**Twitteranalyser**

**Dokumentation Container**

Dokumentation Container

PR Software Engineering II  
LV-Leiter: Dipl. Ing. Dr. Heinz Pozewaunig  
SS 2014

1. Einleitung

Im Rahmen der Lehrveranstaltung ‚Software Engineering II‘ wurde eine Twitteranalyse entwickelt, wobei als Softwareentwicklungsmodell *Scrum* Anwendung fand. Das Projekt, an dem acht Personen gearbeitet haben, wurde in drei Monaten, vom 18. März bis zum 18. Juni 2014 realisiert. Genaue Angaben zu den Kosten für das Produkt finden sich in Kapitel 5.  
  
Wie beim Kundengespräch am 18.03.2014 von Ihnen erwähnt, sollte es möglich sein, dass ein Mitarbeiter in der Marketingabteilung ein Stimmungsbild einer bestimmten Veranstaltung durch die Sammlung von ausgewählten Tweets abrufen kann. Der Mitarbeiter sollte im Programm eine Veranstaltung anlegen, eine Veranstaltung auswählen, löschen, ändern oder speichern können. Es soll ein Zeitraum definiert werden können, bis wann die Tweets ausgewählt werden sollen. Die so erhaltene Sammlung soll analysiert, auf einer Landkarte dann die positiven und negativen Stimmungen angezeigt werden können und über die firmeninterne Analyseseite abrufbar sein.

Diese Anforderungen wurden alle in das Projekt aufgenommen, in fünf Sprints aufgeteilt und schließlich erfolgreich erledigt.

.

2. Projektorganisation

Das Scrum-Entwicklungsteam besteht aus neun Mitgliedern, deren Aufgabenbereiche in folgender Tabelle ersichtlich werden.

|  |  |  |
| --- | --- | --- |
| Name | Stärken/Schwächen | Aufgaben |
| Kesselbacher Max |  | Techn.Dokumentation,  Entwicklung, Tests |
| Kammel Johannes |  | Entwicklung, Design |
| Krumpholz Maximilian |  | Dokumentation, Unit Tests |
| Mertens Henrik |  | Unit Tests |
| Moser Manfred |  | Entwicklung, Design |
| Mösslacher Corinna |  | Entwicklung,  techn. Dokumentation |
| Ressmann Andreas |  | Projektmanagement,  Entwicklung |
| Steinkellner Rosemarie |  | Dokumentation, Kostenschätzung |

Tabelle : Mitglieder und ihre Aufgaben

4. Fragenkatalog vom 17.3.2014

Der folgende Fragenkatalog stellt für das Kundengespräch am 19. März 2014 eine Basis dar. Es wird jedoch auch eine freie, informelle Diskussion geben.

Als Redner fungiert Alexander Ressmann:

1. Anwendung der Software?
2. Was ist der Zweck der Software?
3. Wo soll sie eingesetzt werden?
   1. Eine oder mehrere Veranstaltungen?
   2. Filterfunktion?
      1. Zeitraum Filter
      2. Nach Veranstaltung
4. Gibt es ein Beispiel zu dem was er sich vorstellen?
5. Was sind relevante Informationen (laut Ausschreibung)?
6. Was versteht der Kunde unter einem Trend
   1. Schlüsselwörter?
7. Wo bewegt sich das Budget?
8. Aufbereitung der Daten?
   1. Coporate Identity?
   2. Farben?
9. Verfügbarkeit der Daten?
   1. Export?
   2. Reporting?
   3. Jederzeit abrufbar? Zeitrahmen?
10. User Interface?
    1. Aufbau?
    2. Funktionalität?
11. Wer wird die Software verwenden?
    1. Userverwaltung
    2. Rechte Vergabe
12. Web Interface?
13. Mobile Apps?
14. Nachbetreuung? Wartung? Evolution?

