am häufigsten verwendeten Libraries in Bezug auf JavaScript basiertem Unit Testing.

Hierbei wird mit Mocha die eigentliche Testing Umgebung hinzugefügt und Chai enthält einige hilfreiche Funktionen, welche das Schreiben der Tests und dessen Auswertungen wesentlich einfacher machen. Die geschriebenen Tests können sowohl auf Node.js als auch im Browser ausgeführt werden.

Abbildung 29: mocha und chai

Ein Beispiel wie ein realisierter Chai Test aussehen könnte.

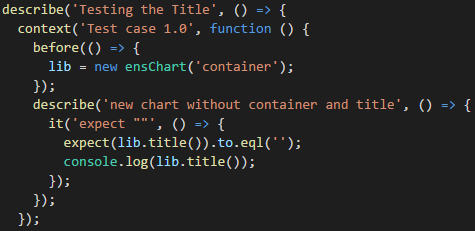
Wie auf der oberen Abbildung zu sehen ist, besteht ein Test aus 5 Hauptbausteinen, wobei einer davon optional ist.

Abbildung 30: Testing des Titels

|  |  |
| --- | --- |
| Describe | Wird zur Beschriftung des durchgeführten Tests verwendet. |
| Context (optional) | Context ist eine weitere Beschriftungsmöglichkeit und wird hier zur Strukturierung des Codes verwendet. Durch den Einsatz der context Funktion können mehrere Tests in einem describe ausgeführt werden. |
| Before | Wird ausgeführt bevor die eigentliche Überprüfung stattfindet. |
| It | Beschreibt was erwartet wird. |
| expect | Hier findet der eigentliche Test statt. Wobei erwartet wird, dass die ausgeführte Funktion true zurück liefert. |

Tabelle 19: Hauptbausteine von Chai[[14]](#footnote-14)

Hierbei ist zu beachten, dass der Vergleich der beiden Values, in diesem Fall ob der Titel wirklich den Standardwert zurückgibt, mit method chaining ungesetzt wurde. Die Funktion Namen von Chai wurden bewusst gewählt und die Leserlichkeit des Codes zu verbessern.

Die mit Chai geschriebenen Tests wurden mithilfe von Mocha getestet. Weil das Testing im Browser durchgeführt wurde, musste eine zusätzliche Dashboard Seite (test.html) umgesetzt werden.

### C:\Users\nicol\AppData\Local\Microsoft\Windows\INetCacheContent.Word\EdrawMax_2017-05-04_21-18-42.pngVom Source Code zur fertigen Library

Tabelle 20: Vom Source Code zur fertigen Library

Die Chart Library wird in ECMAScript 6 (ES6) umgesetzt. Da ES6 zum jetzigen Zeitpunkt nicht mit allen Browsern kompatibel ist wird Babel verwendet. Babel konvertiert die ES6 Files in ES5 Files und macht diese somit mit den momentan verwendeten Browsern kompatibel.

Um schlussendlich nur ein File einbinden zu müssen, werden die Source Files, mit all ihren Abhängigkeiten, mithilfe von Webpack in 2 Versionen eines Files gespeichert.

Einer library.js Version und einer library.min.js Version.

#### Webpack

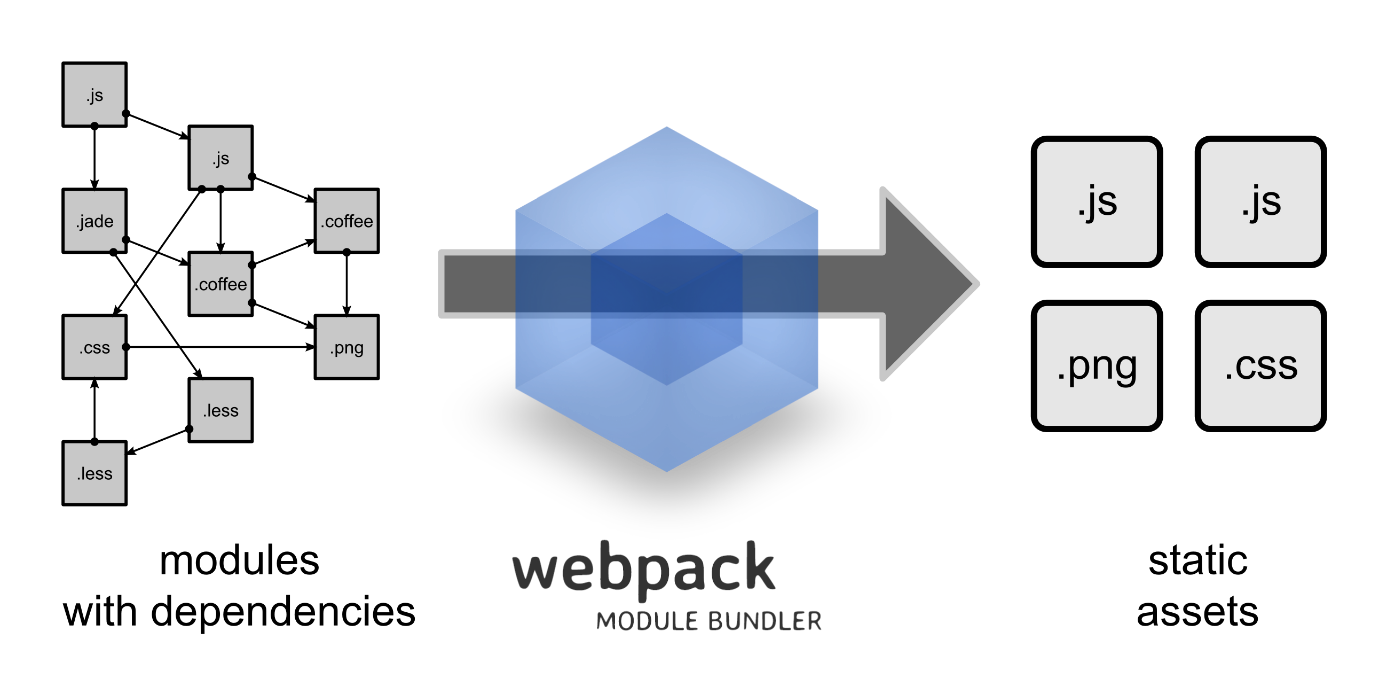


Tabelle 21: Was ist webpack[[15]](#footnote-15)

