Tourney
Projektplan

Softwarepraktikum
Wintersemester 2014/2015

Jonas Auer (2860992)
Fabian Biester (2859084)
Jan Tagscherer (2893134)

Universität Stuttgart
28.10.2014

Inhaltsverzeichnis

1 Einleitung

1.1 Zweck der Software

Die Software soll die bisher händisch erfolgte Organisation von Turnieren übernehmen und vereinfachen. Dazu soll das Programm die Organisatoren durch den kompletten Prozess der Turnierplanung begleiten und sie bei den typischen Aufgaben wie der Teilnehmerverwaltung und der eigentlichen Durchführung der Turniere unterstützen.

1.2 Motivation

Momentan werden die Turniere von der Turnierorganisation noch von Hand organisiert. Die große Menge an schriftlichen Notizen und Tabellen ist unübersichtlich und schwierig zu handhaben. Den zuständigen Organisatoren entsteht dadurch ein unnötiger, zeitraubender Aufwand, der gut durch eine entsprechende Software abgefangen werden kann und damit Zeit und Nerven spart. Diese Software würde zu einem übersichtlicheren und leichter zu handhabenden Prozess der Organisation von Turnieren beitragen.

1.3 Projektüberblick

Bei **Tourney** handelt es sich um eine plattformunabhängige Anwendung, die die Durchführung von Spielturnieren durch Automatisierung der folgenden aufwändigen Vorgängen vereinfacht:

- Voranmeldung und Freischaltung von Spielern bei Anwesenheit
- Verteilen der angemeldeten Spieler auf Turniere und Durchführen der Spiele anhand in der Software definierten modularen Regeln
- Rückgabe der Turnierausgänge an den Administrationsteil des Programms und Auswertung derselben

2 Leistungen der Vertragspartner

2.1 Lieferumfang

Zum Lieferumfang des fertigen Produkts gehören:

- Projektplan
- Spezifikation
- Entwurf
- Systemtestprotokoll
- Das ausführbare Programm
- Der Quellcode des Programms
- Modultests
- Übderdeckungsprotokoll der Modultests
- Kurze Installations- und Startanleitung
- Handbuch
- Zeitabrechnung aller Teammitglieder
- Lizenz

2.2 Leistungen des Auftraggebers

- Erstellen eines Konzeptpapiers, das die Anforderungen an das Programm umreißt
- Bereitstellen von Spielregeln für die geforderten mitgelieferten Module

3 Anforderungen an die Umgebung

3.1 Infrastruktur

An den Austragungsorten der Turniere ist eine dauerhafte Netzwerkanbindung nicht zu gewährleisten. Die Kommunikation zwischen den Geräten der beteiligten Organisatoren stellt sich dadurch als kompliziert heraus. Optimal wäre eine dauerhafte WLAN-Anbindung, allerdings wird aus Gründen der Zuverlässigkeit und Verfügbarkeit primär auf den Austausch von Daten über Massenspeichergeräten, wie zum Beispiel USB-Sticks, gesetzt.
Da viele verschiedene Geräte an dem Prozess beteiligt sind und diese zum Teil auf unterschiedliche Betriebssysteme setzen müssen die Daten in einem plattformunabhängigen Format bereitgestellt werden.

4 Entwicklungsprozess

4.1 Phasen der Entwicklung

Die Entwicklung der Anwendung findet in den folgenden Phasen statt:

- **Analyse**:
  - Erstellen eines Fragenkatalogs
  - Durchführung der Kundenbefragung
  - Strukturieren der Anforderungen
  - Erstellen des Projektplans
- **Spezifikation der Anforderungen**:
  - Ordnen, Dokumentieren, Prüfen, Ergänzen und Korrigieren der Anforderungen
- **Architekturentwurf**:
  - Spezifizieren der Module und deren Interaktion
- **Codierung und Modultest**
- **Integration der Module**
- **Systemtest**
- **Abnahme**:
  - Präsentation der fertigen Software

4.2 Dokumentationsplan

Während des Projekts werden die folgenden Dokumente erstellt und gegebenenfalls aktualisiert:

- Strukturierte Analysenotizen aus der Kundenbefragung
- Projektplanung
- Spezifikation
- Benutzerhandbuch
- Aufwandsprotokolle
- Testplan, Testdaten und Testprotokoll

