

SDDC

Software Defined Data Center

Projektplan

Silvan Adrian
Fabian Binna

1 Änderungshistorie

Datum	Version	Änderung	Autor
17.09.15	1.00	Erstellung des Dokuments	Gruppe
18.09.15	1.01	Organisationsstruktur + Einführung + Projektübersicht	Silvan Adrian

Inhaltsverzeichnis

1	Änderungshistorie	2
2	Einführung	5
2.1	Zweck	5
2.2	Gültigkeitsbereich	5
2.3	Referenzen	5
3	Projektübersicht	5
3.1	Zweck und Ziel	5
3.2	Primäre Features	5
3.3	Erweiterte Features	5
3.4	Lieferumfang	5
3.5	Annahmen und Einschränkungen	5
4	Projektorganisation	5
4.1	Organisationsstruktur	6
4.2	Externe Schnittstellen	6
5	Managment Abläufe	6
5.1	Kostenvoranschlag	6
5.2	Sprints	6
5.3	Besprechungen	6
6	Risikomanagement	7
6.1	Risiken	7
6.2	Umgang mit Risiken	7
7	Arbeitspakete	7
8	Infrastruktur	7
8.1	Entwicklungsinfrastruktur	7
8.2	Tools/Software	7
8.3	Kommunikationsmittel	7
9	Qualitätsmassnahmen	8
9.1	Dokumentation	8
9.2	Projektmanagement	8
9.3	Entwicklung	8
9.3.1	Unit Testing / Test-Driven Development	8
9.3.2	Code Review	8
9.3.3	Code Style Guidelines	8

9.4	Testen	8
9.4.1	Komponententest	8
9.5	Systemtest	8

2 Einführung

2.1 Zweck

Dieses Dokument beschreibt die Planung der SA SDDC.

2.2 Gültigkeitsbereich

Dieses Dokument ist während des ganzen Projekts gültig und wird laufend aktualisiert.

2.3 Referenzen

-

3 Projektübersicht

Es soll eine generische API oder bestehende API verwendet werden, um über ein Dashboard direkt Services zu abonnieren, dabei wird für jede Virtuelle Maschine Compute, Storage und Network benötigt.

3.1 Zweck und Ziel

Gelerntes anwenden und sich mit der gestellten Aufgabenstellung auseinandersetzen und dabei neues lernen.

3.2 Primäre Features

3.3 Erweiterte Features

3.4 Lieferumfang

3.5 Annahmen und Einschränkungen

4 Projektorganisation

Wir setzen beim Project auf Scrum, wodurch lediglich die Projektrollen Product Owner, Entwickler und Scrum-Master vorhanden sind. Es gibt keine Unterscheidung zwischen speziellen Fachgebieten und das Team als ganzes ist gefragt.

4.1 Organisationsstruktur



4.2 Externe Schnittstellen

Projektbetreuer:

Beat Stettler (Beat.Stettler@ins.hsr.ch)

Urs Baumann (Urs.Baumann@ins.hsr.ch)

5 Managment Abläufe

5.1 Kostenvoranschlag

Gemäss Vorgaben wird mit 240 Stunden gesamt gerechnet + die einberechneten Risiken von 41 Stunden.

Gesamtaufwand: 281 Stunden

Risikoaufwand: 41 Stunden

Aufwand pro Woche: ca.: 20 Stunden

Aufwand pro Woche pro Teammitglied: ca.: 10 Stunden

5.2 Sprints

5.3 Besprechungen

Daily Meetings (ca. 10 - 15min) werden während der Pause oder über Skype stattfinden, um sich gegenseitig über Probleme, abgeschlossene Arbeitspakete etc. zu informieren. Nach jedem Abschluss eines Sprints wird ein Sprintreview geplant und durchgeführt, um

zu prüfen ob alles gemäss Plan erledigt wurde + die aktuellste Version der Software released.

6 Risikomanagement

6.1 Risiken

6.2 Umgang mit Risiken

7 Arbeitspakete

Die Arbeitspakete werden in Open Project erstellt und gepflegt. Lesender Zugriff ist anonym möglich, schreibender nur eingeloggt (Projekt ist öffentlich).

Link zur Open Project Instanz : <http://sddc.silvn.com>

8 Infrastruktur

8.1 Entwicklungsinfrastruktur

Name	Hardware	Betriebssystem	IDE
Silvan Adrian	MacBook Pro	OSX 10.10.5	N/A
Fabian Binna	Lenovo T430s	Windows 10	N/A

8.2 Tools/Software

- **BuildServer:** Travis-CI
- **Konfigurationsmanagement:** GIT
- **Notifications:** Slack

8.3 Kommunikationsmittel

- E-Mail
- Skype
- Open Project (Kommentare)
- GitHub (Issues,Kommentare)
- Whatsapp
- Slack

9 Qualitätsmassnahmen

9.1 Dokumentation

Die Dokumentation befindet sich auf einem privaten GitHub Repository. Die Texte werden in LaTeX geschrieben. Die Dokumente werden versioniert.

https://github.com/silvanadrian/SDDC_Doku.git

9.2 Projektmanagement

Für das Projektmanagement wird OpenProject verwendet.

sddc.silvn.com

9.3 Entwicklung

9.3.1 Unit Testing / Test-Driven Development

Die Unit Tests kommen in einen separaten Ordner "Test". Es wird eine möglichst hohe Code Coverage angestrebt. Die Code Coverage wird mit einem Tool (z.B. eclEmma) sichergestellt.

Die Klassen werden mit Hilfe von Test-Driven Development implementiert.

9.3.2 Code Review

Nach jedem Sprint oder bei Abschluss grosser Arbeitspakete wird ein Code Review durchgeführt.

Review Protokoll

9.3.3 Code Style Guidelines

Editor Standard

9.4 Testen

9.4.1 Komponententest

9.5 Systemtest