

*Scrum in Projekten an der FH Salzburg*

**BACHELORARBEIT 2**

StudentIn Matthias Frick, 0810601010  
BetreuerIn DI Brigitte Jellinek

Salzburg, am 01. Juli 2011

## **Eidesstattliche Erklärung**

Ich erkläre hiermit eidesstattlich, dass ich die vorliegende Bachelorarbeit selbständig und ohne fremde Hilfe verfasst, und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Weiters versichere ich hiermit, dass ich die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission weder im In- noch im Ausland vorgelegt und auch nicht veröffentlicht.

Datum: 01. Juli 2011

Unterschrift

## Kurzfassung

Vor- und Zuname: Matthias Frick  
Institution: FH Salzburg  
Studiengang: Bachelor MultiMediaTechnology  
Titel der BA: Scrum in Projekten an der FH Salzburg  
Begutachter: DI Jellinek Brigitte

Die Bachelorarbeit befasst sich mit der Thematik „Scrum in Projekten an der FH Salzburg“. Im Speziellen wird auf die Forschungsfrage „Wie kann der Einsatz von Scrum in Projekten der Studiengänge Multimedia Art und Multimedia Technology, an der Fachhochschule Salzburg, sinnvoll funktionieren?“ eingegangen. Zuerst wird Scrum gemäß der Literatur erläutert und interpretiert. Hierbei werden die unterschiedlichen Ausführungen verschiedener Autoren miteinander verglichen und Unterschiede aufgezeigt. Anschließend wird mithilfe von zwei durchgeführten Interviews mit Scrum-ExpertInnen die Situation im Bezug auf Scrum an der Fachhochschule Salzburg analytisch diskutiert und dargestellt. Die daraus gewonnenen Erkenntnisse werden in einem Fahrtenplan für zukünftige Studierende und Lehrende festgehalten. Die Vorgangsweise in diesem Fahrtenplan spiegelt die Beantwortung der Fragestellung wider. Mit einem Fazit wird diese Bachelorarbeit abgeschlossen.

*Scrum, FHS-Projekte, Prozessablauf, Projektmanagement, Multimedia Art, Multimedia Technology*

## Abstract

This bachelor thesis is about „Scrum in projects at the University of Applied Sciences Salzburg“. In particular the following research question should be answered: „How could Scrum be used in projects of the study courses Multimedia Art and Multimedia Technology at the University of Applied Sciences in a reasonable way?“. First of all Scrum is defined and explained by the means of specialized literature. By doing so the varying explanations from different authors are compared to each other and the differences are pointed out. Afterwards two interviews with Scrum experts analytically discusses and presents the situation considering Scrum at the Univesity fo Applied Sciences Salzburg. The optained results of these interviews are developed into a schedule for future reference for both students and teachers. This will serve as a reflection of the research question and will be the conclusion of this bachelor thesis.

*Scrum, projects at the University of Applied Sciences Salzburg, process operation, project management, Multimedia Art, Multimedia Technology*

# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis</b>	<b>1</b>
<b>1 Einleitung</b>	<b>2</b>
1.1 Relevanz . . . . .	2
1.2 Forschungsgebiet und Ziel . . . . .	2
1.3 Aufbau . . . . .	3
<b>2 Scrum</b>	<b>4</b>
2.1 Scrum - Allgemeines . . . . .	4
2.1.1 Definition . . . . .	4
2.1.2 Geschichtliche Entwicklung . . . . .	4
2.2 Scrum - Rollen . . . . .	5
2.2.1 Scrum Master . . . . .	6
2.2.2 Product Owner . . . . .	6
2.2.3 Entwicklerteam . . . . .	8
2.2.4 Sonstige Rollen . . . . .	9
2.3 Scrum - Voraussetzungen . . . . .	10
2.3.1 Teamgröße . . . . .	10
2.3.2 Teammitglieder & Teamfähigkeit . . . . .	10
2.3.3 Infrastruktur . . . . .	11
2.3.4 Continuous Integration . . . . .	11
2.4 Scrum - Hilfreiches . . . . .	13
2.4.1 Pair Programming . . . . .	13
2.4.2 Test Driven Development . . . . .	14
2.4.3 Story Driven Development . . . . .	14
2.5 Scrum - Artefakte . . . . .	15
2.5.1 Product Backlog . . . . .	15
2.5.2 Sprint Backlog . . . . .	15
2.5.3 Release-Burndown . . . . .	16
2.5.4 Sprint-Burndown . . . . .	17
2.6 Scrum - Timeboxen . . . . .	18
2.6.1 Release-Planning . . . . .	18
2.6.2 Sprint-Planning . . . . .	18
2.6.3 Sprint . . . . .	20
2.6.4 Daily-Scrum . . . . .	21
2.6.5 Sprint Review . . . . .	22
2.6.6 Sprint Retrospektive . . . . .	22
2.7 Scrum - Zyklus . . . . .	23