4.3 Prüfung und Qualitätssicherung

Während der Entwicklung wird der Qualität der Software ein hoher Stellenwert beigemessen, sowohl in Hinsicht auf Zuverlässigkeit, Benutzbarkeit als auch Informationssicherheit. Um hohe Qualitätsstandards zu ermöglichen werden die folgenden Maßnahmen ergriffen:

- Einhalten von Standards wie den Java Code Conventions
- Fortlaufende Dokumentation des Programmcodes während der Entwicklung
- Umfassendes Refactoring zur Verbesserung von schlecht strukturiertem Code
- Reviews und eventuelle Korrektur erstellter Dokumente
- Modultests zur Prüfung einzelner Komponenten
- Systemtests zur Verifikation der Software

5 Richtlinien für die Entwicklung

5.1 Konfigurationsmanagement

Die Konfigurationsverwaltung findet mit Git in einem privaten Repository statt, das von BitBucket gehostet wird.

5.2 Design- und Programmierrichtlinien

Der Fokus der Entwicklung liegt auf der intuitiven Bedienung des Programms. Die Software soll auch ohne das Handbuch gelesen zu haben eindeutig zu verstehen sein.
Im Hinblick auf den Programmcode wird Wert auf hohe Qualität gelegt. Dazu zählen die Einhaltung der **Code Conventions for the Java™ Programming Language**, sowie eine konsequente Dokumentation im Code mit Kommentaren und klarer Benennung von Bezeichnern in englischer Sprache.

5.3 Eingesetzte Werkzeuge

Zur Erstellung von Dokumenten und des eigentlichen Programms kommen eine Reihe von Werkzeugen zum Einsatz:

- **Eclipse** - Entwicklungsumgebung
- **GTD-Manager** - Termindrift- und Gantt-Diagramme
- **PearReview** - Review-Organisation und -Protokollierung
- **Pencil** - UI-Prototyp
- **UMLet** - UML-Modellierung
- **CodeCover** - Durchführung von Glass-Box-Tests
- **Testsuite-Management** - Testfallverwaltung
- **Tulip** - Use-Case-Editor
- **Google Sheets** - Aufwandserfassung

6 Entwicklungsplan

6.1 Meilensteine

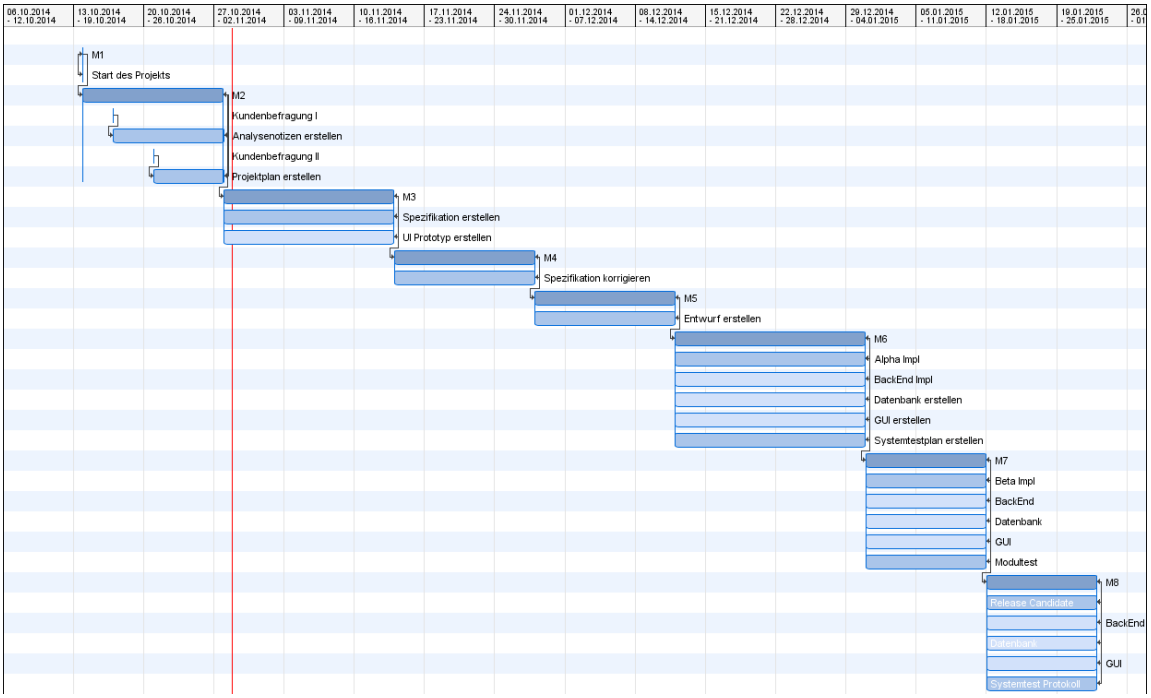
Meilenstein	Name	Dokumente	Datum	Art
M1	Kick-Off	-	14.10.2014	extern
M2	Projektplanung	Analysenotizen, Projektplan	28.10.2014	extern
M3	Spezifikation und UI	Spezifikation, UI-Prototyp	14.11.2014	extern
M4	Korrektur der Spezifikation	Korrigierte Spezifikation, Zwischenstand Zeitabrechnung	28.11.2014	extern
M5	Entwurf	Entwurf	12.12.2014	extern
M6	Alpha	Zwischenstand der Implementierung (Alpha), Systemtestplan	31.12.2014	extern
M7	Beta	Implementierung (Beta), Modultest	12.01.2015	extern
M8	Release Candidate	Implementierung (RC), Systemtestprotokoll	23.01.2015	extern
M9	Abnahme durch den Kunden	-	-	extern
M10	Ende des Softwarepraktikums	-	-	intern

