

Projektplan

Distributed Systems Engineering 2013

Wartelistenverwaltung planbarer Operationen

Gruppe 16

Taylor Peer (0725922), David Stöckl (1128480), Michael Wagner (0827376)

09.04.2013

Projektplanung

Es soll eine Online-Wartelistenverwaltung zur zeitlichen Planung von Operationen umgesetzt werden.

Ziele

- funktionierende Online-Wartelistenverwaltung
- elastische Skalierbarkeit des Systems (auf Lastspitzen reagieren)
- flexibles verteiltes System, Trennung nach Komponenten
- Vier verschiedene Sichten: Öffentlich, Patient, Arzt, Krankenhaus
- Drei grundlegende Komponenten: User-Interface, Allocator, Messenger
- Sicheres System durch ausreichendes Testen der Komponenten

Nicht-Ziele

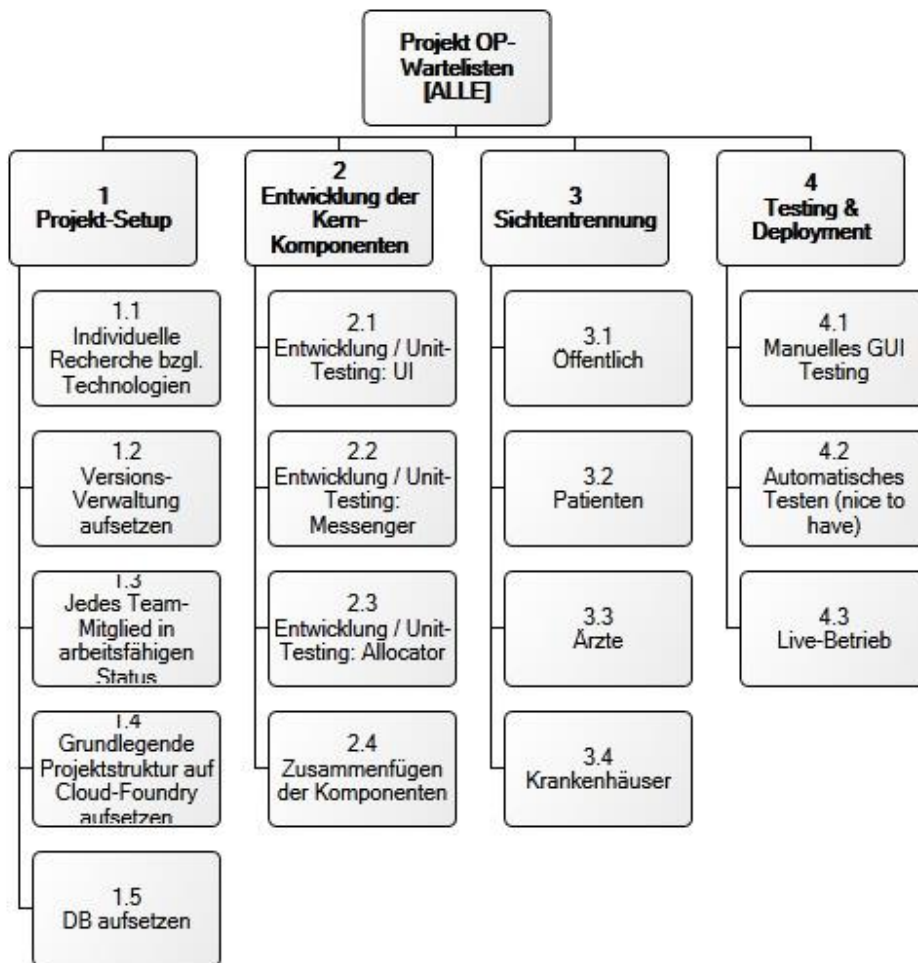
- scheitern an bestimmten Technologien
- laienhafte Umsetzung durch unzureichende Einarbeitung in die für das Team neuen Technologien
- fehlerhafte Umsetzung

Rollenvergabe

Aus den Vorkenntnissen wurde folgende Einteilung der Rollen im Projekt abgeleitet:

Matr. Nr.	Name	Rolle	E-Mail
1128480	David Stöckl	Teamleiter	david.stoeckl@blackbam.at
0827376	Michael Wagner	Technischer Architekt	e0827376@student.tuwien.ac.at
0725922	Taylor Peer	Release-Manager / Tester	e0725922@student.tuwien.ac.at

