

Statusbericht I

(Mashup-Plattform)

1. Gesamtstatus

Das Team konkretisierte in der ersten Woche nach dem Kick-Off-Meeting vom 19.08.2011 die Projektidee mit Hilfe von Brainstorming und Online-Diskussionen. Bei einem persönlichen Treffen wurden aus den eingegangenen Vorschlägen ein gemeinsames Ziel abgeleitet.

Gesamtziel

Das Endprodukt dieser Projektarbeit ist eine Software, welche dem User das Erstellen und Ausführen eigener Mashups^{*} ermöglicht. Jeder Mashup bildet eine Datenquelle, die genutzt werden kann, um daraus neue Mashups zu kreieren. Auf diese Weise können User kollaborativ Daten aus dem Internet filtern und strukturieren.

Nach der Fixierung der Projektidee wurden die Rollen für den weiteren Projektverlauf verteilt.

Rollenverteilung

Alexander Matt (Projektmitarbeiter)
Joël Poyet (Auftraggeber)
Jan Wild (Projektmitarbeiter)
Malte Wintner (Projektleiter)

Der Fokus in der zweiten Woche lag bei der Einführung eines Informations- und Berichtswesens. Da die Zusammenarbeit zum grössten Teil nur über das Internet erfolgen konnte, wurden geeignete Web-Programme (Skype, WIKI, Bug Tracker, ...) auf dem lokalen Rechner, beziehungsweise auf einem gehosteten Webserver installiert. Nach Verabschiedung des Projektauftrages definierte das Team die Teilziele und Anwendungsfälle. Als Teil der Machbarkeitsstudie wurde der Beschluss gefasst, einen Prototypen zu entwickeln. In den darauf folgenden Wochen wurde der Prototyp entworfen und einzelne Basis-Klassen implementiert.

1.1 Bewertung

Das Projekt verlief bis an hin planmässig. Besonders in den wöchentlich durchgeführten Online-Treffen wurde ersichtlich, dass das Team begeistert bei der Projektarbeit war und viele Ideen einbrachte. Beim Entwurf des Prototypen wurde das Konzept der „Virtualisierung“ verfolgt, um die Heterogenität der Web-Datenquellen hinter einer Abstraktionsschicht zu verbergen. Dieses Konzept legte die Basis für die weitere Entwicklung.

* Unter Mashup wird bei dieser Projektarbeit das Verknüpfen von Web-Inhalten unterschiedlicher Datenquellen verstanden.

2. Termine

Bezeichnung	Status	Deadline
Projektidee fixieren	erledigt	26.08.11
Kommunikation etablieren	erledigt	02.09.11
Anwendungsfälle aufzeichnen	erledigt	05.09.11
Anforderungen und Teilziele festlegen	erledigt	09.09.11
Prototyp entwerfen	erledigt	16.09.11
Prototyp implementieren	offen	30.09.11
Prototyp analysieren	offen	07.10.11
Kernsystem (1. Inkrement) entwerfen	offen	21.10.11
Kernsystem implementieren	offen	28.10.11
Kernsystem testen	offen	04.11.11
Gesamtsystem (2. Inkrement) entwerfen	offen	18.11.11
Gesamtsystem implementieren	offen	02.12.11
Gesamtsystem testen und Bugs beheben	offen	16.12.11

2.1 Bewertung

Status: planmässig

Besonderen Wert wurde auf einen klaren und vollständigen Entwurf des Prototypen gelegt, um die Implementierungszeit zu verkürzen und die Qualität zu sichern.

3. Lieferobjekte

Bezeichnung	Status	Deadline
Projektauftrag	erledigt	02.09.11
grober Projektstrukturplan (wird mit der Zeit detaillierter)	(erledigt)	02.09.11
Systemidee	erledigt	02.09.11
Systemanwendungsfälle	erledigt	05.09.11
Lösungsansatz	erledigt	09.09.11
Softwarearchitektur (Prototyp)	erledigt	16.09.11
Klassenmodelle (Prototyp)	erledigt	16.09.11
Coderichtlinien	erledigt	16.09.11
1. Statusbericht (vorliegendes Dokument)	erledigt	20.09.11
Softwarearchitektur (1. Inkrement)	offen	14.10.11
Klassenmodelle (1. Inkrement)	offen	21.10.11
2. Statusbericht	offen	15.11.11
Softwarearchitektur (2. Inkrement)	offen	15.11.11
Klassenmodelle (2. Inkrement)	offen	18.11.11
Domänenmodell	offen	01.12.11
Systemablaufmodelle	offen	01.12.11
Schnittstellenbeschreibung	offen	01.12.11
Datenmodell	offen	01.12.11
Dynamische Modelle	offen	01.12.11
Testkonzept	offen	01.12.11
GUI-Spezifikationen	offen	01.12.11

3.1 Bewertung

Status: planmässig

Für die Software-Entwürfe wurden UML-Klassendiagramme als bevorzugtes Mittel eingesetzt.

4. Qualität

Status: planmässig

Das Know-how des Teams wurde gebündelt, um Design-Fehler frühzeitig zu erkennen. Durch strenge Richtlinien beim Entwurf des Klassendesign konnte die Qualität stetig verbessert werden.

5. andere Probleme

Das Synchronisieren und Comitten des Quellcodes in das Git-Repository bereitete anfangs Probleme. Aber diese konnten mit etwas Übung behoben werden.

6. nächste Schritte

Der nächste Schritt ist die Fertigstellung des Prototypen.

Dieser Meilenstein ist sehr wichtig, da er nicht nur Aufschluss über die technische Machbarkeit, sondern auch die Stärken und Schwächen bei der Teamarbeit aufzeigen wird.

Allfällige Schwachpunkte können dann rechtzeitig korrigiert werden, um das Endprodukt erfolgreich zu implementieren.

7. aktualisierte Rollenorganisation

Alexander Matt (?)

Joël Poyet (?)

Jan Wild (?)

Malte Wintner (?)

8. Vorgehensmodell

Das Team entschied sich für eine inkrementelle Entwicklung, um die Komplexität zu reduzieren. Geplant wurden zwei Inkremente. wobei für das Kernsystem (1. Inkrement) zuerst ein Prototyp entworfen wurde.

Die Qualität des Prototypen wird voraussichtlich schon sehr nahe an das endgültige Kernsystem heran kommen.

Die anstehende Programmierung wird von gegenseitigem Code Review begleitet, um allfällige Fehler frühzeitig zu erkennen und eine einheitliche Codierung zu erreichen.