5. ProtokollE

Thema: Erstes Kundengespräch  
Datum: 19. März 2014  
Ort: Raum Büro Dr. Pinzger, Universität Klagenfurt

**Protokoll zum 18.3.2014**

Von Max Kr :

Usecase :

* Veranstaltung anlegen/auswählen
* Zeitraum definieren
* Laden der zur Veranstaltung gehörigen Tweets
* Filtern und editieren der gefundenen Hashtags
* Keywords für positive und negative Stimmung definieren/hinzufügen
* Speichern der Veranstaltung
* Aufbereitung der Daten
* Export der Daten

Nice to have:

* Stimmungskeywords gewichten
* Not Hashtags
* Veranstaltungstyp incl passender Keywords
* Keywords für positiv und negativ selbst definierbar

**Corinna und Max Ke haben die Anforderungen in User Storys übertragen:**

User Story 1 – Anlegen einer Veranstaltung

Anna aus der Marketing-Abteilung möchte das Stimmungsbild des GrandPrix Spielberg zum ersten Mal vorab ausloten.  
Dazu öffnet sie den Firefox-Browser und ruft die firmen-interne Analyseseite auf.  
Es gibt noch keine angelegten Veranstaltung in der Veranstaltungsübersicht.  
Sie legt also eine neue Veranstaltung an, indem sie auf „Neue Veranstaltung hinzufügen“ klickt. Es erscheint das Eingabeformular.  
Dort muss sie folgende Daten über die Veranstaltung (Veranstaltungs-Parameter) eingeben:

* Angezeigter Name der Veranstaltung (muss nicht eindeutig sein)
* Textuelle Beschreibung der Veranstaltung (Zweck der Analyse, interne Infos, …)
* Liste der zu suchenden Tags (vielleicht mehrere Queries zu einer Veranstaltung)
* Zeitraum der Sammlung der zu analysierenden Tweets
* Minimalanzahl der Tweets zur Benachrichtigung (als Alternative und Ergänzung zum Zeitraum)

Nachdem sie alle relevanten Daten eingegeben hat, klickt sie auf „Sammlung starten“. Da die Sammlung erst startet und noch keine weitere Bearbeitung möglich ist, wird die Veranstaltungsübersicht mit der neuen Veranstaltung angezeigt. Der Status der Veranstaltung wird als „Sammlung aktiv“ angezeigt. Dieser Status ändert sich auf „Sammlung beendet“, sobald die Endkriterien erreicht wurden.

Weitere Informationen / Fragen

Was sind die Endkriterien?  
Reicht die Statusänderung als vom Kunden gewünschte Benachrichtigung?

User Story 2 – Detailansicht einer Veranstaltung

Anna möchte den Zwischenstand ihrer angelegten Veranstaltung begutachten. Dazu klickt sie in der Veranstaltungsübersicht auf die Zeile mit der gewünschten Veranstaltung. Es öffnet sich die Detailansicht der Veranstaltung. In der Detailansicht wird folgendes angezeigt:

* Veranstaltungsparameter (diese können bearbeitet werden)
* Status der Veranstaltung (Sammlung, Analyse, …)
* Details zum Status (Fortschritt der Sammlung)

Anna sieht den Status „Sammlung aktiv“, deshalb können noch keine weiteren Aktionen ausgeführt werden. Sie gelangt über den „Zurück“-Button zurück zur Veranstaltungsübersicht.

Weitere Informationen / Fragen

Aus dem Status ergeben sich die unterschiedlichen Aktionen, welche in dieser Ansicht gestartet werden können

User Story 3 – Status: „Sammlung beendet“

Anna sieht, dass der Status ihrer Veranstaltung in der Veranstaltungsübersicht auf „Sammlung beendet“ gesetzt ist. Deshalb wechselt sie in die Detailansicht der Veranstaltung, um weitere Aktionen zu starten. Folgende Aktionen sind möglich:

1. Analyse
2. Bearbeiten der Tweets
3. Export

User Story 3a - Analyse

Analyse: Analysewörter + Bewertung, Erweiterung der Liste

Anzeige/Darstellung

User Story 3b – Bearbeiten der Tweets

Anna möchte die Liste der gesammelten Tweets nach bestimmten Wörtern filtern. Sie klickt in der Detailansicht der Veranstaltung auf „Bearbeiten der Tweets“. Dort wählt sie die Ausgangsliste oder einen zuvor gespeicherten Filter. Daraufhin wird die Filteransicht geladen – diese besteht aus folgenden Komponenten:

* Editierbare Textfelder „Mit“ und „Ohne“, welche als Filter auf die Tweet-Liste angewendet werden können. Wenn ein Filter ausgewählt wurde, stehen in diesen Feldern die Inhalte dieses Filters, sonst sind sie leer.
* Button „Filter anwenden“, welcher den Filter auf die Tweet-Liste der Datenbank anwendet. Die Ergebnisse werden danach angezeigt.
* Button „Filter speichern“, um den alten Filter mit dem neuen Filter zu überschreiben.
* Button „Neuen Filter speichern“, um mit den aktuellen Parametern einen neuen Filter zu speichern.
* Button „Für Filter ausblenden“, damit die zuvor markierten Tweets für diesen Filter nicht mehr angezeigt werden.
* Button „Für Filter einblenden“, welche die zuvor markierten Tweets, welche schon ausgeblendet waren, wieder einblendet.
* Button „Aus Datenbank löschen“, welcher die zuvor markierten Tweets in der Datenbank auf gelöscht setzt. Dies gilt für alle Filter!
* Eine Checkboxen („Ausgeblendete anzeigen“).
* Anzahl der Suchergebnisse
* Die einzelnen Tweets im Suchergebnis besitzen eine Checkbox, um sie für die Operationen Ausblenden / Einblenden / Löschen auszuwählen.

(SQL-„Viewfilter“ auf die Rohdaten.)

User Story 3c – Export

Anna möchte alle gesammelten, unbearbeiteten Tweets ihrer Veranstaltung exportieren. Dazu klickt sie auf „Daten exportieren“. Daraufhin öffnet sich eine Liste, welche die Tweet-Liste und die gespeicherten Tweet-Filter beinhaltet. Sie wählt einen Eintrag aus, kann den Speicherort der .csv Datei bestimmen und auf „Speichern“ klicken. Danach wird Anna auf die Detailansicht zurück geleitet.

**Erste Anforderungserstellung von Rosemarie und Johannes inkl. Analyse:**

Die Vollständigkeit, Konsistenz, Angemessenheit, Eindeutigkeit, Messbarkeit,

Relevanz, Machbarkeit, Verständlichkeit, gute Strukturiertheit und Modifizierbarkeit

der Anforderungen wurden in einem ersten Testing von Alexander und Rosemarie

durchgeführt. Laufende Überprüfung ist geplant.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Funktionale Requirements | |  |  |  |  |  |
|  |  |  |  |  |  |  |
| frq01: | Veranstaltungen anlegen | |  |  |  |  |
| frq02: | Veranstaltungen bearbeiten | | |  |  |  |
|  | ändern, löschen, aus- und einblenden | | |  |  |  |
| frq03: | Zeitraum definieren | |  |  |  |  |
| frq04: | Keywords anlegen | |  |  |  |  |
|  | Keywords für positive und negative Stimmung definieren/hinzufügen | | | | |  |
| frq05: | Bearbeiten der Tweets | |  |  |  |  |
|  | filtern, speichern, ändern, aus- und einblenden, löschen | | | | |  |
| frq06: | Daten exportieren | |  |  |  |  |
| **Nichtfunktionale bzw. übrige:** | | |  |  |  |  |
| rq01: | Übersicht der Veranstaltungen | | |  |  |  |
| rq02: | Sammeln von tweets | |  |  |  |  |
| rq03: | Positive und negative Stimmung erkennen - Landkarte | | | |  |  |
| rq04: | Soll auf bestehende tweets lernen können | | | |  |  |
| rq05: | Oberfläche soll intuitiv benutzbar und multilingual vorbereitet sein | | | | |  |
| rq06 | Einheitlicher Auftritt – corporate identity | | |  |  |  |

