**Projekt: VirtualCalendar**

Projektplan

[Dokumentstruktur basiert auf RUP „Software Development Plan“]

# Dokumentinformationen

## Änderungsgeschichte

|  |  |  |  |
| --- | --- | --- | --- |
| *Datum* | *Version* | *Änderungsgrund / Kurzfassung der Änderung* | *Autor* |
| 24.03.2017 | 1 | Erstellung | Kuczera |
| 28.03.2017 | 2 | Bearbeitung | Kuczera |
| 30.03.2017 | 3 | Nach Beratung | Artinger |
|  |  |  |  |

## Inhalt

[1. Dokumentinformationen 2](#_Toc478667911)

[1.1. Änderungsgeschichte 2](#_Toc478667912)

[1.2. Inhalt 2](#_Toc478667913)

[2. Einführung (Introduction) 3](#_Toc478667914)

[2.1. Referenzen (References) 3](#_Toc478667915)

[3. Projekt Übersicht (Project Overview) 3](#_Toc478667916)

[3.1. Zweck und Ziel (Purpose and Objectives) 3](#_Toc478667917)

[3.2. Annahmen und Einschränkungen (Assumptions and Constraints) 3](#_Toc478667918)

[3.3. Arbeitsergebnisse (Deliverables) 4](#_Toc478667919)

[4. Projektorganisation (Project Organization) 4](#_Toc478667920)

[4.1. Organisationsstruktur (Organizational Structure) 4](#_Toc478667921)

[4.2. Externe Schnittstellen (external Interfaces) 4](#_Toc478667922)

[5. Management Abläufe (Management Processes) 4](#_Toc478667923)

[5.1. Projekt Kostenvoranschlag (Project Estimates) 4](#_Toc478667924)

[5.2. Projektplan (Project Plan) 4](#_Toc478667925)

[5.2.1. Iterationsplanung / Meilensteine (Iteration Objectives / Milestones) 4](#_Toc478667926)

[5.2.2. Besprechungen (Meetings) 5](#_Toc478667927)

[5.2.3. Abgabe (Releases) 5](#_Toc478667928)

[6. Risiko Management (Risk Management) 5](#_Toc478667929)

[7. Arbeitspakete (Work Package) 5](#_Toc478667930)

[8. Infrastruktur (Infrastructure) 7](#_Toc478667931)

[9. Unterstützende Prozesse (Supporting Process Plans) 7](#_Toc478667932)

# Einführung (Introduction)

## Referenzen (References)

* <https://mva.microsoft.com/en-US/training-courses/windows-10-development-for-absolute-beginners-14541?l=rXZlwKxqB_2505632527>
* <https://mva.microsoft.com/en-US/training-courses/c-fundamentals-for-absolute-beginners-16169?l=BQvowJQIC_306218949>
* <https://mva.microsoft.com/en-US/training-courses/windows-10-getting-started-with-uwp-14431?l=rTA5oGEsB_3005632527>
* <http://www.c-sharpcorner.com/uploadfile/gtomar/difference-between-cpp-and-C-Sharp/>
* <http://stackoverflow.com/questions/692225/what-are-the-differences-between-c-c-sharp-and-c-in-terms-of-real-world-appli>
* <https://www.regensburger-katalog.de/InfoGuideClient.ubrsis/singleHit.do?methodToCall=showHit&curPos=10&identifier=-1_FT_916175504&tab=showAvailabilityActive>

# Projekt Übersicht (Project Overview)

Mit Hilfe der VirtualCalendar App können Termine/Verpflichtungen/Aufgaben eingetragen, verschoben oder gelöscht werden.

Der Benutzer kann:

* Erinnerungen und Alarme für die verschiedenen Verpflichtungen setzen.
* Zwischen Einzelansicht oder Gesamtansicht auswählen.
* Man kann zwischen Tag, Woche, Monat, Jahr auswählen.
* Die eigene Zeitverfügbarkeit einsetzen und die von den anderen Benutzern nachschlagen.
* Meldungen über Terminverschiebung automatisch bekommen.
* Einladungen für Besprechungen/Besichtigungstermine u.a automatisch nach Eintragung bekommen.
* Die Anwendung kann auf einem Windows 10-Phone oder Windows 10-Desktop ausgeführt werden.
* Eine Erweiterung des Tools für Android wäre möglich.

## Zweck und Ziel (Purpose and Objectives)

Wir möchten mit diesem Projekt Familien, WG-Bewohnern und anderen Leuten helfen die ihre Termine miteinander Koordinieren müssen, indem sie ihre Termine durch den VirtualCalendar besser aufeinander abstimmen können. Das Ziel unseres Projektes ist, einen gemeinsamen Kalender zu entwickeln, der Termine der Personen Gruppen besser koordinieren lässt.