In normalen Projekten muss jedes einzelne js File einzeln nacheinander in das gewünschte File eingebunden werden. Da dies sehr umständlich ist und einige unnötige Fehler durch z.B. Einlesen der Files in der falschen Reihenfolge verursachen kann, werden sogenannte module bundeler, wie in diesem Projekt webpack, verwendet.

Webpack bundelt alle Files mit dessen Abhängigkeiten in ein paar wenige Files. Dadurch müssen deutlich weniger Files eingelesen werden. Dies macht die Verwendung wesentlich angenehmer, simpler und fehlerfreier. Der grösste Nachteil ist, dass sehr viele unnötige assets mitgeladen werden, die bei einem Projekt möglicherweise bereits eingebunden wurden.

## Klassendiagramm

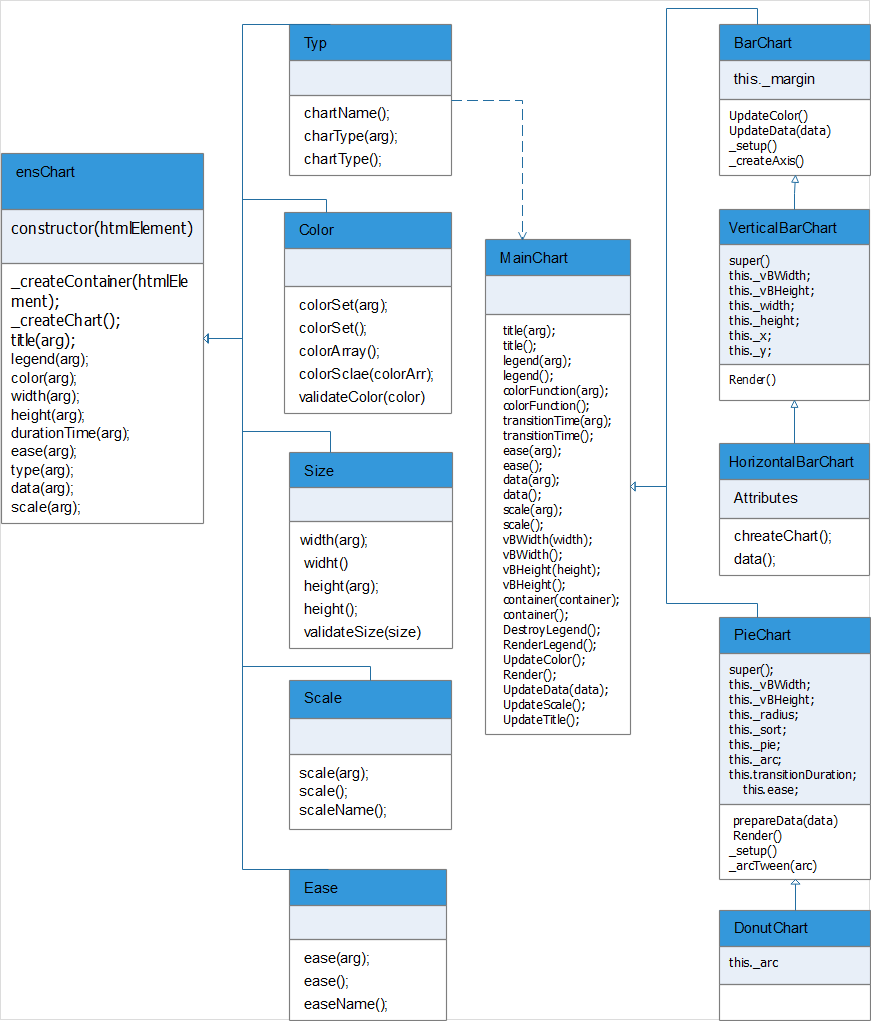


Abbildung 20: Klassendiagramm

Wie anhand der obigen Abbildung des Klassendiagrammes ersichtlich ist, blieb die Grundstruktur gleich, jedoch kamen einige Funktionen dazu bzw. die bereits beschriebenen Funktionen wurden verfeinert.

## Entwicklung

### ensChart Klasse

#### Zielelement (HTML-DIV-Element)

\_createContainer(htmlElement) {

if (htmlElement) {

if (typeof (htmlElement === 'string')) {

if(htmlElement.startsWith("#")){

let split = htmlElement.split("#");

htmlElement = split[1];

}

htmlElement = document.getElementById(htmlElement);

}

if (htmlElement instanceof HTMLElement) { … }

else {

console.error('Selection is neither ElementId nor HTMLElement');

return;

}

}

else {

console.error('You have to give a container');

return;

}

Diese Funktion sucht das HTML-DIV-Element, über dessen angegebene Id, in welchem das Chart angezeigt werden soll.

Der obige Code überprüft zuerst, ob der Parameter mitgeliefert wurde und ob es sich um den Typ String handelt. Anschliessend wird überprüft, ob der String mit einer Raute beginnt. Wenn dies der Fall ist, wird diese entfernt. Mit dem evtl. angepassten String wird anschliessend nach dem entsprechenden HTML-DIV-Element gesucht.