3. Zeiterfassung

Im Weiteren wird der zeitliche Ablauf des Projekts beschrieben. Für jeden Sprint wurde ein Sprintziel definiert, das es zu erreichen galt. Nach dem ersten Sprint sollte bereits eine minimale Version des Twitteranalyser existieren. Dieses Sprintziel wurde nicht ganz erreicht, da ein Kollege überraschend das Team verlassen hatte. Daher konnte das Ergebnis aus Sprint I Veranstaltungen anlegen gemeinsam mit Veranstaltungen anzeigen und bearbeiten erst am Ende von Sprint II erfolreich erreicht werden. Ziel von Sprint III und IV war es, Tweets anzeigen und zu einer Analyse zusammenfassen zu können. Außerdem sollte eine Übersicht der Tweets sowie eine multilinguale Ausgabe der Keywords möglich sein. Alle diese Ziele wurden erfolgreich umgesetzt. Im finalen Sprint V wurde das Ziel bereits angelegte Filter löschen können und die mögliche Darstellung der Analyse in Diagrammen ebenfalls erfolgreich umgesetzt.

Die folgende Tabelle bietet eine Übersicht über den gesamten Projektverlauf. Die Aufgabenbereiche sind dabei den dafür zuständigen Personen zugeteilt. Die grauen Balken zeigen an, über welchen Zeitraum diese Tätigkeiten erledigt wurden.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Aufgabe*** | ***Person*** | ***Sprint I*** |  |  | ***Sprint II*** |  |  | ***Sprint III*** |  |  | ***Sprint IV*** |  |  | ***Sprint V*** |  |  |
| Technische Dokumentation | Mößlacher Kesselb. |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |
| Unittests | Krumpholz |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |
| Mertens |
| Logik | Ressmann |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |
| Kammel |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |
| Design | Moser |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |
| Ressmann |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |
| Sentiment-analyse | Kammel, Moser |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |
| Ressmann |
| Datenbank | Moser |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |
| Ressmann |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |
| Filter-anpassung | Kammel |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |
| Moser |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |
| Ressmann |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |
| Multilingualität | Mößlacher |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |
| Export | Kammel |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |
| Darstellung Analyse | Ressmann |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |
| Moser |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |
| Dokumentation | Steinkellner |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |
| Krumpholz |
| Usability Tests | Kesselb. |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |
| Präsentation | Mertens, Krumpholz, Moser |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |

Tabelle : Projektzeitplan

In der folgenden Tabelle finden sich die genauen Stundenanzahlen, die pro Sprint in das Projekt investiert wurden und die gesamte Stundenanzahl aller Mitarbeiter, die sich auf 678 Stunden beläuft. Die folgende Abbildung zeigt den Zeitaufwand pro Sprint in Prozent.

|  |  |
| --- | --- |
| *Sprint* | *Stunden* |
| Sprint I | 183:00 |
| Sprint II | 165:30 |
| Sprint III | 105:30 |
| Sprint IV | 104:00 |
| Sprint V | 120:00 |
| *Gesamt* | 678:00 |

Tabelle 3: Zeitaufwand pro Sprint

4. Product Backlog

Die nachfolgende Abbildung stellt die ursprüngliche Liste mit Anforderungen, die nach dem Kundengespräch aufgestellt wurde, dar.

|  |  |  |
| --- | --- | --- |
| Beschreibung | User story | Priorität |
|  | points |  |
| Anlegen einer Veranstaltung | **3** | **high** |
| Webanwendung - Detailansicht | 1 | high |
| Parameteränderung | 3 | low |
| Auswertung | 3 | high |
| Filterobjekterstellung |  | high |
| Funktionalität: Löschen | 2 | medium |
| Filterobjektspeicherung |  | high |
| Funktionalität: Ausblenden | 7 | medium |
| Export | **2** | **low** |
| Darstellung | 5 | high |
| Benachrichtigung |  | low |
|  |  |  |

Daraus wurde dann der folgende Product Backlog erstellt, der für die weiteren Sprints zur Abarbeitung bestimmt war. Den jeweiligen Anforderungen sind Aufgabenbereiche zugeteilt. Zudem ist angegeben, in welchem Sprint die Tätigkeit erfüllt wurde.