## Annahmen und Einschränkungen (Assumptions and Constraints)

Wir benötigen eine Datenbank, mit der wir die Daten der Nutzer speichern und aufrufen müssen. Diese ist essentiell für unser Programm, daher stellt ihre Nicht-Funktionalität auch das größte Risiko dar. Unser Programm würde ohne funktionierende Datenbank seinen Zweck nicht erfüllen, da in diesem Fall nur eine Einzelperson den Kalender verwenden könnte und die Daten auf dem lokalen Rechner gespeichert würden.

Außerdem benötigen wir eine Verknüpfung zu einem E-Mail-Server, um Einladungen zu verschicken und Erinnerungen zu empfangen. Bei Nicht-Funktionalität können keine Gruppen mehr erstellt werden und man bekommt keine Erinnerungen mehr.

## Arbeitsergebnisse (Deliverables)

* Projektplan
* Code Listing
* Software Installationspaket
* Bedienungsanleitung
* Dokumentation

# Projektorganisation (Project Organization)

Zuerst werden die Oberflächen gestaltet:

1. Startansicht
2. Tagesansicht
3. Wochenansicht
4. Monatsansicht
5. Einzelansicht
6. Gruppenansicht
7. Fenster.

Danach wird die Datenbank aufgebaut und mit unserem Programm verbunden.

Im Anschluss wird die Verknüpfung zum E-Mail-Server hergestellt.

Als nächster Schritt werden die Funktionalitäten implementiert.

Test und Fehlerkorrektur werden regelmäßig bei jedem Arbeitspaket durchgeführt.

## Organisationsstruktur (Organizational Structure)

* Andra Draghiciu (Qualitätsmanagement)
* Cristian Sanchez (Konfigurationsmanagement)
* Annalena Artinger (Fehlermanagement)
* Karin Günnewig (Qualitätsmanagement)
* Kathrin Kuczera (Admin)

## Externe Schnittstellen (external Interfaces)

* Email Server
* Datenbank System

# Management Abläufe (Management Processes)

## Projekt Kostenvoranschlag (Project Estimates)

Es stehen 16 Wochen zur Verfügung, wobei alle Wochen vollkommen gebraucht werden. Es werden jeweils 9 Stunden pro Woche aufgebracht.

## Projektplan (Project Plan)

Siehe dazu „VC\_Zeitpool2.xlsx“.

### Iterationsplanung / Meilensteine (Iteration Objectives / Milestones)

* Meilenstein 1 (Projektplan 04.04.2017): Es liegt ein ausgearbeiteter Projektplan vor, welcher unsere Aufgaben und Ziele umfasst.
* Meilenstein 2 (Requirements 25.04.2017): Die Start-, Tages-, Wochen- und Monatsansicht des Kalenders sind designt. Klassen sind grob definiert. UML-Diagramme sind erstellt.
* Meilenstein 3 (Architektur / End of Elaboration 16.05.2017): Stakeholder werden befragt und deren Anforderungen werden berücksichtigt. Klassen sind alle definiert. Einzel- und Gesamtansicht sind mit den Kalenderansichten verbunden. Datenbank ist aufgebaut. Funktionen sind zum Teil implementiert.
* Meilenstein 4 (Design Review 07.06.2017): Die Datenbank ist eingebunden und interagiert mit dem E-Mail-Server. Kalenderfunktionen sind vollständig implementiert.
* Meilenstein 5 (Präsentation 05.07.2017): Unser Projekt ist vollständig entwickelt und ausführbar. Dieses hat eine ansprechende Oberfläche und ist leicht zu bedienen.

### Besprechungen (Meetings)

Unsere Meetings finden regelmäßig jeden Dienstag um vorrausichtlich 13:30Uhr und am Freitag um 11:45Uhr an der OTH-Regensburg statt. Der Grund dafür ist unsere Ergebnisse, sowie unsere nächsten Schritte bei der Software Erstellung zu besprechen. Daraufhin werden Aufgaben gerecht untereinander verteilt.

### Abgabe (Releases)

|  |  |  |
| --- | --- | --- |
| Typ | Datum | Inhalt |
| Prototyp 1 | 25.04.2017 | Kalenderansichten |
| Prototyp 2 | 16.05.2017 | Oberfläche und einzelne Funktionen |
| Betaversion | 07.06.2017 | Funktionsfähiger Testkalender |
| Finalversion | 05.07.2017 | Voll funktionsfähiger Kalender |

# Risiko Management (Risk Management)

|  |  |
| --- | --- |
| Risiko | Gegenmaßnahme |
| Datenbank nicht komplett funktionsfähig |  |
| E-Mail-Server interagiert nicht mit unserem Programm |  |
| Kompatibilität |  |