<b>3 Scrum an der Fachhochschule Salzburg</b>	<b>25</b>
3.1 Gegebenheiten an der Fachhochschule Salzburg (FHS)	26
3.1.1 Projekttypen durch Ausbildungsschwerpunkte	26
3.1.2 Projekttypen nach Abläufen	26
3.1.3 Projektumfang	27
3.1.4 Potenzielle Scrum-Projekte	29
3.1.5 Vorwissen	29
3.1.6 Hindernisse und Ablenkungen	30
3.1.7 Entlohnung	30
3.2 Scrum - Rollen - Situation an der FHS	31
3.2.1 Scrum Master	31
3.2.2 Product Owner	32
3.2.3 Entwicklerteam	35
3.2.4 Sonstige Rollen	35
3.3 Scrum - Voraussetzungen - Situation an der FHS	38
3.3.1 Teamgröße	38
3.3.2 Teammitglieder & Teamfähigkeit	39
3.3.3 Infrastruktur	40
3.4 Scrum - Hilfreiches - Situation an der FHS	46
3.5 Scrum - Artefakte - Situation an der FHS	47
3.6 Scrum - Timeboxen - Situation an der FHS	48
3.6.1 Release-Planning	48
3.6.2 Sprint-Planning	49
3.6.3 Sprint	49
3.6.4 Daily-Scrum	49
3.6.5 Sprint Review	50
3.6.6 Sprint Retrospektive	50
<b>4 Fahrtenplan</b>	<b>51</b>
4.1 Ideale Variante	51
4.2 Kompromiss-Variante	55
<b>5 Fazit</b>	<b>59</b>
<b>Literatur</b>	<b>60</b>
<b>Abbildungsverzeichnis</b>	<b>63</b>
<b>Tabellenverzeichnis</b>	<b>64</b>

<b>A Interviews</b>	<b>A-1</b>
A.1 Interview mit Frau Jeanny Gucher . . . . .	A-1
A.2 Interview mit Herrn Stefan Randelshofer . . . . .	A-6
<b>B E-Mails</b>	<b>B-1</b>

## Abkürzungsverzeichnis

- bzw.** beziehungsweise
- D.h.** Das heißt
- DPM** Design- & Produktmanagement
- FH** Fachhochschule
- FHS** Fachhochschule Salzburg
- ITS** Informationstechnik & System-Management
- MMA** Multimedia Art
- MMT** Multimedia Technology
- PP** Pair Programming
- SDD** Story-Driven-Development
- TDD** Test-Driven-Development
- vgl.** vergleiche
- z.B.** zum Beispiel

# 1 Einleitung

„While working on a SCRUM development team is intense, developers are rewarded by high team spirit, a deep sense of accomplishment, and a feeling that development can be an enjoyable and satisfying experience.“ (Schwaber 1997, S. 21)