Arbeitspakete



Aufwandsschätzung

Projekt-Setup	12 Stunden / Mitglied für die Recherche der neuen Technologien 4 Stunden / Mitglied für das Setup
Entwicklung der Kern-Komponenten	15-25 Stunden pro Mitglied für je eine Komponente + gegenseitiges Testen 5 Stunden pro Mitglied für das Zusammenfügen
Sichtentrennung	10 Stunden / Mitglied für je 1,3 Sichten und gegenseitige Kontrolle
Testing / Deployment	5-10 Stunden / Mitglied
Gesamt / Mitglied	51 - 66 Stunden / Mitglied
Gesamt	ca. 176 Stunden

Zeitliche Planung (Ressourcen Planung / Milestones)

Datum	Nr.	Milestone	Beschreibung
17.4.2013	1	Projektsetup abgeschlossen (AP 1)	<ul style="list-style-type: none"> - alle Mitglieder haben sich individuell über die unbekannten Technologien informiert - Versionierung funktioniert - grundlegende Projektstruktur inkl. MongoDB lauffähig
21.5.2013	2	Entwicklung der Kern-Komponenten abgeschlossen	UI, Messenger, Allocator stellen die gewünschten Funktionalitäten zur Verfügung und können integriert werden
28.5.2013	3	Das grundlegende System ist fertig und funktioniert	- Komponenten integriert
4.6.2013	4	Sichten-Integration	<ul style="list-style-type: none"> - Sichten integriert - Security implementiert
15.6.2013	5	Release	<ul style="list-style-type: none"> - Tests abgeschlossen - Live-Betrieb wird aufgenommen

Individuelle Arbeitseinteilung des Teams (Iceberg-Liste)

Milestone	Tätigkeit	Teammitglied
1	Recherche	Alle
1	Lokale Arbeitsumgebung aufsetzen	Alle
1	Versionsverwaltung Setup	Taylor Peer
1	Setup Cloud-Foundry	David Stöckl
1	Setup Kontrolle	Michael Wagner
2	Implementierung UI-Komponente	Michael Wagner
2	Implementierung Messenger-Komponente	David Stöckl
2	Implementierung Allocator-Komponente	Taylor Peer
2	Komponenten zusammenfügen	Alle

3	Sichten-Integration: Jeweils eine Sicht pro Mitglied	Alle
4	Testleitung	Taylor Peer
4	Testen	Alle
4	Deployment	Taylor Peer

Technische Planung

Als Vorgehensmethode für das Projekt hat sich das Team einstimmig für das klassische **Wasserfallmodell** entschieden (strikt sequentieller Projektablauf).

Anforderungen > Spezifikation > Planung > Entwurf / Design > Implementierung > Integration > Deployment

Begründung: Es ist kaum zu erwarten, dass die Anforderungen sich wesentlich verändern und die kleine Projektgröße lässt andere Vorgehensweisen unnötig aufwändig erscheinen.

Die einzelnen Phasen werden vom Projektleiter in Übereinstimmung mit dem ganzen Team freigegeben, bevor in die jeweils nächste Phase übergegangen wird. Der Schwerpunkt soll auf der Implementierungsphase liegen.

Außerdem soll die Implementierungsphase des Projekts **teilweise testgetrieben** stattfinden, insofern dass die einzelnen Komponenten (Arbeitspaket 2) ausreichend durch Unit-Tests getestet werden sollen, bevor diese ins Gesamt-System integriert werden.

Technologien / Tools

Die folgenden Technologien / Tools werden verwendet:

Cloud Foundry als Laufzeitumgebung bzw. Server
Spring Framework
Apache Maven
JUnit

Als Entwicklungsumgebung fungiert für alle Teammitglieder die **Spring Tool Suite (STS)**. Diese Laufzeitumgebung ermöglicht es auf einfache Art und Weise, die Tools Apache Maven sowie JUnit-Testing und das Spring Framework zu verwenden, sowie die Kommunikation mit dem CloudFoundry Account.

Für eine gute Trennung der Komponenten ist angedacht, **Spring Roo 1.2.3** zu verwenden. Auch **MongoDB** wird verwendet, aufgrund der integrierten GEO-Suche. Das Team behält sich allerdings vor, PostgreSQL zu verwenden, falls es sich doch als bessere Wahl herausstellen sollte. Des Weiteren wird **RabbitMQ** als Message-oriented Middleware verwendet.

Für die Versionsverwaltung ist **Git** geplant.

Keine weiteren Technologien / Tools sollen durch dieses Dokument festgelegt werden.