#### Data (array of objects)

data(arg) {

if (!arguments.length) {

return this.\_data ? this.\_data : [];

} else if (Array.isArray(arg)) {

this.\_data = arg;

this.\_chart.data = this.data();

return this;

}

console.error('Data is not an array');

return this;

}

Da die Datenstrukturen bei Charts sehr unterschiedlich sein können, wird in der Mainklasse nur überprüft, ob die gelieferten Daten einem Array entsprechen. Zusätzliche Validierungen werden erst in den betreffenden Charts durchgeführt.

#### Titel (string)

title(arg) {

if (!arguments.length) {

return this.\_title ? this.\_title : '';

} else if (typeof arg === 'string') {

this.\_title = arg;

this.\_chart.title = this.title();

return this;

}

console.error('Title not a String');

return this;

}

Mit der Funktion Titel wird ein Titel für das Chart erfasst. Dieser wird anschliessend über dem Chart angezeigt. Der Default Wert ist ein leerer String.

Zu Beginn wird überprüft, ob ein Parameter mitgeliefert wurde. Ist dies nicht der Fall, wird entweder der Default Value oder der zuvor gesetzte Value zurückgegeben.

Falls ein Parameter angegeben wurde, wird überprüft, ob dieser zum Typ String gehört. Wenn dies der Fall ist, wird der mitgelieferte Parameter als neuer Value gesetzt.

#### Legende (bool of visibility)

legend(arg) {

if (!arguments.length) {

return this.\_legend ? this.\_legend : false;

} else if (typeof arg === 'boolean') {

this.\_legend = arg;

this.\_chart.legend = this.legend();

return this;

}

console.error('Legend is not a Boolean');

return this;

}

Mit der Funktion Legende wird festgelegt, ob eine Legende zu den Datensätzen erstellt werden soll. Diese wird anschliessend unter dem Chart angezeigt. Der Default Wert ist der boolescher wert false.

Zu Beginn wird überprüft, ob ein Parameter mitgeliefert wurde. Ist dies nicht der Fall, wird entweder der Default Value oder der zuvor gesetzte Value zurückgegeben.

Falls ein Parameter angegeben wurde, wird überprüft, ob dieser zum Typ Boolean gehört. Wenn dies der Fall ist, wird der mitgelieferte Parameter als neuer Value gesetzt.

#### TransitionTime (number)

durationTime(arg) {

if (!arguments.length) {

return this.\_durationTime ? this.\_durationTime : 1000;

} else if (isFinite(arg)) {

this.\_durationTime = Math.abs(Number(arg));

this.\_chart.transitionTime = this.durationTime();

return this;

}

console.error('Legend is not a (positive) Number ');

return this;

}

Mit der Funktion durationTime wird festgelegt wie lange die Animation der Charts dauern soll. Der Default Wert ist der numerische Wert 1000. Hierbei ist zu beachten, dass der Value in Millisekunden angegeben wird.

Zu Beginn wird überprüft, ob ein Parameter mitgeliefert wurde. Ist dies nicht der Fall, wird entweder der Default Value oder der zuvor gesetzte Value zurückgegeben.

Falls ein Parameter angegeben wurde, wird überprüft, ob dieser endlich ist bzw. ob es eine reale Zahl ist. Wenn dies der Fall ist, wird der mitgelieferte Parameter als neuer Value gesetzt.

### Ausgelagerte Klassen

Da einige der Values weitere, komplexere Validierungen erfordern, wurden diese, wie bereits erwähnt, in separate Klassen ausgelagert, Jedoch erfolgt jeweils eine einfache Validierung sowie das Zurückgeben der Parameter über die Mainklasse.

#### Typ (string)

In der Main Klasse:

type(arg) {

if (!arguments.length) {

return this.\_cConfigType.chartName;

} else if (typeof arg === 'string') {

this.\_cConfigType.chartType = arg;

this.\_createChart();

return this;

}

console.error('Chart type is not a string');

return this;

}

#### Farben (string oder array mit Farbwerten)

In der Main Klasse:

color(arg) {

if (!arguments.length) {

return this.\_cConfigColor.colorArr;

} else if (typeof arg === 'string' || Array.isArray(arg)) {

this.\_cConfigColor.colorSet = arg;

this.\_chart.colorFunction = this.\_cConfigColor.colorSet;

return this;

}

console.error('Color is not a string or array');

return this;

}

#### Grösse (string in px oder number in px)

In der Main Klasse:

width(arg) {

if (!arguments.length) {

return this.\_cConfigSize.width;

} else if (isFinite(arg) || typeof arg === 'string') {

this.\_cConfigSize.width = arg;

if (this.\_chartContainer) {

this.styling.width = this.\_cConfigSize.width;

this.\_chartContainer.attr('style', "height: " + this.styling.height + ";" + " width: " + this.styling.width + ";");

}

return this;

}

console.error('width not a number or string');

return this;

}

#### Easing (string)

In der Main Klasse:

ease(arg) {

if (!arguments.length) {

return this.\_cConfigEase.easeName;

} else if (typeof arg === 'string') {

this.\_cConfigEase.ease = arg;

this.\_chart.ease = this.\_cConfigEase.ease;

return this;

}

console.error('ease not a string');

return this;

}

## Testprotokoll

Die Testprotokolle wurden in englischer Sprache verfasst gemäss dem Punkt sechs der «Richtlinien zur Softwareentwicklung» von enersis suisse AG: «Dokumentationen sind in englischer Sprache zu verfassen und sollen nach Möglichkeit mit grafischen Darstellungen zum besseren Verständnis ergänzt werden.»

Während der Realisierungsphase zeigte sich, dass zusätzliche Testfälle zu den in der Konzeptphase definierten Testfällen nötig waren. Diese wurden entsprechend ergänzt. Die folgenden Abbildungen stellen die Unit Tests dar, welche mit Mocha und Chai realisiert wurden.

