

Meeting Protokolle

Andreas Zweili

16. März 2019

Meeting vom 23.12.2018

Protokoll der 1. Besprechung

Treffpunkt: Bhf Olten

Datum: 23.12.2018, 14:00 Uhr - 15:30 Uhr

Informationen:

Vorstellungsrunde

Marco

Beruflich Bereits viel gemacht mit ca. 40 selbständig gemacht.

Schulisch Kam nach und nach und nimmt mittlerweile einen Grossteil seiner Zeit in Anspruch.

Familie Hat Frau und Kinder.

Hobbies/Interessen Südamerika (lebte 2 Jahre dort) und Salsa?

Andreas

Beruflich Quereinsteiger, gelernter Automatiker, Faszination Informatik, Deployment, Orchestrierung, Automatisierung

Schulisch API Spezialisierung mehr angesprochen aufgrund der Dozenten während IBZ Ausbildung.

Hobbies IBZ, Musik, Konzerte

Termine besprochen

Abgabe der DA Informatik erfolgt vereinfacht, Widerspruch zur offiziellen Vorgabe. Email mit den Unterlagen am 18.3. an Marco Frei reicht. Man muss nichts beim Sekretariat abgeben.

Nur 1 Diplomordner z.Hd. Kandidat mit separat Selbständigkeitserklärung unterschrieben. Ist an der Präsentation mitzunehmen. Die Erklärung wird dann gleich von C.Herren kontrolliert.

Beobachtung von Marco Frei:

Exakte Zeitplanung, auch Miteinbezug der eigenen Intensität (geplante Stunden Aufwand) pro Woche, Arbeitspakete verständlich, PSP soweit detailliert.

Einige Kapitel des DA Antrags können 1:1 in die Doku oder allenfalls mit ergänzenden Resultaten aus der Analyse übernommen werden.

Feinziele entsprechen bereits den ausgearbeiteten Anforderungen aus der bereits durchgeführten Projektanalyse.

Feedback Andreas:

Mussziele müssen erreicht werden.

Aufbau der Arbeit

Initialisierung / Voranalyse

- Einleitung
- Umfeld, Motivation
- Projekthandbuch (Projektdokumentation) mit Ausgangslage, Problembeschreibung, Projektziele, Abgrenzungen, Zeitplanung, Projektplanung, Projektmetho-

dik Wasserfallmethodik (Begründung)

- Projektrisiken (betriebliches Umfeld) der Ist Umgebung
- Projektcontrolling Soll-Ist (Zeit, Ressourcen, Kosten (eigene Lohnkosten annehmen)), Bewertung
- allenfalls Zeitplan mit Phasen der Wasserfallmethodik anpassen, Milestone setzen

Analyse

- Ist - Analyse, Stakeholderanalyse, Umweltanalyse, Marktanalyse
- SWOT (Stärken, Schwächen)
- quasi hypothetische Risikoanalyse der Ist Umgebung, um daraus Bewertung der eigenen Lösung auch mit einer Risikoanalyse miteinzubeziehen, evtl. durch fiktiver User, Use Case
- Soll - Analyse
- Use Case mit Aktoren festlegen
- Anforderungskatalog kann aus den Zielen hergeleitet werden, kann evtl. noch erweitert oder angepasst werden, z.B. Präzision, Umwelt, Verwendung von Testarten etc.

Konzept

- Big Picture aufzeigen, Zusammenspiel erläutern
- Start der Variantendiskussion
- Entscheid, Begründung des Lösungswegs
- Detailkonzept mit techn. Plan
- Diagramme. UML, Klassendiagramm, etc.
- Testkonzept. Testarten (Unit, Integrations-Testing), Festlegung der Testarten, Rahmenbed.

Realisierung

- Open-Source Projekt als gesamtes Werk, leistet im Rahmen der DA wichtige Vorarbeit
- mit minimalen Funktionen (welche?) -> Backup, Restore, mount, delete, Archiv
- Beweis der Funktion? wie? -> Testumgebung aufbauen, erläutern
- Linux System Python Skript -> self extracting binary wird interpretieren

Ausblick:

- kritische Würdigung, lessons learned
- Verwendung im kommerziellen Umfeld
- Bewertung der Risiken anhand der eigenen Lösung

Nächste Termine

Meeting #2: 26.01.2019 14:00

Meeting #3: Ende Feb. exakter Termin noch ausstehend.