## 1.1 Relevanz

Das Agile Manifest dient nun schon seit zehn Jahren (Stand: 02.05.2011) als wichtiger Anhaltspunkt bei der Entwicklung von Software-Projekten und verankert Scrum als eine essentielle agile Entwicklungsmethode (vergleiche (vgl.) Abschnitt 2.1.2). Dies verdeutlicht, dass sich agile Entwicklungsmethoden etabliert haben und unverzichtbar sind. Dabei ist insbesondere das Vorgehensmodell Scrum nicht mehr aus dem Prozess der Software Entwicklung wegzudenken. Laut Frau Gucher ist Scrum mittlerweile die „State-of-the-art“-Methode geworden, um Software-Projekte zu managen und zu steuern (vgl. A.1, Zeile 16ff). Neben der Meinung von ExpertInnen wie Frau Gucher spricht auch der verstärkte Einsatz dieses Vorgehensmodell für die Aktualität und Relevanz dieser Thematik (West & Grant 2010, S. 2). Unternehmen wie beispielsweise Microsoft, Oracle oder Yahoo setzen ebenfalls Scrum ein (Sixtensson 2011, S. 15). Scrum kann aber nicht nur im professionellen Arbeitsumfeld beziehungsweise (bzw.) in der Industrie eingesetzt werden, sondern könnte auch in anderen Einsatzgebieten der Software Entwicklung eine tragende Rolle bei der Realisierung von Projekten einnehmen.

An der FHS wird zum jetzigen Stand (28.04.2011) im Rahmen des Bachelorstudiengangs Multimedia Technology (MMT) im 3. Semester die integrierte Lehrveranstaltung „Software Engineering“ gehalten. In dieser Lehrveranstaltung liegt laut Curriculum ein Schwerpunkt auf dem Themengebiet der „Agilen Softwareentwicklung“ (MMT 2008). Scrum wird in diesem Rahmen gelehrt und soll deshalb auch von Studierenden verwendet werden.

Daraus wird ersichtlich, dass auch an Hochschulen Projekte verstärkt mit Scrum umgesetzt werden sollten und diese Entwicklungsmethode auch in anderen Anwendungsgebieten an Bedeutung und Relevanz gewinnt.

## 1.2 Forschungsgebiet und Ziel

Daher soll in dieser Arbeit erforscht werden, inwiefern und mit welchen Mitteln ein Scrum-Prozess auch im Umfeld einer Hochschule genutzt werden kann. Aus diesem Grund befasst sich diese Bachelorarbeit mit dem Thema „Scrum in Projekten an der FHS“. Konkret wird auf die Forschungsfrage „Wie kann der Einsatz von Scrum in Projekten der Studiengänge Multimedia Art (MMA) und MMT, an der FHS, sinnvoll funktionieren?“ eingegangen.

Bei der Beantwortung der Forschungsfrage ist auch die Analyse von Scrum ein wichtiges Ziel. Warum Scrum überhaupt zum Einsatz kommen soll und nicht andere Vorgehensmodelle aus dem Bereich des Software Engineerings angewendet werden, nimmt ebenfalls einen grundlegenden Aspekt ein. Solche Vorgehensmodelle könnten unter anderem zum Beispiel (z.B.) das Wasserfallmodell (Sommerville 2001, S. 57f.) oder spiralförmige Entwicklungen, wie das Spiralmodell, (Sommerville 2001, S. 65f.) sein. In dieser Arbeit hat eine Abgrenzung von anderen Vorgehensmodellen zu erfolgen, um die Forschungsfrage korrekt beantworten zu können.

Das Ergebnis dieser Bachelorarbeit wird ein Fahrtenplan für zukünftige StudentInnen, Lehrende und für Studiengangleitungen sein. Mithilfe dieses Fahrtenplans soll es möglich sein, alle Voraussetzungen für den sinnvollen Einsatz von Scrum in FHS-Projekten schaffen zu können.

