

Dokumentation

für die DHBW Software Systems gGmbH

Entwicklung eines SWE-CASE-TOOLs

|  |  |
| --- | --- |
| Analyse Team: | David Eckel, Florian Schiffel (9073402), Stefan Maier (7514857) |
| Kunde: | DHBW Software Systems gGmbH |
| Projektleiter Kunde: | Bohl, Peter, Dipl-Ing. |
| Datum der Abgabe: | 17.06.2020 |

Inhalt

[1. Einleitung 3](#_Toc42508024)

[2. Installations-/ Debugging-Hinweise 3](#_Toc42508025)

[3. Programmstruktur 3](#_Toc42508027)

[3.1. Struktur des Programms 3](#_Toc42508028)

[3.2. Struktur der Projektdaten 3](#_Toc42508029)

[4. Entwurfsmuster 3](#_Toc42508033)

[4.1. Interfaces 3](#_Toc42508034)

[4.2. Singleton 3](#_Toc42508035)

[5. Programmtests 3](#_Toc42508036)

[5.1. Import/Export 3](#_Toc42508037)

[5.2. Selbstoptimierung 3](#_Toc42508038)

[6. Ergänzungen 3](#_Toc42508039)

# Einleitung

Aufgaben Beschreibung/ einleitende Worte (max. 1 Seite)

# Installations-/ Debugging-Hinweise

was ist bei der Installation zu beachten, welche IDE, Maven/XML aktivieren etc. (inkl. Versionen)

Visual Paradigm (wie installieren, wo Diagramme gespeichert etc.)



# Programmstruktur

Um die leider nach wie vor oft zum Einsatz kommende, für die professionelle Durchführung eines Software-Projektes jedoch unzureichende Code and Fix Vorgehensweise direkt von Anfang an zu unterbinden, wurden bereits zu Beginn des Projekts die Anforderungen genauestens analysiert und die Planung im Team besprochen. Kurz auf V-Modell bzw. Wasserfallmodell eingehen? Hierbei wurde schnell klar, dass sich die gestellten Anforderungen an Datenhaltung, Ein- und Ausgabe sowie Steuerung des zu erstellenden Programms am besten durch den Einsatz des Model-View-Controller Prinzips realisieren lassen. Zusätzlich wurden bei der Analyse der Anforderungen mehrere fast vollständig unabhängige Teilbereiche des komplexen Endprodukts erkannt. So ist die Erfassung von Produktfunktionen zwar sehr ähnlich zur Erfassung der Produktdaten, bei der Umsetzung der Produktfunktionen bewährte Methoden können somit bei der Umsetzung der Produktdaten angewendet werden und erfordern einen geringeren Aufwand in der Entwicklung. Eine wirkliche Zusammenarbeit der beiden Module ist jedoch nicht gegeben, die einzigen Schnittstellen sind die Datenhaltung im Modell sowie die Abfrage zur Aufwandsabschätzung. Auch sind die drei Module der Daten, Funktionen sowie Abschätzung logisch gesehen komplett unabhängig von den allgemeinen Produktinformationen wie Zielbestimmung, Produkteinsatz sowie Umgebung. Zusätzlich zur Aufgliederung in Model, View und Controller konnte eine vertikale Dekomposition in vier Bereiche erreicht worden. In allen drei Bereichen der MVC-Musters konnte somit eine Aufteilung in unabhängige Modelteile, Controller und Views gewährleistet werden.

Package Diagramm

Grober Aufbau

# Struktur des Programms

Klassendiagramme mit einbeziehen und Verweis auf Visual Paradigm

# Struktur der Projektdaten



# Entwurfsmuster

# Interfaces

# Singleton

# Programmtests

jeweils Kontrollflussgraphen, wie vorgegangen wurde und Erklärung wie Test ausgeführt werden kann

# Import/Export

# Selbstoptimierung

# Ergänzungen

sonstige Informationen