Der Product Backlog sieht nun wie folgt aus:

|  |  |  |
| --- | --- | --- |
| Sprint | Beschreibung | Priorität |
| 1 | Als <MM> will ich eine Veranstaltung anlegen, | 100 |
|  | sodass das Sammeln der Tweets beginnt |  |
| 2-1 | Als <MM> will ich eine vorhandene Veranstaltung anzeigen | 60 |
| 2-2 | Als <MM> will ich eine vorhandene Veranstaltung bearbeiten | 55 |
| 3-2 | Als <MM> will ich das Tweets nach Anlegen einer Veranstaltung | 99 |
|  | zur Verfügung haben, so dass ich später eine Sentimentanalyse |  |
|  | fahren kann |  |
| 3-4 | Als <MM> will ich die gesammelten Tweets einer Veranstaltung | 55 |
|  | alle anzeigen, sodass ich ein Übersicht bekomme |  |
| 3-1 | Als <MM> will ich eine positiv/negativ Sentiment Analyse der Tweets | 40 |
|  | erhalten |  |
| 4-1 | Als <MM> will ich die vorhandenen Tweets einschränken, | 30 |
|  | sodass nur relevante Tweets analysiert werden |  |
| 5-1 | Als <MM> will ich bereits angelegte Filter auch löschen können. | 30 |
| 3-3 | Als <MM> will ich eigene Keywords in beliebiger Sprache für die | 27 |
|  | positiv / negativ Semtimentanalyse bestimmen können. |  |
| 2-1 | Als <MM> möchte ich eine Mindestanzahl von zu sammelden Tweets | 26 |
|  | pro Veranstaltung angeben, damit ich bewerten kann, |  |
|  | ob das Analyseergebnis sinnvoll ist. |  |
| 4-3 | Als <MM> will ich einzelene Tweets manuell löschen, sodass diese | 20 |
|  | nicht wieder analysiert werden. |  |
| 4-4 | Als <MM> will ich das aktuelle Filterobjekt speichern, sodass ich diese | 15 |
|  | später erneut anwenden kann |  |
| 4-2 | Als <MM> will ich einzelene Tweets manuell für einen Filter | 19 |
|  | ausblenden, sodass diese nicht analysiert werden. |  |
| 5 | Als <MM> will ich zu einer bestimmten Veranstaltung gehörende | 15 |
|  | Daten als .csv exportieren. |  |
| 3-2 | Als <MM> will ich verschiedene Darstellungen der analysierten Daten. | 16 |
| 1-5 | Als <MM> will ich über Tweets informiert werden. | 10 |

Tabelle 4: Product Backlog

7. Sprints

In diesem Kapitel werden die einzelnen Sprints näher beschrieben. Die einzelnen Tabellen enthalten neben den Items aus dem Product Backlog die erledigt werden sollten, die Aufgaben und die Stundenaufstellung der Teammitarbeiter.