### 1.3 Aufbau

Diese Arbeit soll neben der Analyse von Scrum auch einen Fahrtenplan für zukünftige Software-Prozesse an der FHS hervor bringen. Damit ein korrekter Fahrtenplan erarbeitet werden kann, muss zunächst auf den Scrum-Prozess selbst und dessen Funktionalität eingegangen werden. Daher werden, nach einer kurzen allgemeinen Einführung, zunächst wichtige Begrifflichkeiten wie Rollen, Voraussetzungen, Artefakte und Timeboxen erklärt. Zusätzlich wird noch Hilfreiches in Bezug auf Scrum vorgestellt. Die Begrifflichkeiten sind für das Verständnis von Scrum wichtig. LeserInnen sollen dadurch einen umfassenden Einblick über Scrum und Basiswissen über die Thematik erlangen.

Dies ist die Grundvoraussetzung, um im Kapitel „Scrum an der Fachhochschule Salzburg“ die konkrete Situation an der FHS verstehen zu können. Sind diese Grundlagen erläutert, soll eine Analyse des Scrum-Prozesses an der FHS durchgeführt werden. Die Untersuchung wird durch Interviews von zwei ExpertInnen unterstützt und liefert so einen Überblick über die Situation an der FHS. Die so gewonnenen Erkenntnisse bilden zudem die Basis für die Beantwortung der Forschungsfrage und werden auch in die Aufstellung des Fahrtenplans mit einfließen.

Im Anschluss wird dieser eigentliche Fahrtenplan im Detail zusammengestellt. Er soll zukünftigen StudentInnen, LektorInnen und Studiengangleitungen aufzeigen, wie Scrum sinnvoll an der FHS bzw. in Projekten der Studiengänge MMA und MMT eingesetzt werden kann. Der Fahrtenplan wird in zwei verschiedenen Varianten („Ideal-“, und „Kompromiss“-Variante) zusammengestellt.

Abschließend werden im Fazit die wichtigsten Ergebnisse zusammenfasst und ein Ausblick über zukünftige Möglichkeiten von Scrum an der FHS gegeben.

## 2 Scrum

Dieses Kapitel beschäftigt sich mit Scrum gemäß der Literatur. Zuerst wird Scrum im Allgemeinen kurz vorgestellt. Dabei wird der Begriff „Scrum“ definiert, Geschichtliches erläutert und weitere Entwicklungen ausgeführt. Als nächstes werden die verschiedenen Rollen und Voraussetzungen, welche im Scrum-Prozess gegeben sind bzw. erfüllt werden müssen, näher erläutert. Anschließend werden hilfreiche Tools zur Entwicklung im Scrum-Prozess vorgestellt. Die nächsten Unterpunkte sind Scrum Artefakte und Scrum Timeboxen. Das Kapitel schließt mit dem Scrum Zyklus, der alle bis dahin erläuterten Punkte und deren Zusammenhänge zusammenfasst.

### 2.1 Scrum - Allgemeines

#### 2.1.1 Definition

Scrum ist ein Vorgehensmodell der agilen Softwareentwicklung zur Durchführung von Projekten. Die Arbeit wird dabei in einem iterativ ablaufenden Prozess erledigt. Normalerweise hat eine Iteration eine Länge von 30 Tagen. Dies ist jedoch je nach gewünschtem Release-Zyklus und eventuell bestehenden Abhängigkeiten zu externen KundInnen auch auf zwei Wochen verkürzbar. (Larman 2003, S. 109)

Pichler beschreibt Scrum als ein agiles Managementframework zur Entwicklung von Software. Dieses Framework gibt den Rahmen des Entwicklungsprozesses an. (Pichler 2008, S. 1) Wirdemann sieht Scrum, laut seinen Ausführungen, ebenfalls als Framework an. Das Framework gibt dabei die Grundstruktur inklusive Rahmen vor. Die Ausmaße des Rahmens werden aber von Mitgliedern eines Projektes selber ausgefüllt und so geformt, dass das beste Ergebnis erzielt werden kann. (Wirdemann 2009, S. 27)

