Diary Team Red

barta3
hammd1
paras1
reubd1
stola3

Table of Contents

Woche 1

CS1 Task 1 / 26.2.2013

Entscheide
- Dokumentsprache: Deutsch
Output Task 1:
~/doc/task01
- Task01.docx
- fragen.xml
Vorgehen:
-
Tasks "Target users" und "Key Features" wurden im
Plenum
besprochen, damit alle die gleiche Basis haben.
- Restliche
Tasks individuell
Tasks inkl. Verantwortung
[x] Task 1 Grundlagen barta3 / hammd1
[x] Präsentation 1.3.13
[x] Slides vorbereiten hammd1
[x] Vortrag / Präsentation hammd1

CS1 Task 2 / 26.2.2013

Output Task 2: ~/doc/task02
- Task02_SE_Entscheid.docx
Tasks inkl. Verantwortung
[x] Software Engineering Entscheide paras1 /
[x] Präsentation 1.3.13
[x] Slides vorbereiten paras1
[x] Vortrag / Präsentation solta3

reubd1 / stola3

Woche 3

CS2 Task 1 / 5.3.2013

Output Task 1:
~/doc/cs2_tasks
- CS2_T1_task1.docx
CS2 Task 2 / 7.3.2013
Output Task 2:
~/doc/cs2_tasks
- CS2_T2_Anwendungsfallbeschreibung.docx

Woche 4

CS1 Task 3 / 12-15.3.2013

Vorgehen: Teamarbeit gemäss Design Thinking Prozess
Outputs Task 2:
~/doc/cs1_tasks/task3_design_thinking/
- Scopes
- Features
- Fragen / Interview
- User Stories

Offene Tasks bis Woche 5
[x] Präsentation (paras1)
[x] Präsentation Inputs all
[x] Feature Tagebuch (reubd1)
[x] Feature Wanrsystem (hammd1)

Woche 5

Diary wurde nicht gut gepflegt. Die Rückmeldung hat dies bestätigt. Fazit: DocBook war zu umständlich.
Entscheid: Google Drive mit Exports ins git.

CS 1 Task 4

19.-22.3.2013 (Team komplett)
Requirements

- Use Case Diagram
- Grundgerüst des Dokument "R#CS1-T4-REQ" erstellt
- Aufgabenaufteilung
    - barta3 Architektur / Sytstem Model
    - hammd1 Anwendungsfallbeschreibung #5 Erinnerung darstellen
    - paras1 System Requirements
    - reubd1 Anwendungsfallbeschreibung #4 Medikament verwalten
    - stola3 User Requirements / Testing

19.-22.3.2013
Inviduell an zugeordnetem Task gearbeitet

22.3.2013 (Team komplett)
Dokument visualiseren
Diskussion zum Diary
Präsentation der Ergebnisse

Woche 6

CS 1 Task 5

2.4.2013 (Team komplett)
Review für Team Green als ganzes Team
Requirements Review erhalten von Team White. Ergebisse unter
/doc/cs1_tasks/task05_requirements_review

CS 1 Task 6

2.4.2013 (Team komplett)
Context Model der Applikation erstellt
Activity Model des Use Cases #4 (Medikament verwalten) erstellt

Exercise 3

5.4.2013 (Team komplett)
UML State Diagram → arbeit als Team auf Papier

Woche 7

CS 1 Task 7

12.4.2013(Team komplett)
Entscheide: Das Domänenmodell wird auf die wesentlichen Klassen reduziert und diese werden dafür vollständig aufgeführt.
Tasks:

- Review des Requirementdokuments
- CRC Cards für die Hauptklassen erstellt
- Erster Entwurf des Domänenmodells
- Verantwortungen und Abhängigkeiten definiert

16.4.2013(Team komplett)
Tasks:

- Review des Domänenmodells
- CRC Cards erweitert
- Verfeinerung des Domänenmodells
- Erstellung des Sequenzdiagrammes
- Review des Sequenzdiagrames im Team
- Vorbereitung der Präsentation

19.4.2013

Tasks:

- Fertigstellung der Präsentation zu Task 7
  - CRC Cards “Profile” und “Medication”
  - Domain Model
  - Sequence Diagrams “System Clock” und “Medication”
- Präsentation im Team
  Abgelegt unter doc/cs1_tasks/task07_uml_class_sequence/P_Task07.pdf

Woche 9

CS 1 Task 8

23.4.2013(Team komplett)

Entscheide: Einsatz des MVC Pattern und Aufbau der Architektur gemäss Layer Pattern

Tasks:

- Domainmodel mit MVC Klassen ergänzt
- Packages definiert
- Präsentation erstellt

Woche 10

Vorgehen: Teamarbeit

Entscheide: Task direkt im Journal dokumentieren

CS 1 Task 9

For your mobile application, redesign your Architecture taking into account

1. the Vaadin framework
   Domainmodel aus Task 8 sollte umsetzbar sein
2. the need for persistent data storage
   → Wir haben uns für eine DB entschieden, mit folgenden Varianten als persistence layer:
   a. Variante Hibernate
      https://vaadin.com/wiki/-/wiki/Main/Using%20Hibernate%20with%20Vaadin
      Vorteile: einfacher Austausch der DB (z.B. HSQLDB oder Derby)
      Nachteil: Braucht einen Application Server wie z.B. Glassfish oder JBoss
   b. Variante Vaadin JPAContainer
      https://vaadin.com/book/vaadin7/-/page/jpacontainer.html
      Vorteile: Gute Integration in Vaadin, simple
   c. Variante Spring Roo
      https://vaadin.com/springroo
3. and architectural design patterns in general

Woche 11 / 12

Vorgehen: Teamarbeit zur Aufteilung der verschiedenen Tasks

Sprint Backlog:	hammd1, reubid1
Product Backlog:	reubid1, hammd1
Project (Vaadin/Tomcat):	paras1, stola3
State Pattern:	barta3
Visitor Pattern:	barta3

Project setup

→ paras1 & stola3

Da im Team noch keine Vaadinkenntnise vorhanden waren, wurde ein Referenzprojekt gesucht. Diese Suche wurde unterschätzt.
Schlussendlich wurde jedoch ein super Projekt mit der Basis von Vaadin JPAConteiner und Hibernate gefunden.

Nach dem Überarbeiten des pom.xml läuft das Referenz Projekt sauber und wir werden können von der sauberen Vaadin Dokumentation profitieren. (siehe GIT Sub-Maven-Eclipse-Project jpacontainer-addressbook-demo)