# Arbeitspakete (Work Package)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Arbeitspaket | Inhalt | Verantwortliche/r | Zeit | Abhängigkeit |
| Projektplan |  | alle | 1Woche |  |
| UML-Diagramme | UseCase-Diagramm  Aktivitätsdiagramm | alle | 4h |  |
| Klassen | Klassen definieren | Sanchez | 9h |  |
| Startseite | Oberfläche mit Button-Dummies  (Einzelansicht, Gesamtansicht, Termin erstellen, Gruppe erstellen, Profil bearbeiten, Aufgabenliste, Exit) designen  Textblock-Dummy  Test | Sanchez | 3h |  |
| Einzelansicht | Oberflächengestaltung  Auswahl von Tag, Monat, Woche, Jahr  Termine einblenden  Test |  |  | Kalenderansichten |
| Gruppenansicht | Oberflächengestaltung  Auswahl von Tag, Monat, Woche, Jahr  Termine einblenden  Personen Farbe anzeigen  Test |  |  | Kalenderansichten |
| Tagesansicht | Oberflächengestaltung  Termine mit Zeit einblenden  Test | Günnewig | 9h | Datenbank |
| Wochenansicht | Oberflächengestaltung  Tage und Termine [wie Stundenplan]  Test | Kuczera | 9h | Datenbank |
| Monatsansicht | Oberflächengestaltung  Tage und Termine einblenden  Test | Draghiciu | 9h | Datenbank |
| Jahresansicht | Oberflächengestaltung  Jahr auswählen Button  Monate anzeigen  Test | Artinger | 9h |  |
| Datenbanken | Datenbank finden  Datenbank einbinden  Namen und Passwort eingeben  Test | Sanchez | 6h |  |
| Fenster | Oberflächengestaltung  Profil anlegen  Gruppen anlegen und Benutzer einladen (Admin)  Gruppe suchen (Benutzer)  Termine anlegen  Test |  |  | Datenbank  Datenbank  Datenbank  Datenbank |
| Funktionalitäten | Termin erstellen, löschen, bearbeiten  Gruppe erstellen  Termine markieren  Benachrichtigungen schicken  Test |  |  | Datenbank  Datenbank  E-Mail-Server |
| Test | Betaversion  Finalversion | alle |  |  |
| Fehlerkorrektur |  | Artinger |  |  |
| Dokumente | Sitzungsprotokolle  Zeiterfassung  Programmdokumentation  Bedienungsanleitung | Individuell, je nach Absprache |  |  |
| Konfigurationsmanagement |  | Sanchez |  |  |
| Persönliche Weiterbildung | Informationsbeschaffung, Recherche | alle |  |  |
| Logogestaltung |  | Draghiciu | 2h |  |
| Qualitätsmanagement |  | Günnewig, Draghiciu |  |  |
| Review-Vorbereitung | Power-Point Präsentation vorbereiten und Vortrag einstudieren |  |  |  |

# Infrastruktur (Infrastructure)

* Visual Studio Enterprise 2015
* Subversion
* Bugzilla
* UMLET
* Rechner mit Windows 10
* TurtoiseSVN
* GIMP
* Photoshop (wenn nötig)
* Datenbank, E-Mail (noch nicht ausgewählt)

# Unterstützende Prozesse (Supporting Process Plans)

Konfigurationsmanagement-Plan:

Wir arbeiten mit dem Konfigurationsmanagement-Programm Subversion, in dem wir unsere Codeteile verwalten und zusammenführen. Herr Sanchez ist dafür verantwortlich und verwaltet die Zusammenführung der einzelnen Teile.

Wir speichern jeden Entwicklungsstand einzeln ab, wodurch jeder wiederherstellbar ist. Alles wird in die Dropbox hochgeladen und dort abgespeichert.

Fehlerbehandlungs-Plan:

Zur Fehlerbehandlung verwenden wir das Programm Bugzilla, welches von Frau Artinger verwaltet wird. Jeder Fehler wird sofort dokumentiert und bewertet, welche Auswirkungen dieser Fehler mit sich bringt und in welchem Zeitraum dieser behoben werden soll.

Nach Fehlerbehebung wird die Korrektur zusätzlich dokumentiert und der Fehlerstatus aktualisiert.

Qualitätsmanagement-Plan:

Der geschriebene Code wird ausreichend kommentiert, sodass er für andere verständlich und nachvollziehbar ist. Frau Günnewig überwacht diese Dokumentation auf Vollständigkeit und Verständlichkeit.

Zuerst werden die einzelnen Komponenten getestet auf Funktionalität. Sobald alle Einzelteile fehlerfrei laufen, werden diese zusammengeführt und sichergestellt, dass sie in ihrer Zusammenarbeit die gestellten Anforderungen erfüllen.

Frau Draghiciu stellt sicher, dass jede Sitzung protokolliert wird und prüft diese Protokolle auf Vollständigkeit. Außerdem ist sie für die Zeiterfassung der Gruppe zuständig.