7.1. Sprint I

Das Ziel für Sprint I ist es, eine erste Version vom Tweeteranalyser zu erstellen. In der folgenden Tabelle stehen die Items aus dem Product Backlog, die in **Sprint I** erledigt werden sollten. Diese Version soll minimal sein.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
| ID | User Story ID | Sprint Ziel |  |  |  |  |
| T1a | 1 | **Als MM möchte ich eine Veranstaltung anlegen, sodass das Sammeln** | | | | |
|  |  | **der Tweets beginnen kann** | | |  |  |
| T1b | 2\_1 | **Als MM möchte ich eine vorhandene Veranstaltung anzeigen.** | | | | |
| T1c | 2\_2 | **Als MM möchte ich eine vorhandene Veranstaltung bearbeiten.** | | | | |
|  |  |  |  |  |  |  |
|  |  | **ID** |  | **Soll Std** | **Ist Std.** | **Aufgabe** |
|  |  | 1a,1b,1c | Corinna | 4 | 6 | Techn.Dokumentation |
|  |  | 1a,1b,1c | Max Ke | 4 | 6 | Benutzer Dokumentation |
|  |  | 1a,1b,1c | Max Kr | 5 | 7 | Unit Tests |
|  |  | 1a,1b,1c | Rosemarie | 3 | 2 | Dokumentation, Mulitlingualität |
|  |  | 1a,1b,1c | Henrik | 3 | 2 | Multilingualität |
|  |  |  |  |  |  |  |
|  |  | 1a | Andreas | 6 | 7 | Projektsetup ausarbeiten |
|  |  |  |  | 6 | 8 | Basislogik |
|  |  | 1a | Hubert | 7 | 0 | Design Entwurf |
|  |  | 1a | Manfred | 8 | 7 | Datenbank Entwurf |
|  |  |  |  |  |  |  |
|  |  | 1b | Hubert | 8 | 0 | GUI Code,GUI Entwurf |
|  |  |  |  |  |  | Logik, Datenbank |
|  |  |  |  |  |  |  |
|  |  | 1c | Hubert | 6 | 0 | Logo Entwurf, Logo Design |
|  |  | 1c | Manfred | 2 | 3 | in PHP einlesen |
|  |  | 1c | Johannes | 5 | 6 | Sentiment und Twitter |

Tabelle 5: Sprint Backlog für Sprint I

Stundenübersicht Sprint 1:

7.2. Sprint II

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | User Story ID | Sprint Ziel | |  |  |  |
| T1a | 1 | **Als MM möchte ich eine Veranstaltung anlegen, sodass das Sammeln** | | | | |
|  |  | **der Tweets beginnen kann** | | |  |  |
| T1b | 2\_1 | **Als MM möchte ich eine vorhandene Veranstaltung anzeigen.** | | | | |
| T1c | 2\_2 | **Als MM möchte ich eine vorhandene Veranstaltung bearbeiten.** | | | | |
| T2a | 3\_2 | **Als <MM> will ich Tweets nach Anlegen einer Veranstaltung zur** | | | | |
|  |  | **Verfügung haben, sodass ich später eine Sentimentanalyse fahren kann** | | | | |
| T2b | 3\_4 | **Als <MM> will ich alle gesammelten Tweets einer Veranstaltung** | | | | |
|  |  | **anzeigen, sodass ich eine Übersicht bekomme** | | | | |
|  |  |  |  |  |  |  |
|  |  | **ID** |  | **Soll Std** | **Ist Std.** | **Aufgabe** |
|  |  | 1a,1b,1c | Corinna | 2 | 3 | Techn.Dokumentation |
|  |  | 1a,1b,1c | Max Ke | 2 | 3 | Benutzer Dokumentation |
|  |  | 1a,1b,1c | Max Kr | 5 | 7 | Unit Tests |
|  |  | 2a,2b | Max Ke | 4 | 5 | User Test, User Manuel |
|  |  | 2a,2b | Rosemarie | 3 | 2 | Dokumentation, Multilingualität |
|  |  | 2a,2b | Henrik | 3 | 2 | Unit Tests |
|  |  | 2a,2b | Corinna | 4 | 4 | Sequenz Diagramme |
|  |  | 2a,2b | Max Kr | 5 | 7 | Unit Tests für Business Logik |
|  |  | 1a | Andreas | 7 | 8 | Design Entwurf |
|  |  | 1b | Andreas | 8 | 6 | GUI Code,GUI Entwurf |
|  |  | 1b | Johannes | 5 | 6 | GUI Optimierung |
|  |  | 2a | Manfred | 4 | 3 | Datenbank Modellklassen anlegen |
|  |  | 2a | Manfred | 3 | 2 | Datenbank Design anpassen |
|  |  | 2a | Manfred | 3 | 4 | Sammelapp implementieren |
|  |  | 2a | Andreas | 6 | 7 | Projektsetup ausarbeiten |
|  |  | 2a | Max Kr | 6 | 7 | Sched. Task für die SammelApp anlegen |
|  |  | 2b | Manfred | 4 | 5 | Logik, Datenbank |
|  |  | 2b | Johannes | 3 | 4 | GUI für Anzeige der Tweets |
|  |  | 2b | Johannes | 4 | 5 | Business Logik in GUI |