6.2 Arbeitspakete

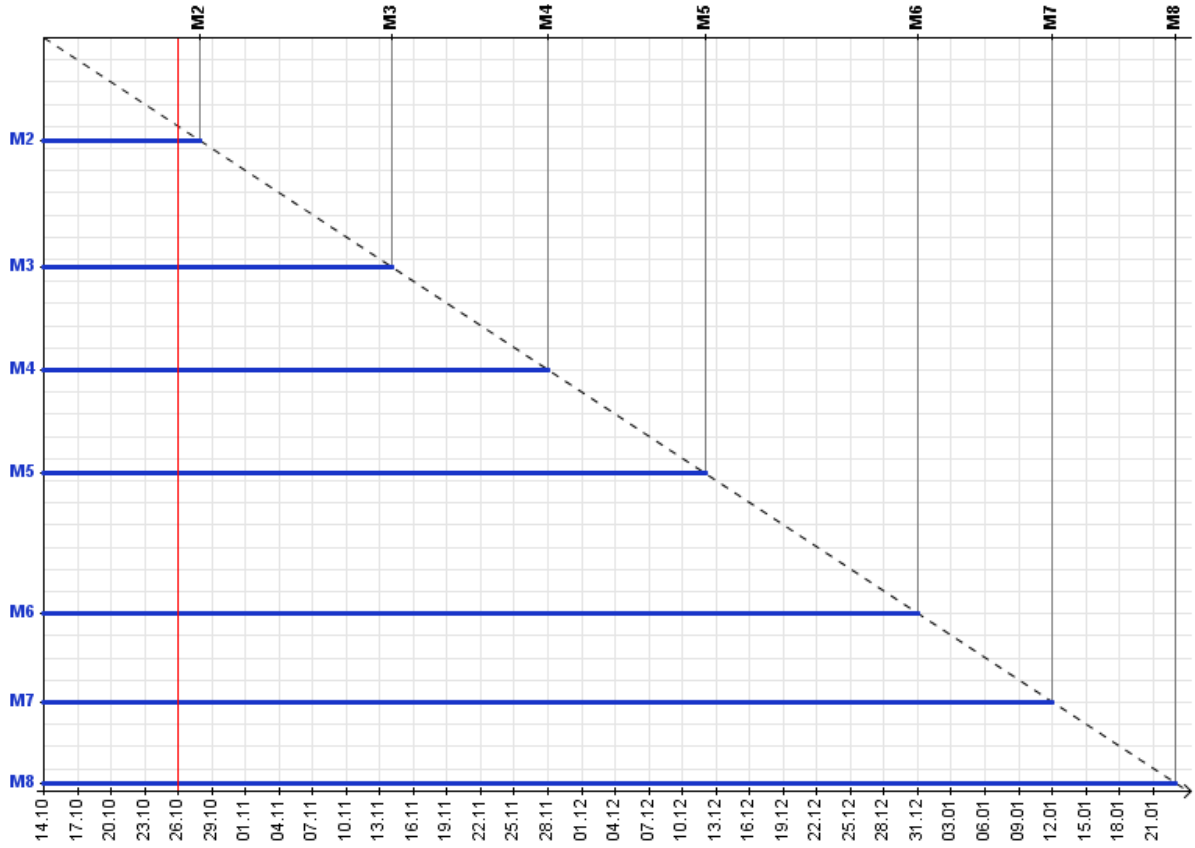
- **Analyse**
  - Kundenbefragung
  - Analysenotizen erstellen
  - Projektplan erstellen
  - Spezifikation verfassen
- **Entwurf**
  - Use-Cases entwerfen
  - UI-Prototyp entwickeln
  - Datenbankintegration entwerfen
  - Administrationsmodul entwerfen
  - Integration der Regelmodule entwerfen
  - Turniermodul entwerfen
- **Implementierung**
  - UI entwickeln
  - Datenbank implementieren
  - Administrationsmodul entwickeln
  - Modularisierung der Turnierregeln implementieren
  - Turniermodul implementieren
  - Komponenten integrieren
  - Geforderte Regelmodule erstellen
- **Test**
  - Nutzerschnittstelle testen
  - Modultests durchführen
  - Systemtest ausführen
  - Testprotokolle verfassen
- **Dokumentation**
  - Aufwandsprotokoll erstellen
  - Benutzerhandbuch verfassen

