

# **Studienarbeit 2011**

## **Weiterentwicklung einer KSM**

David Henn

Mai 2011

Huppladuppla

### **1 Danksagung**

Schubert

Danke, dass wir selbstständig arbeiten durften und sie ihr Ego zurückgehalten haben

Dreher

Merci for the work

Mischa

Danke fürs mitmachen

Yves

Danke für den Input von außen und die Motivation

## 2 Einleitung

zweiter Teil der Studienarbeit, die sich um KSM dreht Schwerpunkt Bugfixing und Weiterentwicklung

## 3 Projektplanung

Bei der Projektplanung sind für KSM mehrere Dinge zu berücksichtigen.

- Der Zustand des Projekts
- Die knappe Zeit
- Die Aufgaben der Teammitglieder
- Die Anforderungen des Projektinhabers Herr Prof. Schubert

Aufgrund des chaotischen Zustands des Quellcodes, sowie der restlichen Projektartefakte wurde in Absprache mit Herrn Prof. Schubert ein Zwei-Stufen-Plan<sup>1</sup> verabschiedet.

Während der ersten, längeren Phase wird vom Projektteam Bugfixing und Refactoring betrieben, um das Projekt weiter zu ordnen. Um dieses Ziel zu erreichen wird nach dem agilen Entwicklungsmodell *Scrum* vorgegangen.

Nachdem das Projekt stabilisiert wurde, soll in der zweiten Phase die Funktionalität des KSM weiterentwickelt werden.

## 4 Erste Phase

Wie im Kapitel 3 beschrieben, wurde in der ersten Phase des Projektes versucht den Zustand des Gesamtprojektes zu verbessern. Da dies im Allgemeinen viele kurze Aufgaben sind, sowie während dieser Phase ständig neue Aufgaben identifiziert werden, wurde mit Scrum ein äußerst Vorgehensmodell gewählt, das es dem Team ermöglicht diese Aufgaben äußerst flexibel abzuarbeiten.

---

<sup>1</sup>siehe Anhang

## 4.1 Das Scrum Modell

In diesem Abschnitt soll kurz auf das Scrum Modell, sowie der Einsatz in diesem Projekt eingegangen werden.

Im Gegensatz zu traditionellen Entwicklungsmodellen geht Scrum iterativ vor. Einzelne Iterationen werden in Scrum "Sprint" genannt. Vor jedem Sprint legt das Projektteam fest, welche Aufgaben in dieser Zeit abgearbeitet werden sollen. Diese Aufgaben werden im Sprint-Backlog festgehalten. Am Ende eines solchen Sprints muss eine lauffähige Version stehen. Da ein Sprint nur eine Laufzeit von zwei bis vier Wochen hat, kann es durchaus passieren, dass einzelne Aufgaben innerhalb eines Sprints nicht fertiggestellt werden können. Dies ist jedoch nicht schlimm, solange die Version trotzdem lauffähig bleibt. Nicht fertiggestellte Aufgaben werden im nächsten Sprint weitergeführt.

Während eines Sprints stehen die Entwicklern in engem Kontakt zueinander. Jeden Tag gibt es ein kurzes Scrum Meeting, in dem jeder Entwickler kurz darlegt, was er am Tag davor geschafft hat und was er an diesem Tag zu tun gedenkt. Dies stellt sicher, dass stets das gesamte Team weiß, wo sich das Projekt zu diesem Zeitpunkt befindet und was für Entscheidungen getroffen wurden.

Der gesamte Scrum Prozess wird in [DHV11] genau beschrieben.

In KSM Projekt wurde eine Sprintdauer von jeweils zwei Wochen festgelegt. Das Sprint-Backlog, sowie das übergeordnete Projekt-Backlog wurden in Form eines Excel-Dokuments<sup>2</sup> geführt. Die Aufgaben in den Backlogs wurden von den einzelnen Teammitgliedern definiert. Als Scrum Master, der den Prozess überwacht und koordiniert fungierte Herr Tobias Dreher, der durch das Thema seiner Studienarbeit den größten Überblick über die Baustellen des Programms besitzt.

## 4.2 Ergebnis

Während dieser Stabilisierungsphase wurde der Produktreifegrad des KSM Projekts deutlich verbessert.

- Durch viele kleine und größere Änderungen wurden Inkonsistenzen im Design, sowie der Benutzerführung beseitigt

---

<sup>2</sup>Siehe Anhang

- Die Icons des Programms wurden ausgetauscht und durch aussagekräftigere Symbole ersetzt
- Es wurde ein Build Prozess definiert, der es ermöglicht, eine auslieferbare Version des Programms zu generieren
- Es wurden viele Codeleichen entfernt
- eine Projekthomepage wurde erstellt
- Das Projektverzeichnis wurde strukturiert und vereinheitlicht

Durch diese und viele weitere Änderungen wurde die Benutzbarkeit des Programms gesteigert, sowie der Entwicklungsprozess geordnet und vereinheitlicht.

## 5 Weiterentwicklung

Polygone sind eine komplizierte Sache, Geraden, die mittels Funktionen beschrieben werden noch mehr..

## 6 Nachfolger

Run for the Hills idiots!!!

## 7 Fazit

Jippieeee. Vorbei!!!!

## 8 Anhang

Projektplan  
CD

## 9 Literatur

- [Kri07] DAVID KRIESEL: **Ein kleiner Überblick über Neuronale Netze.**  
*www.dkriesel.com*, 2007, Abgerufen am 15.04.2011
- [DHV11] TOBIAS DREHER, MISCHA VOGT, DAVID HENN: **Agile Softwareentwicklung - Einsatz und Werkzeuge im Open Source Umfeld.** Seminararbeit  
Software-Engineering II, 2011