#### 2.1.2 Geschichtliche Entwicklung

Scrum im klassischen Sinn wurde von Ken Schwaber, Mike Beedle und Jeff Sutherland entwickelt. Im Jahre 1995 wurde Scrum erstmals formell vorgestellt und veröffentlicht. (Schwaber & Sutherland 2010, S. 2) Mittlerweile existieren Erweiterungen von Scrum im klassischen Sinn, wie beispielsweise von Boris Gloger. Dieser fügt noch zusätzliche Elemente wie beispielsweise die Rolle „Manager“ hinzu (Gloger 2010, S. 13). Diese Bachelorarbeit befasst sich jedoch nur mit Scrum im klassischen Sinn und allen damit verbundenen Rahmenbedingungen und Bestandteilen.

Die Ursprünge von Scrum sind weiter zurückzuführen. Im Jahre 1986 wurde eine Vorgängervariante von Scrum im klassischen Sinn im Harvard Business Review Paper „The New New Product Development Game“ von Ikujiro Nonaka und Hirotaka Takeuchi erwähnt. Laut Takeuchi und Nonaka soll Scrum den Entwicklungsprozess beschleunigen und flexibler machen. Sie beschreiben in diesem ersten Artikel über Scrum in Produktionsumgebungen, dass sechs Merkmale das eigentliche Geheimnis sind, wie die Prozessentwicklung verbessert werden kann: (Takeuchi & Nonaka 1986)

1. Built-In instability
2. Self-organizing project teams
3. Overlapping development phases
4. „Multilearning“
5. Subtle control
6. Organizational transfer of leading

These characteristics are like pieces of a jigsaw puzzle. Each element, by itself, does not bring about speed and flexibility. But taken as a whole, the characteristics can produce a powerful new set of dynamics that will make a difference.

Takeuchi und Nonaka kamen zum Inhalt dieses Papers basierend auf durchgeführten Fallstudien in Firmen aus der Computer-, Drucker- und Autoindustrie. Ausschlaggebend für die Selektion dieser Branchen war der hohe Anteil an Softwareentwicklung innerhalb dieser. Sie nennen in ihren Ausführungen den neuen Prozessablauf „the holistic or rugby approach“ und wollen damit die Herangehensweise demonstrieren. Bei Scrum findet der gesamte Prozessablauf im Rahmen eines Teams über mehrere, sich überlappende Phasen, statt. Da das Team als eine Einheit agiert und in Selbstorganisation und Eigenregie arbeitet, ziehen Takeuchi und Nonaka die Verbindung zum Rugby-Sport. (Takeuchi & Nonaka 1986)

Schwaber, Beedle und Sutherland haben in ihren Weiterentwicklungen von Scrum im klassischen Sinn von den ersten Ausführungen von Takeuchi und Nonaka profitiert. Die Selbstorganisation eines Teams und der iterative Prozessablauf stehen bei Scrum im klassischen Sinn im Mittelpunkt.

Scrum ist eine agile Entwicklungsmethode und erfüllt deshalb die Bedingungen der agilen Softwareentwicklung. Um dies besser verstehen zu können, muss zuerst der Begriff „agil“ erklärt werden:

„agil“ ist ein Synonym für „beweglich“, „wendig“. Im Bezug auf Softwareengineering schließt der Begriff die Effektivität und Beweglichkeit in sich ein. Darunter ist zu verstehen, dass ein agiler Prozess zu jeder Zeit beweglich und effektiv sein muss. Ein still stehender Prozess kann deshalb nie agil sein, dies trifft gleichzeitig auf einen nicht effektiven Ablauf zu. (Cockburn 2002, S. 178)

Die Grundsätze bzw. Werte der agilen Softwareentwicklung wurden im Jahre 2001 im Agilen Manifest festgehalten. (Highsmith 2001) Die Autoren des Agilen Manifests sind folgende Personen: Ken Schwaber, Jeff Sutherland, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin und Dave Thomas. Zu den Bedingungen zählen folgende Punkte: (Beck et al. 2001)

1. Individuals and interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Responding to change over following a plan

Diese Grundsätze bzw. Werte spielen im Scrum-Prozess eine wichtige Rolle, da sie Teile der später näher erläuterten Rahmenbedingungen beeinflussen und dafür sorgen, dass Scrum überhaupt erst funktionieren kann.