6.3 Terminplan

6.3.1 Gantt-Diagramm



6.3.2 Termindrift-Diagramm



7 Risiken

7.1 Risiken und ihre Bewertung

1. **Personenausfall**

   Risiko: Hoch
   Es kann nicht ausgeschlossen werden, dass Gruppenmitglieder krankheitsbedingt ausfallen.
   Außerdem wäre denkbar, dass sich ein Mitglied dazu entschließt das Studium abzubrechen oder in einen anderen Studiengang zu wechseln, was zu permanenten Ausfall führen würde. Dieses Risiko wird aber als sehr unwahrscheinlich eingeschätzt.

2. **Falsche Funktionalität wird entwickelt**

   Risiko: Hoch
   Es ist durchaus wahrscheinlich, dass sich die Vorstellungen und Erwartungen an die Funktionalität der Software aus Sicht des Kunden und der der Entwickler unterscheidet. Dies kann dazu führen, dass unnötig Zeit in die Entwicklung von unwichtiger Funktionen fließt und zu wenig Zeit in die Entwicklung von vom Kunden gewünschten Aspekten investiert wird.

3. **Benutzerschnittstelle entspricht nicht den Vorstellungen des Kunden**:

   Risiko: Mittel
   Die genauen Vorstellungen des Kunden an die Benutzerschnittstelle des Programms sind schwer in Worte zu fassen. Es besteht keine Möglichkeit über sämtliche Aspekte der Bedienung mit dem Kunden Rücksprache zu halten.

4. **Termin kann nicht eingehalten werden**

   Risiko: Mittel
   Fehleinschätzungen von Arbeitsaufwänden und unzureichende Planung kann dazu führen, dass Meilensteine und andere wichtige Termine nicht eingehalten werden können.

7.2 Gegenmaßnahmen

1. **Personenausfall**

   Da es unmöglich ist dieses Risiko zu eliminieren, müssen entsprechende Vorkehrungen getroffen werden, für den Fall, dass das Risiko eintritt. Dazu zählt eine gute fortlaufende Dokumentation des aktuellen Stands des Projekts, sowie regelmäßige Kommunikation innerhalb des Teams, um den Überblick über das Projekt synchron zu halten und es den übrigen Mitgliedern gegebenenfalls zu ermöglichen, bei einen temporären Ausfall einzuspringen.
   Fällt ein Teammitglied dauerhaft aus, muss mit dem Betreuer Rücksprache gehalten werden, um die weiteren Maßnahmen zu besprechen. Zu den denkbaren Maßnahmen zählen insbesondere die Einschränkung des Funktionsumfangs, sowie die Verzögerung von Meilensteinen.