### Constructor

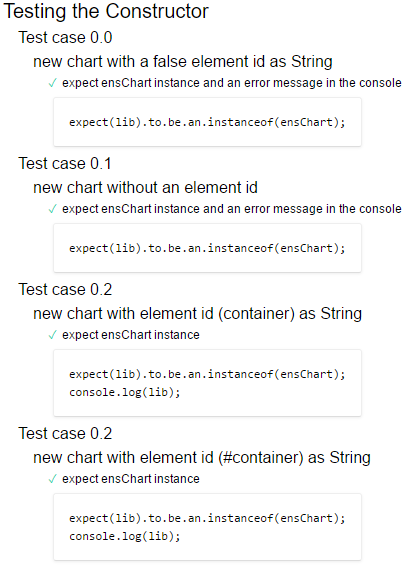


Abbildung 21: testing the Constructor

### Title

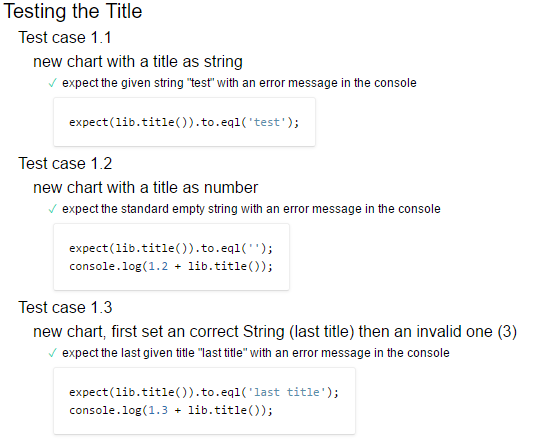


Abbildung 22: Testing the Title

### Legend

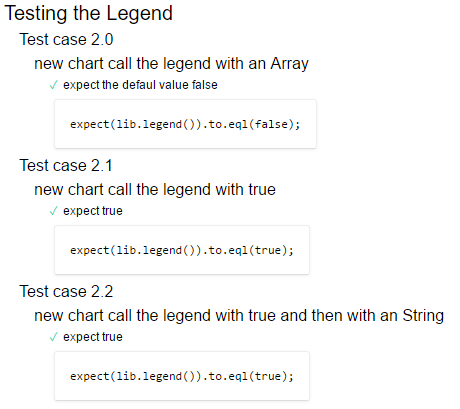


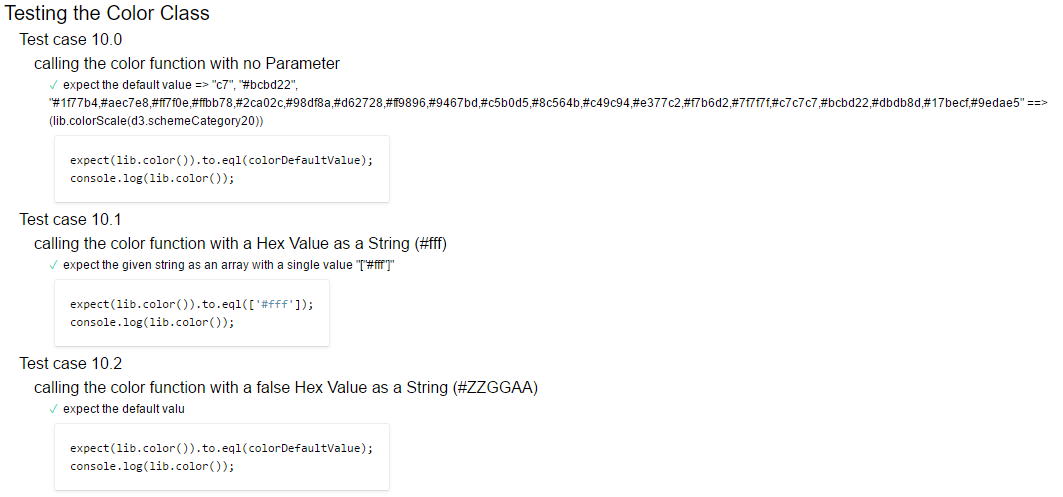
Abbildung 23: Testing the Legend

### DurationTime



Abbildung 24: Testing the DurationTime

### Color Klasse



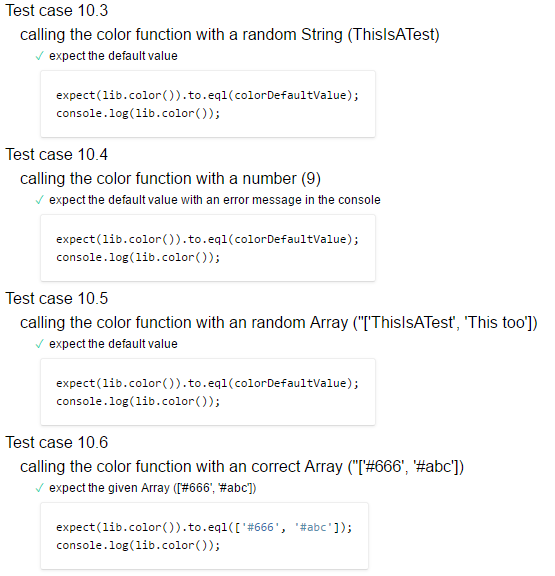


Abbildung 25: Testing the Color Class