## 2.2 Scrum - Rollen

Scrum im klassischen Sinn hat drei Rollen: Scrum Master, Product Owner und das Entwickler-team. Alle diese drei Rollen zusammen werden als Scrum-Team bezeichnet und alle Mitglieder werden „Pigs“ genannt. Im vorliegenden Kapitel werden diese Rollen beschrieben, erläutert und vorgestellt. Zusätzlich werden noch sonstige Rollen wie Außenstehende, so genannte „Chickens“, und die Stakeholder-Rolle erläutert, da diese den Scrum-Prozess ebenfalls beeinflussen. Es werden die einzelnen Verantwortlichkeiten der jeweiligen RolleninhaberInnen erläutert, Aufgabenbereiche eingegrenzt und die idealen Eigenschaften für jede Rolle festgehalten.

A chicken and a pig are together when the chicken says, „Let’s start a restaurant!“ The pig thinks it over and says, „What would we call this restaurant?“ The chicken says, „Ham n’ Eggs!“ The pig says, „No thanks, I’d be committed, but you’d only be involved!“

(Schwaber & Sutherland 2010, S. 6)

### 2.2.1 Scrum Master

Die Scrum MasterIn ist eine Person in einem Scrum-Team. Sie hat geteilte Arbeitsaufgaben und arbeitet ca. 50% in der Programmierung und 50% im Management. Eine Scrum MasterIn leitet das Daily-Scrum-Meeting (vgl. 2.6.4), das Sprint Review (vgl. 2.6.5) und die Sprint Retrospektive (vgl. 2.6.6). (Larman 2003, S. 115)

Scrum MasterInnen sind die ChefInnen über den Prozess und den Prozessablauf, sie sind aber keine Vorgesetzten der anderen Scrum-Team Mitglieder. Eine Scrum MasterIn kann beispielsweise kein anderes Teammitglied kündigen. Sie kann nur prozessfördernde Maßnahmen, wie beispielsweise die Länge von Sprints (siehe 2.6.3) von vier auf zwei Wochen zu verringern, anordnen. (Cohn 2010, S. 117) Cohn schreibt des Weiteren, dass die Scrum MasterIn weiters die Rolle des Personal-Trainers des Scrum-Teams einnehmen kann. Sie hält sozusagen den Ablauf „in Schwung“. Eine Scrum MasterIn muss dafür sorgen, dass das Scrum-Team Scrum richtig anwendet und alle Richtlinien und Regeln des Ablaufes einhält. Die Scrum MasterIn hilft dem Scrum-Team und der dahinter stehenden Organisation Scrum einzuführen. Sie ist dabei behilflich, dem Team Selbstorganisation und gemeinsames Arbeiten anzueignen.

Eine Scrum MasterIn sorgt dafür, dass das Scrum-Team ohne Probleme und Hindernisse arbeiten kann. Sie beseitigt „Hindernisse“ - so genannte „Impediments“, wie beispielsweise eine nicht funktionierende Testumgebung. Sie steht jeder EntwicklerIn für Rückfragen und Probleme zur Verfügung und bildet die Achse zwischen Management (Product OwnerIn (vgl. 2.2.2)) und dem Entwicklerteam (vgl. 2.2.3) (Schwaber & Sutherland 2010, S. 6f). Eine gute Scrum MasterIn ist willig, Verantwortung zu übernehmen. Hier ist wichtig festzuhalten, dass nicht die Scrum MasterIn alleine für den Erfolg eines Scrum-Teams die Verantwortung trägt. Diese ist innerhalb des Scrum-Teams aufgeteilt. Die Scrum MasterIn ist aber dafür zuständig, dass das bestmögliche Ergebnis am Ende eines Sprints (Sprint siehe 2.6.3) herauskommt.