Stundenübersicht Sprint 2:

7.3. Sprint III

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | User Story ID | Sprint Ziel | |  |  |  |
| T3a | 3\_1 | **Als <MM> will ich eine positiv/negativ Sentiment Analyse** | | | | |
|  |  | **der Tweets erhalten** | | |  |  |
| T3b | 4\_1 | **Als <MM> will ich die vorhandenen Tweets einschränken,** | | | | |
|  |  | **sodass nur relevante Tweets analysiert werden** | | | | |
|  |  |  |  |  |  |  |
|  |  | **ID** |  | **Soll Std** | **Ist Std.** | **Aufgabe** |
|  |  | 3a,3b | Corinna | 2 | 3 | Techn.Dokumentation |
|  |  | 3a.3b | Max Ke | 2 | 3 | Benutzer Dokumentation |
|  |  | 3a,3b | Max Kr | 5 | 7 | Unit Tests |
|  |  | 3a,3b | Max Ke | 4 | 5 | User Tests |
|  |  | 3a,3b | Rosemarie | 3 | 2 | Kostenschätzung,Multilingualität |
|  |  | 3a,3b | Henrik | 3 | 2 | Unit Tests |
|  |  | 3a,3b | Andreas | 2 | 4 | Paging, Springer |
|  |  | 3a,3b | Johannes | 5 | 7 | Controller-u.Modelklasse |
|  |  | 3a,3b | Manfred | 5 | 7 | Controller- u. Modelklasse |
|  |  | 3a,3b | Johannes | 3 | 4 | Datenbank |
|  |  | 3a,3b | Manfred | 3 | 4 | Datenbank |

Stundenübersicht Sprint 3:

5. Kosten

Nachfolgend findet sich eine Aufstellung der Kosten für das Projekt (siehe auch die Aufwandsschätzung vom 05.05.2014):

Die Kostenschätzung mittels Function-Point Verfahren ergibt 54 UFP und 47,52 AFP.

Die Kostenschätzung mittels Application-Point Verfahren beträgt bei einem System mit 12 Screens, 4 Berichten mittlerer Komplexität, 6 3GL Modulen und ca. 25 % Wiederverwendung, eine gute Umgebung vorausgesetzt und wenn wir schätzen, dass die Mitarbeiter mittlere Erfahrung haben 104 AP’s und 17 NAP’s = 6 Personenmonate

Der Durchschnitt der beiden Varianten ergibt nun 13 Personenmonate.

Die Gesamtkosten errechnen sich wie folgt:

Stundensatz von 90 Euro x 678 Stunden 61.020,00 €

Nettorechnungsbetrag 61.020,00 €

+ 20 Prozent Umsatzsteuer 12.204,00 €

**Bruttorechnungsbetrag 73.224,00 €**

6. Qualitätsplan

Schon während der Entwicklung des Software-Produkts wurden regelmäßig Unittests durchgeführt, um eine hohe Qualität zu gewährleisten. Zudem wurde das Produkt von Mitarbeitern und externen Personen kontinuierlich auf Benutzerfreundlichkeit getestet. Im Anhang befinden sich Links zu den Ergebnissen dieser Tests. Außerdem ist dort eine Verlinkung zu den Code-Metriken zu finden.

7. Technische Produktbeschreibung

Im Folgenden wird auf die technische Realisierung der Software näher eingegangen. Für nähere Details siehe auch die technische Dokumentation, die sich auf der CD befindet.

Die Software wurde mittels der objektorientierten Programmier- und Skriptsprachen PHP und Java realisiert.

Die Struktur des Projektes ist eine klassische MVC Architektur, realisiert mit Zend Framework (Version 1).