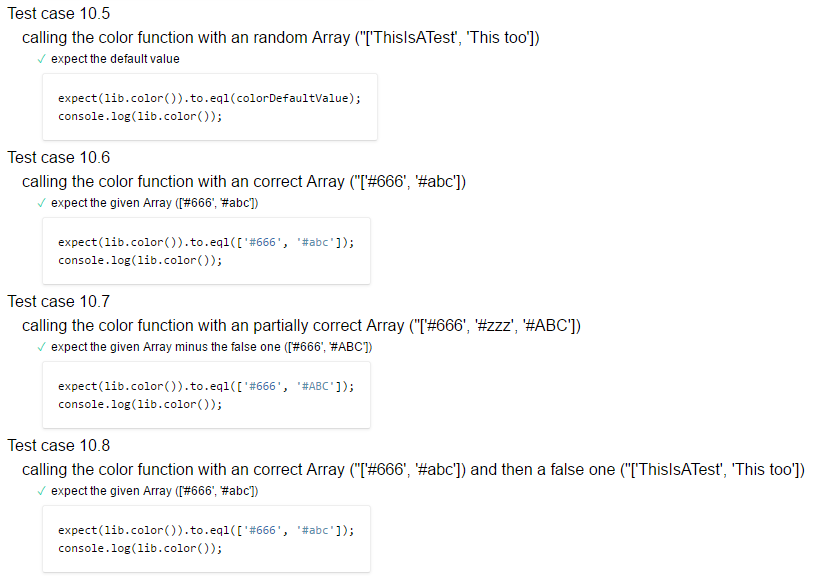


Abbildung 26: Testing the Color Class

### Ease Klasse

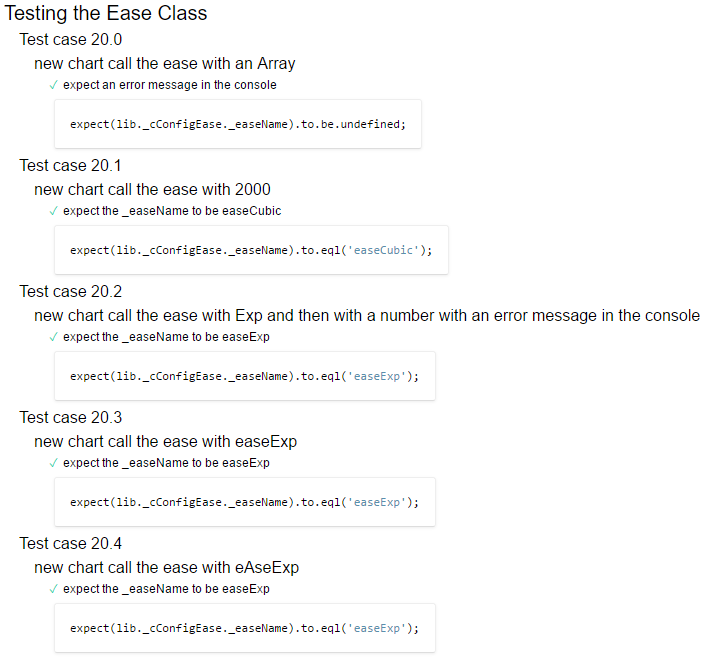


Abbildung 27: Testing the Ease Class

### Scale Klasse

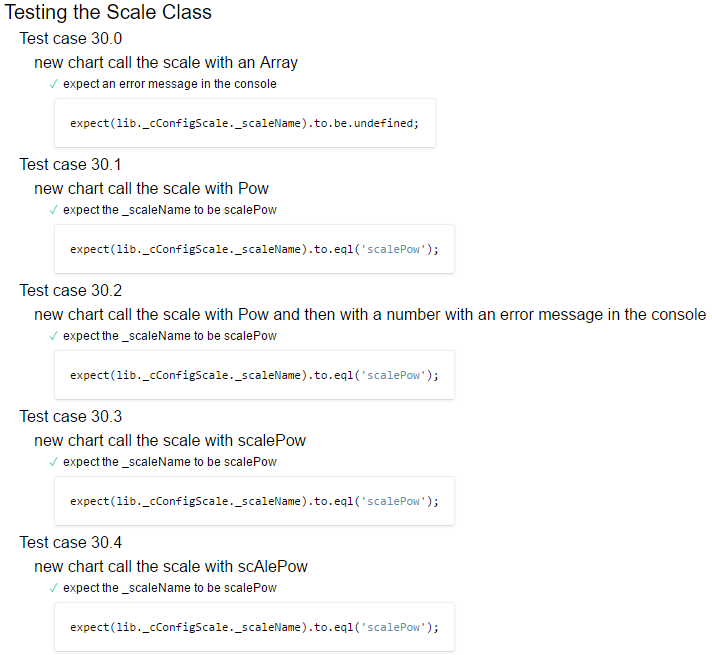


Abbildung 28: Testing the Scale Class

### Size Klasse

#### width



Abbildung 29: Testing the Size Class (width)

#### height

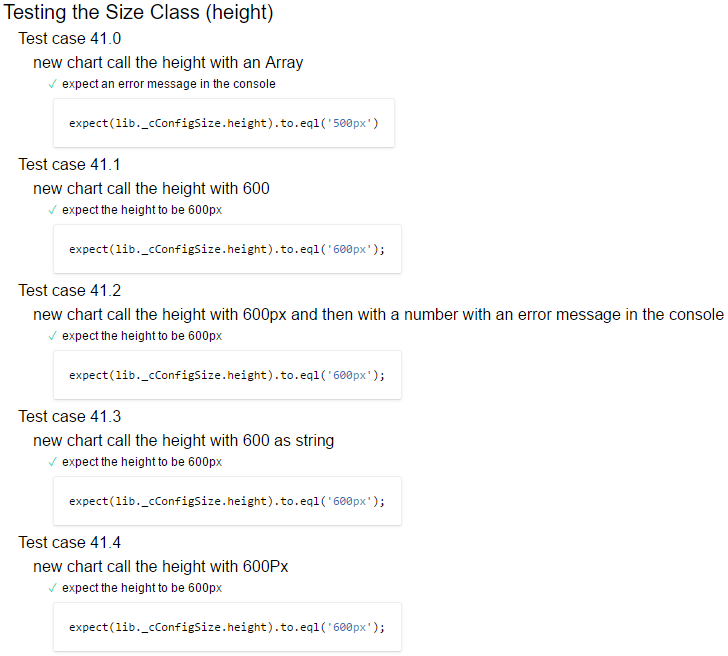


Abbildung 30: Testing the Size Class (height)