Sie hat eine Art Mediatorrolle zwischen dem Management und dem Scrum-Team. Dabei ist es wichtig, dass die Scrum MasterIn selber auch programmiert und im Prozess mitarbeitet. Nur so kann sie sich in die Situation des Entwicklerteams versetzen und tatsächlich am Commitment (Commitment vgl. 2.6.2) teilhaben. Dadurch ist sie selber überzeugt, dass das Commitment geschafft wird und sie versucht alles dafür zu tun, um den Prozess nicht zum Stocken geraten zu lassen. Erfolgreiche Scrum MasterInnen beeinflussen, bei korrekter Arbeitsweise, den Scrum-Prozess auf positive Art und Weise. Der Erfolg getaner Arbeit wird dadurch erhöht. (Cohn 2010, S. 120) Scrum MasterInnen sollten sich aber nicht in den Mittelpunkt stellen, sondern versuchen, ihrem Team schnellstmöglich eigenständiges Arbeiten beizubringen, sodass die Rolle der Scrum MasterIn in den Hintergrund rückt. Hat das Scrum-Team erfolgreich gelernt wie Scrum einzusetzen ist, so wird die Rolle der Scrum MasterIn zwar nicht überflüssig, aber es wird dann keine Vollzeit Scrum MasterIn mehr benötigt. Dadurch können Scrum MasterInnen sich mehr auf das eigentliche Programmieren konzentrieren und dadurch gute Arbeit leisten. (Pichler 2008, S. 19f)

Eine gute Scrum MasterIn ist laut Cohn verantwortungsvoll, bescheiden, gemeinschaftlich, engagiert, einflussreich und verfügt über ein gutes Fachwissen, ist sachkundig. (Cohn 2010, S. 119f) Idealerweise wird die Rolle der Scrum MasterIn vom Scrum-Team selber bestimmt, da es sich schließlich um eine sich selbst organisierende Einheit handelt. Bei noch nicht vorhandenem Scrum-Team oder ersten Versuchen, Scrum überhaupt einzusetzen, wird es schwieriger. In diesen Fällen muss die verantwortliche EntwicklungsleiterIn gemeinsam mit dem Management entscheiden. (Pichler 2008, S. 22)

### 2.2.2 Product Owner

Die Product OwnerIn repräsentiert laut Pichler die Endkundenbedürfnisse. Sie steuert die Entwicklung über die Priorisierung des Product Backlogs (siehe 2.5.1). Die enge Zusammenarbeit mit dem Team über den gesamten Prozessverlauf des Projektes steht dabei im Vordergrund. Die Product OwnerIn nimmt im Scrum-Prozess eine zentrale Stelle ein. Sie ist die Achse zwischen

Stakeholdern und Entwicklerteam (vgl. 2.2.3). Somit vereint die Rolle Produkt- und Projektmanagementaufgaben und ist gleichzeitig fest in die Softwareentwicklung integriert. (Pichler 2008, S. 9)

Laut Pichler wird die Rolle der Product OwnerIn häufig mit einer ProduktmanagerIn oder einer MarketingmitarbeiterIn besetzt. (Pichler 2008, S. 10). Schwaber und Sutherland sagen, dass die Rolle auch von einer Person aus dem Entwicklerteam besetzt werden kann. Die Product OwnerIn kann nicht gleichzeitig die Scrum MasterIn sein. (Schwaber & Sutherland 2010, S. 7)