Pendent

- roter Faden der Gesamtstruktur anhand Analyse und Synthese abgleichen
- Grobstruktur trennen auf Analyse und Konzept, z.B. vor dem Start der Variantendiskussion
- zusätzlich Grundidee, Vision in der Einleitung
- fiktive Situation im Rahmen der Analyse -> Ist Risikoanalyse zum Leben bringen
- Überlegung, ob alle Ziele 1:1 den Anforderungen genügen, welche als Voraussetzung fürs Testing dienen

- möglicher Zugriff auf Datenablage

Doing

Andreas erstellt folgende Lieferobjekte:

- Besprechungsprotokoll mit Entscheidungen
- Arbeitsjournal, wochengenau (einfaches Logbuch, z.B. Excel oder in Zeitplan integriert, mit Bezug zur erstellten Projektplanung) -> Gitlog führen plus erweitern (Planung, Ergebnis, Eindruck)
- Projektplan mit Phasen, Milestones, Arbeitspaketen
- Zeitplan mit Intensität als Tabelle oder Gantt (250h inkl. mögliche Pufferzeit 50h), Einbau in Bericht
- Festlegung der Systemgrenze, evtl. mit UML Werkzeugen (Bubble Charts)
- messbare Muss- und Kannziele
- Traktandenliste für nächstes Meeting erstellen
- Risikoanalyse machen
- Festlegung der Grobstruktur, in Bezug zum Projektantrag

Noch unvollständig

Nichts.

Persönlicher Eindruck von Marco Frei:

- präzise Analyse
- klare Ziele, Vorstellung
- ausgezeichnet im Zeitplan

- detaillierte Ergebnisse
- aktuelle Ausgangssituation könnte präziser formuliert werden
- Engineering Vorgehen verstärkt hervorheben

Meeting vom 26.01.2019

Protokoll der 2. Besprechung

Treffpunkt: IBZ Aarau

Datum: 26.1.2019, 14:00 Uhr - 15:15 Uhr

Allgemeines

Beobachtung: exakte Zeitplanung, auch Miteinbezug der eigenen Intensität (geplante Stunden Aufwand) pro Woche, Arbeitspakete verständlich, Projektstrukturierung soweit detailliert -> erledigt

ausgezeichnet im Zeitplan

open source Projekt als gesamtes Werk, aktuelle Source Code Version auf github unter <https://github.com/borg-qt/> -> erledigt

einige Kapitel des DA Antrags können 1:1 in die Doku oder allenfalls mit ergänzenden Resultaten aus der Analyse übernommen werden -> erledigt

Beobachtung: Feinziele entsprechen bereits den ausgearbeiteten Anforderungen aus der bereits durchgeführten Projektanalyse -> erledigt

roter Faden der Gesamtstruktur anhand Analyse und Synthese abgleichen -> erledigt

Grobstruktur trennen auf Analyse und Konzept, z.B. vor dem Start der Variantendiskussion -> erledigt

zusätzlich Grundidee, Vision in der Einleitung -> erledigt

fiktive Situation im Rahmen der Analyse -> Ist Risikoanalyse zum Leben bringen -> Erweiterungen mit use cases abgebildet - erledigt

Überlegung, ob alle Ziele 1:1 den Anforderungen genügen, welche als Voraussetzung fürs Testing dienen > erledigt

möglicher Zugriff auf Datenablage -> erledigt

<https://github.com/Nebucatnetzer/thesis>

Besprechungsprotokoll mit Entscheidungen führen -> erledigt

Zeitplan mit Intensität als Tabelle oder Gantt (250h inkl. mögliche Pufferzeit 50h),
Einbau in Bericht -> erledigt

detailliertere Informationen zu gewissen Punkten der Konzeptarbeit, d.h. Technik Borg-Backup erläutern, QT Framework erläutern, Logging zeigen, Backup erstellen, Netzwerk prüfen, Datenduplizierung erklären, Konfiguration in Plain Textdatei erstellen, Cross Plattform Techniken erläutern (insbesondere da BorgBackup Support nur unter Linux, somit nicht vollständig möglich), JSON Format erläutern, Zusammenspiel und Schnittstelle zwischen Python Skript und bestehendem Binary beschreiben, Verwendung der Umgebungsvariablen für Programmierung, Passwort Handling beim Aufruf des Binary
-> auf gutem Wege

geplante 1 Woche Reserve in der Phase Realisierung, während dem Codieren -> doing

Arbeitsjournal, wochengenau (einfaches Logbuch, z.B. Excel oder in Zeitplan integriert, mit Bezug zur erstellten Projektplanung) -> Gitlog führen plus erweitern (Planung, Ergebnis, Eindruck) -> doing

Qualitätskontrolle und Controlling ansprechen (wichtiger Bestandteil der Bewertung)
-> doing

Aufbau der Arbeit

Initialisierung / Voranalyse:

Einleitung -> erledigt

Projekthandbuch (Projektdokumentation) mit Ausgangslage, Problembeschreibung, Projektziele, Abgrenzungen, Zeitplanung, Projektplanung, Projektmethodik Wasserfallmethodik (Begründung) -> erledigt

Projektrisiken (betriebliches Umfeld) der Ist Umgebung -> erledigt

allenfalls Zeitplan mit Phasen der Wasserfallmethodik anpassen -> erledigt

Milestone setzen -> erledigt

Umfeld, Motivation -> nochmals nachfragen, insbesondere Programmierung mit Python und Nutzung des QT Framework für das GUI ist Neuland, Bedeutung hervorheben

nachträgliche Berücksichtigung weiterer möglicher Kannziele -> doing

Projektcontrolling Soll-Ist (Zeit, Ressourcen, Kosten), Bewertung -> doing

Analyse:

SWAT (Stärken, Schwächen) -> erledigt

quasi hypothetische Risikoanalyse der Ist Umgebung, um daraus Bewertung der eigenen Lösung auch mit einer Risikoanalyse miteinzubeziehen -> recht projektbezogene Darstellung -> erledigt

Anforderungskatalog kann aus den Zielen hergeleitet werden, kann evtl. noch erweitert oder angepasst werden, z.B. Präzision, Umwelt, Verwendung von Testarten etc. -> erledigt

Soll - Analyse -> erledigt

use case Diagramm mit Aktoren festgelegt -> erledigt

Ist - Analyse, Stakeholderanalyse, Umweltanalyse, Marktanalyse -> alle Usergruppen festgelegt -> erledigt

allenfalls Wahl bestimmter Usergruppen noch begründen, z.B. für das Testing -> doing

geplante Durchführung des Testings -> Transparenz, mit Auswertung der Ergebnisse -> doing

Festlegung der Systemgrenze, evtl. mit UML Werkzeugen -> nochmals prüfen

Konzept:

Aktivitätsdiagramm gemacht -> erledigt

Kontextdiagramm gemacht -> erledigt

Testkonzept. Testarten (Unit, Integrations-Testing), Festlegung der Testarten, Rahmenbed. -> 25 Testfälle definiert, Testfälle aus use cases abgeleitet -> erledigt

big picture aufzeigen, Zusammenspiel erläutern -> auf gutem Wege, kann mit einem zusätzlichen Klassendiagramm vervollständigt werden

Technologien nachvollziehbar beschreiben, insbesondere Zusammenspiel der Komponenten (JSON, QT, Python, binary, Umgebungsvariablen -> nochmals nachfragen

Variantendiskussion -> nochmals prüfen

Entscheid, Begründung des Lösungswegs -> noch verfeinern, Bedeutung open source Projekt dabei hervorheben, doing

Detaillkonzept mit techn. Plan -> doing, insbesondere die Zuordnung, Abstimmung der Klassen, Funktionen in den entsprechenden SW Modulen

SW Design mit einem «Klassendiagramm» über verwendete Module, Klassen, Schnittstellen, Input Output am Schluss der Realisierung fertig darstellen, verwendete Module beschreiben, wird im Nachhinein abgebildet -> doing

Konsistente Darstellung von Testfällen und deren Durchführung, auch in Bezug zu den Muss- und effektiv realisierten Kannzielen -> doing

Realisierung:

Realisierung mit Linux System Python Skript und QT Framework -> erledigt

Konfiguration aus dem GUI wird als Plain Text neu erstellt, Passwort als Plaintext gespeichert, wird jedoch beim Aufruf dem Binary nicht mitgegeben, da dies in einer Umgebungsvariable definiert wird -> erledigt

Bau und Erläuterung der Testumgebung -> doing

Entwicklung basiert auf Python, mit eigen erstellten Realisierungselementen, welche nicht einem Standardaufbau entsprechen -> diese kurz erläutern, Codeausschnitte einbauen und mit referenzierter Quelle ergänzen -> doing

Implementation der DB als Kannziel, Logging als Kannziel, direkte Integration in DB wäre wünschenswert -> nochmals nachfragen

Fehlerbehandlungen erfolgen nur an einer Stelle, entweder direkt über vorhandenes Backup Binary oder neu über das erstellte GUI -> doing

Berechtigungsprobleme (z.B. bei include-Verzeichnissen) werden nicht speziell behandelt, Verarbeitung wird übersprungen -> doing

allg. gültige Umgebungsvariablen verwenden -> doing

Netzwerk Prüfung erfolgt via Port Check auf Ziel IP, wo Repository drauf läuft -> doing

geplanter Abschluss der Programmierung 2. März -> doing

Ausblick:

kritische Würdigung, lessons learned

Weiterverwendung als openSource Projekt für angedachte Weiterentwicklungen

Bewertung der Risiken anhand der eigenen Lösung -> Rückmeldung gegeben

Projektabschluss mit Überprüfung der Ziele, anhand des DA Antrags resp. Pflichtenheftes

Nächstes Treffen

nächstes Treffen am 3.3, 14 Uhr in Olten

noch unvollständig: nichts

Meeting vom 02.03.2019

Protokoll der 3. Besprechung

Treffpunkt: IBZ Aarau

Datum: 26.1.2019, 14:00 Uhr - 15:00 Uhr

Ablauf

Kurze Einleitung mit Smalltalk. Erkundigung nach Stand der Arbeit und Besprechung der noch offenen Punkte. Sind soweit alle erfüllt. Analyse könnte um Funktionen von Borg erweitert werden. -> Roter Faden fehlt noch ein bisschen.

Ansonsten soweit gut auf Kurs noch finalisieren und dann die digitale Abgabe wie beim ersten Meeting besprochen.