### Typ Klasse

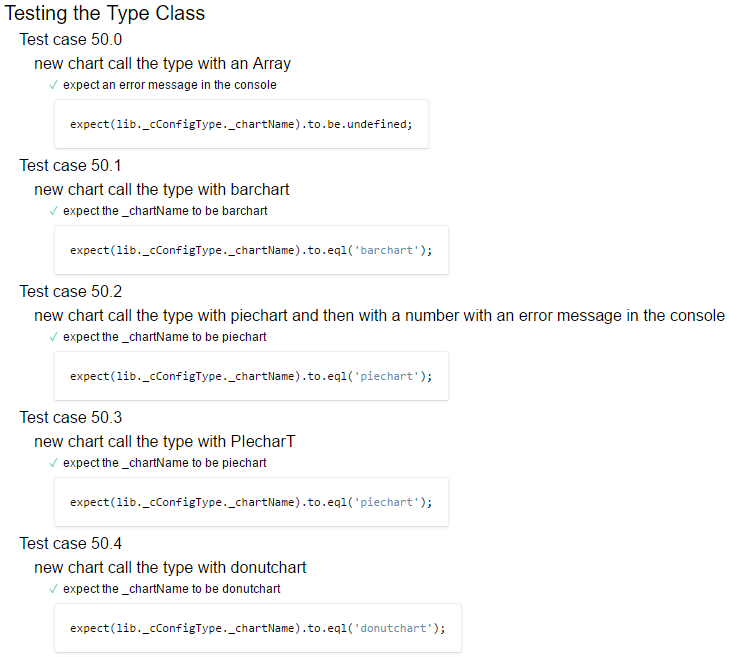


Abbildung 31: Testing Type Class

## Vergleich der Anforderung und der Umsetzung

### Funktionale Anforderungen

Funktionale Anforderungen beschreiben WAS das System bzw. Produkt leisten soll. Die funktionalen Anforderungen beschreiben somit die gewünschten Funktionalitäten des Systems, aber auch dessen Daten oder Verhalten.

* Die Chart Library soll folgende zwei Diagrammtypen zur Verfügung stellen:
  + Donut-Chart
  + Barchart
* Folgende Eigenschaften der beiden Charts sollen allgemein konfigurierbar sein:
  + Titel (string)
  + Typ (string)
  + Daten (array of objects oder analog den Anforderungen aus den bestehenden Charts)
  + Farben (string oder array mit Farbwerten)
  + Zielelement (HTML-DIV-Element)
  + Grösse (string in px oder number in px)
  + Legende (bool of visibility)
  + TransitionTime (number)
  + Easing (string)
* Die Chart Library soll als API zur Verfügung stehen, d.h. es gibt eine definierte Schnittstelle, wo entsprechende Parameter übergeben werden können.
* Die an die API übergebenen Parameter sollen validiert werden.
* Die Chart Library muss erweiterbar sein. Hierzu soll eine zusätzliche (frei definierbare) Eigenschaft, welche nur für eines der beiden Charts in der Library gilt, zur Erzeugung des Charts zur Verfügung stehen.

### Nicht funktionale Anforderungen

Nicht funktionale Anforderungen beschreiben WIE GUT das System etwas leisten soll und decken im Prinzip die qualitativen Aspekte und Bedingungen des Systems oder Funktionalitäten ab (z.B. Erweiterbarkeit).

* Die Chart Library soll als Java Script Dokument(e) zur Verfügung stehen, so dass diese zukünftig in Projekte von enersis integriert werden kann.
* Für die Darstellung und Manipulation der Daten in Form von Charts soll im Wesentlichen die D3.js Library genutzt werden.
* Für die Charts sind bestehende SVG-Templates zu nutzen.
* Die beiden oben genannten Charts
  + Donut-Chart und
  + Barchart

sollen mit Hilfe der API / Library als Beispielimplementation entwickelt und visualisiert werden.“.

## API Dokumentation vorbereiten

Eine API (ausgeschrieben[[16]](#footnote-16): application programming interface deutsch «Programmierschnittstelle» oder «Schnittstelle zur Anwendungs-Programmierung») in diesem Zusammenhang ist es die Schnittstelle zwischen dem Entwickler und der Library welche dokumentiert wird.

Diese API Dokumentation soll die Grundlage liefern, damit die enersis Entwickler die Chart Library ohne Probleme verwenden und erweitern können. Zudem soll die Dokumentation Klarheit bei möglichen Fehlern schaffen.

Die API wird ebenso wie die Kommentare im Source Code in englischer Sprache verfasst gemäss dem Punkt sechs der «Richtlinien zur Softwareentwicklung» der enersis suisse AG.

### Struktur der API Dokumentation

Für jeden Eingabe Parameter werden folgende Informationen beschrieben.

* Die Funktion die aufgerufen wird als Titel
* Eine kurze Beschreibung und die möglichen Eingabeparameter
* Ein simples Beispiel

Zusätzlich wird beschrieben, welche Schritte getätigt werden müssen, um ein neues Chart hinzuzufügen.

