

Exposé

drum-machine

Caspar Raap

W-Seminar: Software-Entwicklung

Inhaltsverzeichnis

1. Extreme Programming.....	2
1.1. Allgemein.....	2
1.2. In diesem Projekt genutzte Elemente:.....	3
2. Mein Thema.....	3
2.1. Allgemeine Informationen.....	3
2.2. Storycards.....	3
2.3. Verortung in der Literatur.....	3
3. Zeitplan.....	4
4. Quellen.....	4

1. Extreme Programming

1.1. Allgemein

Extreme Programming ist eine agile Methode der Softwareentwicklung¹. Das Ziel von agilen Methoden ist es, eine kooperative, flexible Art der Softwareentwicklung in die Tat umzusetzen². Agile Methoden hängen stark von den sie anwendenden Teams ab, so schreibt Don Wells : „*The most important thing to know about Agile methods or processes is that there is no such thing. There are only Agile teams.*“³. Auch Extreme Programming setzt als eine agile Methode einen Schwerpunkt auf Teamwork. Das Team sollte sich, um möglichst effizient zu sein, selbst organisieren⁴.

Extreme Programming beinhaltet 5 Grundwerte⁵:

Einfachheit: Das Team setzt alle nötigen und alle angefragten Features um, aber nicht mehr.

Kommunikation: Jeder ist Teil des Teams und es wird an allen Teilen des Projekts zusammen gearbeitet. Deswegen ist regelmässige Kommunikation zwischen den Teammitgliedern essentiell.

Rückmeldung: Es wird darauf geachtet, die Software früh und häufig vorzuführen und auf die daraus resultierenden Rückmeldungen zu reagieren.

Respekt: Gegenseitiger Respekt ist eine der Grundlagen für eine funktionierende und ertragsreiche Zusammenarbeit.

Mut: Es wird die Wahrheit über den Fortschritt und Einschätzungen gesagt. Es wird davon ausgegangen, dass das Projekt gelingt. Man braucht vor nichts Angst zu haben, da niemand alleine arbeitet. Es wird sich an Veränderungen angepasst, sobald sie eintreten.

Ein Extreme Programming Projekt beginnt bei den "User-Stories". In den User-Stories beschreiben die Nutzer auf Storycards in jeweils ungefähr drei Sätzen Dinge, die die Software für sie beinhalten muss. Im Zuge dessen werden gleichzeitig die "Acceptance-Tests" erstellt, mit denen man überprüfen kann, ob die Storycards erfüllt wurden⁶. Als nächstes wird beim "Release Planning" ein Release Plan erstellt, in dem die Grundzüge des Projektes dargelegt werden. Der wichtigste Teil des Release Planning ist es, die Zeit, die man für die verschiedenen Storycards braucht, einzuschätzen und diese daraufhin in eine Reihenfolge zur Bearbeitung in Abhängigkeit der Wichtigkeit und der benötigten Zeit einzuordnen⁷. Ein wichtiger Einfluss dabei ist die "Projekt Velocity"⁸ die angibt, wie viele Storycards in einer bestimmten Zeit implementiert werden können. Als nächstes wird mit der Arbeit begonnen. Alle 1-3 Wochen wird eine neue Iteration der Software mit den neuesten Errungenschaften herausgegeben, die daraufhin mithilfe der Acceptance-Tests auf Funktionalität überprüft werden. Die Rückmeldungen, die von den Nutzern kommen, wird bei der nächsten Iteration mit in den Entwicklungsprozess einbezogen. Versionen, die die Zustimmung der Nutzer bekommen, werden in kleinen Releases veröffentlicht⁹. Ein wichtiger Kniff in der technischen

1 Wells, Don : <http://www.extremeprogramming.org>

2 Wells, Don : <http://www.agile-process.org>

3 Wells, Don : <http://www.agile-process.org>

4 Wells, Don : <http://www.extremeprogramming.org>

5 Wells, Don : <http://www.extremeprogramming.org/values.html>

6 Wells, Don : <http://www.extremeprogramming.org/rules/userstories.html>

7 Wells, Don : <http://www.extremeprogramming.org/rules/planninggame.html>

8 Wells, Don : <http://www.extremeprogramming.org/rules/velocity.html>

9 Wells, Don : <http://www.extremeprogramming.org/map/project.html>

Entwicklung ist, dass die Unit-Tests; Code der prüft, ob ein Softwareteil das tut, was es soll; vor dem eigentlichen Code programmiert werden. Dies ermöglicht, den Simpelsten Code zu schreiben, der die gewünschten Aufgaben erfüllt¹⁰.

1.2. In diesem Projekt genutzte Elemente:

In meinem Projekt möchte ich nicht alle Elemente des Extreme Programming einsetzen, da dies unpraktikabel wäre. Ich beschränke mich deshalb auf einige wenige, die auf die Größe meines Projektes passen.