### Architektur

Der Beginn der Anwendung ruft die Index.php Datei auf welche den Zend Bootloader für die primäre Initialisierung des Projekts ladet. Alle Anfragen, die über die Website empfangen werden, werden an die entsprechende Steuerung durch das Zend Framework weitergeleitet. Die Steuerung wird durch die Unterordner der Anwendung bestimmt. Alle Hauptmenüpunkte haben eine entsprechende Steuerung. Der Standard Controller ist der Index Controller.   
Alle Menüunterpunkte sind Aktionen im entsprechenden Controller. Jeder Klick auf einen Unterpunkt ruft eine andere Aktion in der Steuerung auf. Die Index Action wird aufgerufen, wenn der Controller aufgerufen wird. Die Anwendung nutzt Twitter-Bootstrap. Dies ist für die Gestaltung der grafischen Benutzeroberfläche und der Präsentation (z.B. der Datumsauswahl) verantwortlich.

### Controllers

Die Controller Logik ist in fünf Controller aufgeteilt:

Index Controller (Haupt Controller), API Controller, Error Controller, Map Controller und Statistics Controller

### Views

Die phtml-Dateien in "Views" sind für die Darstellung der Web-Seite eingesetzt. Sie enthalten neben den HTML-Code auch PHP-Code für die dynamischen Aspekte der Seite. Alle Nutzer-Interaktionen rufen eine Aktion im Controller hervor, zu der die View gehört. Für die Views "Analyse", "Event" und "Tweets" ist es der API-Controller. Die View "Ereignis" erzeugt eine Liste als Übersicht aller derzeit bestehenden Veranstaltungen. Für die Ansicht der Karte und der Statistik werden Google Maps und Google Visualisation für die Darstellung der Karte und Diagramme verwendet.

### Apimanagement

Die Dialoge sind durch die Datei "apimanagement.js" verwaltet. Alle Dialoge der Web-Anwendung werden unter Verwendung von EJS-Code generiert, der geladen und dynamisch über JQuery angezeigt wird. Das API Management sendet JSON API-Anfragen an den Controller. Der Controller sendet Antworten oder Validierungsfehler.

### Sprachen

Die CSV-Dateien enthalten die Texte, die auf der Webseite angezeigt werden.

Models  
Für alle Datenbanktabellen, die relevant für die Ansichten sind, existiert ein entsprechendes Modell.

8. Risikoanalyse

In der nachfolgenden Tabelle werden mögliche Risiken, die Wahrscheinlichkeit ihres Eintritts und Pläne zur Vermeidung des jeweiligen Risikos aufgelistet. Die Auswertung, der Durchschnitt von Wahrscheinlichkeit und Schweregrad, wird von 2 = geringes Risiko bis 4 = hohes Risiko definiert.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Risiko*** | ***Wahrschein-lichkeit*** | ***Schwere-grad*** | ***Auswertung*** | ***Gegenmaßnahmen*** |
| Personalausfall | Möglich | Moderat | 2 | Konfliktarbeit, Motivation der Mitarbeiter |
| Unrealistische Termin- und Budgetplanung | Wahrscheinlich | Hoch | 4 | Genaue Aufwands- und Kostenschätzung |
| Entwicklung falscher Funktionalitäten | Möglich | Hoch | 3 | Genaue Anforderungsanalyse, Kundengespräche protokollieren, stetiger Kundenkontakt, |
| Entwicklung einer falschen Benutzeroberfläche | Möglich | Hoch | 3 | Kundengespräche protokollieren, stetiger Kundenkontakt, genaue Anforderungsanalyse, Usability-Tests |
| Anforderungen ändern sich kontinuierlich | Möglich | Moderat | 2 | Leicht änderbare Software entwickeln |
| Nichterfüllen der gewünschten Anforderungen | Möglich | Hoch | 3 | Genaue Zeitplanung, gute Arbeitsaufteilung |
| Mangelnde Echtzeit-Performance | Möglich | Hoch | 3 | Kontinuierliche Tests, Messung und Bewertung |
| Datenverlust | Möglich | Schwer-wiegend | 4 | Backups, Versionsverwaltung mit Git |

Tabelle 2: Risikoanalyse

11. Tabellenverzeichnis

12. Abbildungsverzeichnis

13. Quellen

14. Anhang