# API documentation ensChart

# General Information

This Library was realized with the “Webpack library starter”[[17]](#footnote-17)[[18]](#footnote-18) from Krasimir Tsonev. This Template features useful Libraries like:

* Webpack 2 based.
* ES6 as a source.
* Exports in a umd format so your library works everywhere.
* ES6 test setup with Mocha and Chai.
* Linting with ESLint.

# Getting started

Include the downloaded js File in your web page. With the script Tag. It should look something like this.

<script src="./lib/ensChart.js"></script>

<div id=”testId”></div>

<script src="./js/test.js"></script>

After you have done that you now can access the Library.  
Ensure the given path fits to the location of the Library file.

## Initializing the Library

Opening your test.js file you can now initialize the Library by writing something like this.

Lib = new ensChart(‘testId’);

Please pay attention to the parameter you’re giving. Because that’s the Id name of the div you want the chart to appear in.

You can give the Id with or without an #. But you must give an ID a class name won’t work.

# The different possible parameters you can set

All but a few of these are completely optional only the container parameter and the data are mandatory.

## Lib.title(*arg*)

The title function will be used to define the Title that will be shown at the top of the chart.

Lib.title('default Title');

The only type that is allowed is a string. If you try to set something else you will only get the default value: ‘’ and an error message in the console.

## Lib.data(*arg*)

The data function will be used to define the data that will be used in the chart.

Lib.data([{value: 6, title: ‘First Value’}, { value: 9, title: ‘Second Value’}, {…}]); || Lib.data([{ xAxis: 1, yAxis: 9, name: 'First Part' }, { xAxis: 2, yAxis: 4, name: 'Second Part' }, { xAxis: 3, yAxis: 9, name: 'Third Part' }])

The data is an array of objects. In general, it only needs the Title and the Value Properties to work correctly.

But for example, in the bar chart you need the xAxis value as well.

## Lib.legend(*arg*)

The legend function will be used to define if a Legend will be created.

Lib.legend(true);

The only type that is allowed is a boolean. If you try to set something else you will only get the default value: false and an error message in the console.

## Lib.color(*arg*)

The color function will be used to define what colors the charts will use. You can either give a String or an Array of Strings.

|  |
| --- |
| **Note: It is important that the strings are Hex values, but it is not important if you write an # or not. Also it is not case sensitive, that means it is not important if you write “#aabb00”, “aabb00”, “AB0”, “#AB0” or even “#Ab0”.** |

Lib.color(['#666', '#abc', ‘#4Ad’]); || Lib.color(‘a0a1a2’);

The only Type that is allowed is a String or an Array of Strings. If you try to set something else you will only get the default value:

d3.schemeCategory20  ["#1f77b4", "#aec7e8", "#ff7f0e", "#ffbb78", "#2ca02c", "#98df8a", "#d62728", "#ff9896", "#9467bd", "#c5b0d5", "#8c564b", "#c49c94", "#e377c2", "#f7b6d2", "#7f7f7f", "#c7c7c7", "#bcbd22", "#dbdb8d", "#17becf", "#9edae5"]

and an error message in the console.

## Lib.ease(*arg*)

The ease function will be used to define what kind of animation the chart will have. The String you have to give will execute a d3 function, therefore you only can give the name of a easing d3 function.

On this Site, you have some Examples: http://bl.ocks.org/hunzy/9929724.

Lib. durationTime(3000);

The only Type that is allowed is a positive number. If you try to set something else you will only get the default value: 1000 and an error message in the console.

## Lib.durationTime(*arg*)

The durationTime function will be used to define how long the animation of the chart will take. The number’s unit is set to milliseconds (ms).

Lib. durationTime(3000);

The only Type that is allowed is a positive number. If you try to set something else you will only get the default value: 1000 and an error message in the console.

## Lib.scale(*arg*)

The durationTime function will be used to define how long the animation of the chart will take. The number’s unit is set to milliseconds (ms).

Lib. durationTime(3000);

The only Type that is allowed is a positive number if you try to set something else you will only get the default value: 1000 and an error message in the console.

## Lib.width(*arg*)

The width function will be used to define wide the chart is allowed to be. Since the Chart is responsive it will resize itself to the biggest possible version.

At the end your input will be a number in a String with the ending px. So, if your input is 90 the style will be set to “90px”. It doesn’t matter if you give the parameter as a number or a string.

Lib. width(90); || Lib. width(“90”); || Lib. width(“90px”);

## Lib.height(arg)

The height function will be used to define wide the chart is allowed to be. Since the Chart is responsive it will resize itself to the biggest possible version.

At the end your input will be a number in a String with the ending px. So, if your input is 90 the style will be set to “90px”. It doesn’t matter if you give the parameter as a number or a string.

Lib. height(90); || Lib. height(“90”); || Lib. height(“90px”);

## Lib.type(*arg*)

With the type function, you decide what Chart you want to use. At the moment, you can choose between:

|  |  |
| --- | --- |
| Nr. | Chart name |
| 1 | Piechart |
| 2 | Donutchart |
| 3 | HorizontalBarchart |
| 4 | VerticalBarchart |

Lib. type(Piechart);

|  |
| --- |
| **Note: It is important that you write it in one word.** |

# If something doesn’t work

First you should consider the console. If something went wrong, every function will give you an error message there and tell you what went wrong.

# If you want to include a new Chart into the Library

The steps for that are fairly simple. First of course you must get the Source Code and install it with Node in the project. After that follow these steps.

1. Save your new File in the Chart folder
   1. Make sure your New Chart class extends the MainChart
      1. Sample code: export default class PieChart extends MainChart {
2. Go to the type.config.js File
   1. Include your new Chart at the top of the file
      1. Sample code: import DonutChart from '../charts/pieChart/donut.chart';
   2. Write a new case in the setter method chartType
      1. Sample code: case 'piechart': this.\_chartType = new PieChart(); break;
   3. Test your new Chart
3. Done.