Umgesetzt werden :

- Die 5 Grundwerte:
 - Ich werde den Code darauf beschränken nur das zu tun, was zur Erfüllung der Storycards notwendig ist.
 - Ich werde mich mit so vielen Leuten wie möglich konstruktiv über mein Projekt austauschen.
 - Ich werde die Rückmeldungen der Testnutzer und der Kommunikationspartner ernst nehmen und berücksichtigen.
 - Ich werde Mitwirkende (Testnutzer usw.) respektieren, auch wenn sie eventuell nicht so viel Ahnung von der Materie haben.
 - Ich werde nicht zögern, mir für dieses Projekt eventuell nötiges Zusatzwissen anzueignen und schwierige Probleme oder Features offen anzugehen.

-Das Grundmodell der Planung inklusive Storycards in Verbindung mit der Entwicklung und des separaten Testens einzelner Iterationen

2. Mein Thema

2.1. Allgemeine Informationen

Mein Projekt ist ein virtuelles Instrument mit dem Namen „Drum-Machine“, in dem man mithilfe von vier Schlagzeuggeräuschen Rythmen frei erstellen kann.

2.2. Storycards

Storycard	Testsituation
Die Software soll vier Töne(eines Schlagzeuges) besitzen	Es gibt vier Schlagzeuggeräusche
Das Abspielen der einzelnen Töne soll programmierbar sein	Man kann frei bestimmen, wann ein Geräusch im Takt abgespielt wird
Die Länge des programmierbaren Bereiches soll 2 Takte betragen	Es gibt 2 Takte, in denen man die Geräusche setzen kann
Jeder Takt soll in acht Noten aufgeteilt sein	In jedem Takt gibt es 8 voneinander gleich

¹⁰ Wells, Don : <http://www.extremeprogramming.org/rules/testfirst.html>

(8tel-Noten) die einzeln ansteuerbar sind	entfernte Stellen an denen man Geräusche platzieren kann
Die Anzahl der 4tel-Noten pro Minute (”BPM“) soll frei wählbar sein	Es wird die angegebene Anzahl an Viertelnoten pro Minute in gleichmässigem Abstand gespielt
Das Abspielen der Töne soll pausierbar sein	Das Abspielen der Töne kann gestoppt und wieder gestartet werden

2.3. Verortung in der Literatur

Man kann „Drum-Machine“ als einen Softwaresynthesizer einordnen, da es ein Programm zur Erzeugung von Ton ist¹¹. Spezifischer ist das Programm als ein Software Sampler zu betrachten, also eine Software zur Emulierung der Funktionen eines Samplers¹². Ein Sampler ist ein Instrument das aufgenommene Töne abspielen kann¹³. Diese können in einer Reihenfolge programmiert und anschliessend abgespielt werden¹⁴. Des Weiteren ist das Programm wie die meisten Sampler polyphon¹⁵¹⁶.

3. Zeitplan

Bis 2.6.17	Exposé fertigstellen und abgeben
Bis 2.10.17	Arbeit fertigstellen mit Dokumentation
Bis 30.10.17	Formulierung fertigstellen

4. Quellen

Wells, Don : <http://www.extremeprogramming.org> , zuletzt abgerufen am 20.5.17

Wells, Don : <http://www.agile-process.org> , zuletzt abgerufen am 20.5.17

https://en.wikipedia.org/wiki/Software_synthesizer , zuletzt abgerufen am 21.5.17

https://en.wikipedia.org/wiki/Software_sampler , zuletzt abgerufen am 21.5.17

[https://en.wikipedia.org/wiki/Sampler_\(musical_instrument\)](https://en.wikipedia.org/wiki/Sampler_(musical_instrument)) , zuletzt abgerufen am 21.5.17

[https://de.wikipedia.org/wiki/Polyphonie_\(Elektrofon\)](https://de.wikipedia.org/wiki/Polyphonie_(Elektrofon)) , zuletzt abgerufen am 21.5.17

11 https://en.wikipedia.org/wiki/Software_synthesizer

12 https://en.wikipedia.org/wiki/Software_sampler

13 [https://en.wikipedia.org/wiki/Sampler_\(musical_instrument\)](https://en.wikipedia.org/wiki/Sampler_(musical_instrument))

14 [https://en.wikipedia.org/wiki/Sampler_\(musical_instrument\)](https://en.wikipedia.org/wiki/Sampler_(musical_instrument))

15 [https://en.wikipedia.org/wiki/Sampler_\(musical_instrument\)](https://en.wikipedia.org/wiki/Sampler_(musical_instrument))

16 [https://de.wikipedia.org/wiki/Polyphonie_\(Elektrofon\)](https://de.wikipedia.org/wiki/Polyphonie_(Elektrofon))