2. **Falsche Funktionalität wird entwickelt**

   Durch die klare Priorisierung der wichtigsten vom Kunden geforderten Funktionalitäten kann dieses Risiko teilweise eingedämmt werden. Die wichtigsten Anforderungen an das Produkt lassen sich entweder aus den von dem Kunden bereitgestellten Dokumenten erschließen, oder wurden in den Kundenbefragungen geklärt. Ein Restrisiko kann alerdings niemals ausgeschlossen werden.

3. **Benutzerschnittstelle entspricht nicht den Vorstellungen des Kunden**

   Laut Anforderungen des Kunden liegt bei der Benutzerschnittstelle der Fokus auf der intuitiven Bedienung des Programms. Es wurden keine konkreten Vorstellungen über Layout oder Design geäußert. Es wird daher angestrebt, die Oberfläche und die Menüführung so simpel wie möglich zu halten, sodass auch Personen ohne entsprechenden Kontext zu der Aufgabe des Programms die Software mit Erfolg bedienen können. Sollte die Benutzeroberfläche zu stark von den Vorstellungen des Kunden abweichen, muss noch einmal genauere Rücksprache gehalten werden, um die konkreten Kritikpunkte des Kunden abzuändern.

4. **Termin kann nicht eingehalten werden**

   Gerade zum Beginn des Projekts ist es wahrscheinlich, dass der tatsächliche Aufwand des Projekts unterschätzt wird. Das Team wird versuchen den Aufwand so präzise wie möglich einzuschätzen.
   Kann ein Termin trotzalledem voraussichtlich nicht eingehalten werden, wird Rücksprache mit den zuständigen Betreuern gehalten. Da die einmalige Verschiebung pro Meilenstein unproblematisch ist, wird gegebenenfalls von dieser Möglichkeit gebrauch gemacht. Anschließend haben Maßnahmen besprochen zu werden, wie und ob der folgende Meilenstein wieder rechtzeitig erreicht werden kann.

8 Projektorganisation

8.1 Ansprechpartner und Beteiligte

Kunde:

Heidelberger Spieleverlag
Dr. August-Stumpf-Strasse 7-9
74731 Walldürn
Deutschland

Team 31:

Jonas Auer
Matrikelnummer 2860992
st108421@stud.uni-stuttgart.de

Fabian Biester
Matrikelnummer 2859084
st108056@stud.uni-stuttgart.de

Jan Tagscherer
Matrikelnummer 2893134
st111459@stud.uni-stuttgart.de

Tutor:

Nicolas Mauch
nicosurf89@gmail.com

Betreuer:

Jan-Peter Ostberg
Institut für Softwaretechnologie
Universitätsstraße 38
70569 Stuttgart
Deutschland
jan-peter.ostberg@informatik.uni-stuttgart.de

Ivan Bogicevic
Institut für Softwaretechnologie
Universitätsstraße 38
70569 Stuttgart
Deutschland
ivan.bogicevic@informatik.uni-stuttgart.de

Wolfgang Fechner
Institut für Softwaretechnologie
Universitätsstraße 38
70569 Stuttgart
Deutschland
wolfgang.fechner@informatik.uni-stuttgart.de

9 Versionshistorie

- Version 1.0 (22.10.2014)
  - Grundstruktur des Projektplans erstellt
- Version 1.1 (24.10.2014)
  - Kapitel 1.1, 1.2, 2.1, 3.1, 5.2 und 5.3 hinzugefügt
- Version 1.2 (25.10.2014)
  - Kapitel 2.2, 4.1, 4.2, 5.1 und 6.1 hinzugefügt
- Version 1.3 (26.10.2014)
  - Kapitel 1.3, 4.3, 6.2, 8.1 und 9 hinzugefügt
- Version 1.4 (27.10.2014)
  - Kapitel 7 hinzugefügt
  - Gantt- und Termindriftdiagramm hinzugefügt
- Version 1.5 (29.10.2014)
  - Abkürzungen im Gantt-Diagramm ausgeschrieben
  - Liste der Beteiligten ergänzt