# Literatur und Quellenverzeichnis

Hinweis: An dieser Stelle muss ein Literatur- und Quellenverzeichnis eingefügt werden.

(es kann auch mit der Fussnote ein Hinweis auf die Quelle gemacht werden, diese muss aber im Quellen VZ ersichtlich sein

Method chaining Internet Quelle:

Name des Autors (falls erkennbar), „Titel der Seite“, Webadresse, Datum des letzten Zugriffs

Beispiel Internet

Quellenangabe „Koala“, http://de.wikipedia.org/wiki/Koala, 22.03.2008

Buch Quelle:

Erklärung Name des Autors: Titel. Verlag, Jahr, Seite, auf der der zitierte Text steht.

Beispiel Buch

Quellenangabe Ernst Walter Bauer: Humanbiologie. Cornelsen, 2006, S. 50.

# Glossar

|  |  |
| --- | --- |
| Begriff | Bedeutung |
| Achsenskalierung | Die Grössen Veränderung einer Achse eines Diagrammes (Beispiel Barchart) |
| Barchart | Eine Art von Diagramm (im deutschen ein Balkendiagramm) |
| Boilerplate (Code) | Sich immer wiederholender Codeteile. |
| custom controls | benutzerdefinierte Steuerelement |
| Donutchart | Eine Art von Diagramm (im deutschen ein Ringdiagramm) |
| ECMA-Script | Standardisierte Form von Java Script |
| Highchart-Charts | Highcharts ist ein Produckt der Firma Highsoft. Highcharts sind interaktive Java Script Diagramme |
| Java Script | Eine Programmiersprache die ursprünglich 1995 von Netscape für dynamisches HTML in Webbrowsern entwickelt wurde[[19]](#footnote-19) |
| Library | Bezeichnet in der Programmierung eine Sammlung von Unterprogrammen/-Routinen, die Lösungswege für thematisch zusammengehörende Problemstellungen anbieten.[[20]](#footnote-20) |
| Piechart | Eine Art von Diagramm (im deutschen ein Kreisdiagramm) |
| Pkorg | PkOrg unterstützt die verantwortlichen Prüfungsorganisationen bei der Durchführung der IPA (Individuelle Produktiv-Arbeit).[[21]](#footnote-21) |
| rendern | Software-Rendering bezeichnet die Methode der Grafikberechnung ohne spezialisierte Hardware, d. h. nur durch die CPU ohne Unterstützung durch eine Grafikkarte oder Vergleichbares.[[22]](#footnote-22) |
| stacked Barchart | Eine Art von Diagramm (im deutschen ein gestapeltes Balkendiagramm) |
| Template | Eine Art Vorlage die von einem anderen Programmierer übernimmt und mit seinen eigenen Änderungen anpasst. In diesem Projekt wäre das ein bereits konfiguriertes Projekt das jedoch noch keinerlei Inhalt hatte. |
| Unit Tests | Ein Modul Test auch als Komponententest bekannt. Welcher jede einzelne Funktion Testet. |

Tabelle 22: Glossar

1. Leitfaden\_Hermes5\_1.pdf [↑](#footnote-ref-1)
2. Ausgeschrieben «Source Code» (deutsch Quelltext oder Quellcode) [↑](#footnote-ref-2)
3. Leitfaden\_Hermes5\_1.pdf(26.04.2017) [↑](#footnote-ref-3)
4. https://de.wikipedia.org/wiki/SMART\_(Projektmanagement)(26.04.2017) [↑](#footnote-ref-4)
5. https://www.fin.be.ch [↑](#footnote-ref-5)
6. http://textart4u.blogspot.ch/search?q=shell [↑](#footnote-ref-6)
7. https://www.computerwoche.de/a/tipps-zur-qualitaetssicherung,1867513,4(30.05.2017 [↑](#footnote-ref-7)
8. https://www.ecma-international.org/ecma-262/6.0/ [↑](#footnote-ref-8)
9. https://babeljs.io/(05.05.2017) [↑](#footnote-ref-9)
10. http://eslint.org/docs/about/(05.05.2017) [↑](#footnote-ref-10)
11. https://de.wikipedia.org/wiki/Lint\_(Programmierwerkzeug)(05.05.2017) [↑](#footnote-ref-11)
12. http://eslint.org/docs/about/(05.05.2017) [↑](#footnote-ref-12)
13. http://mherman.org/blog/2015/09/10/testing-node-js-with-mocha-and-chai/#.WRAoN9ryguU [↑](#footnote-ref-13)
14. http://chaijs.com/ [↑](#footnote-ref-14)
15. https://mantro.net/en/blog/what-is-webpack [↑](#footnote-ref-15)
16. https://de.wikipedia.org/wiki/Programmierschnittstelle(08.05.2017) [↑](#footnote-ref-16)
17. https://github.com/krasimir/webpack-library-starter(08.05.2017) [↑](#footnote-ref-17)
18. http://krasimirtsonev.com/blog/article/javascript-library-starter-using-webpack-es6(08.05.2017) [↑](#footnote-ref-18)
19. https://www.google.ch/search?q=donutchart+deutsch&rlz=1C1CHBF\_deCH736CH736&oq=donutchart+deutsch&aqs=chrome..69i57.4222j0j4&sourceid=chrome&ie=UTF-8#q=Java+Script(09.05.2017) [↑](#footnote-ref-19)
20. https://de.wikipedia.org/wiki/Programmbibliothek(09.05.2017) [↑](#footnote-ref-20)
21. https://www.pkorg.ch/informatik/(09.05.2017) [↑](#footnote-ref-21)
22. https://de.wikipedia.org/wiki/Software\_Rendering(09.05.2017) [↑](#footnote-ref-22)