Die Rolle der Product OwnerIn hat in der Regel eine Person inne. Es gibt Ausnahmen, bei denen es ein kleines Team von zwei Personen mit dieser Rolle gibt. Wenn der Scrum-Prozess erstmalig eingesetzt wird, hat meistens die Scrum-MasterIn die größten und umfangreichsten Arbeiten zu erledigen. Ist das Scrum-Team eingespielt und hat Erfahrungen gesammelt, so wird die Product OwnerIn immer wichtiger und entscheidender. Die Rolle der Product OwnerIn wird deshalb als die stressigste Rolle bezeichnet. Aufgrund dessen kann wegen Überlastung einer Person auch ein Product-Owner-Team existieren. Hierbei ist aber wichtig festzuhalten, dass dieses Team wiederum eine Einzelperson hat, die bei Bedarf selbstständig und alleine Entscheidungen treffen kann und dies auch tut. Dies ist besonders wichtig, da Scrum-Teams während des laufenden Prozesses keine langwierigen Entscheidungsprozesse eines etwaigen Komitees abwarten können. Diese würden die Dauer des Projektes und somit dessen Erfolg negativ beeinflussen. Daher müssen wichtige Fragestellungen einfach und schnell beantwortet werden. (Cohn 2010, S. 128f)

Schwaber und Sutherland sehen die Möglichkeit des Vorhandenseins eines Product-Owner-Teams gar nicht. Laut ihren Ausführungen gibt es immer nur eine Product OwnerIn. Hat diese zu viel Arbeit, kann sie durch ein Komitee unterstützt werden. Die Rolle „Product Owner“ hat aber trotzdem nur eine einzige Person. (Schwaber & Sutherland 2010, S. 7) Dies bestätigt Cohns Ansichten, dass es trotz Product-Owner-Team eine Einzelperson für die Entscheidungen braucht. Der Autor hat während eines 3-Monate dauernden Praktikums eigene Erfahrungen mit einem Product-Owner-Team gesammelt. Der Arbeitsablauf hat hierbei gut geklappt. Am Ende hat aber immer derselbe der beiden Product Owner die Entscheidung getroffen.

Product OwnerInnen sind für die Kommunikation mit den Stakeholdern zuständig. Sie erstellen gemeinsam mit diesen die Anforderungen für neue Features. Diese Anforderungen und Konzepte werden im Product Backlog (vgl. 2.5.1) festgehalten und andauernd von der Product OwnerIn wieder angepasst und richtig gestellt. Besonders wichtig ist dabei die Kommunikation mit Stakeholdern und Entwicklerteam. Die Regelmäßigkeit der Meetings mit Stakeholdern ist wichtig, um eventuell veränderte Anforderungen laufend klären zu können. (Pichler 2008, S. 10) Eine Product OwnerIn priorisiert das Product Backlog und stellt dies jedem zur Verfügung. Durch die richtige Priorisierung des Product Backlogs stellt die Product OwnerIn sicher, dass wichtige Arbeiten mit dem höchsten Business-Value zuerst erledigt werden. (Schwaber & Sutherland 2010, S. 7) Sie ist dafür verantwortlich, dass das Entwicklerteam durch seine Arbeit den höchsten Gewinn für das Unternehmen erzielen kann. Priorisiert eine Product OwnerIn falsch, werden die falschen Features entwickelt. Dies wiederum heißt, dass das Produkt nicht so gewinnbringend sein wird, wie es womöglich sein hätte können. Die Entscheidungen der Product OwnerIn müssen von allen im Unternehmen respektiert werden. Entwicklerteams sollen nur von der Product OwnerIn die Prioritäten vorgeschrieben bekommen. Hier darf sich keine andere Person einmischen. Die Product OwnerIn ist Frau/Herr über das Product Backlog und hat damit die Macht und die Verantwortung, die richtigen Aufgaben zu verteilen und die Priorität festzulegen. Eine Product OwnerIn muss daher transparent arbeiten und wenn möglich alle Meinungen innerhalb des Unternehmens auf die eine, richtige, für das Unternehmen am gewinnbringendste, eingrenzen. Dies macht diese Rolle, laut Schwaber und Sutherland, so anspruchsvoll, stressig, aber sogleich auch interessant. (Schwaber & Sutherland 2010, S. 7)

Laut Cohn muss ein gute Product OwnerIn eine Vision zum Produkt bereitstellen. Ist die Product OwnerIn selbst vom Erfolg und vom Produkt an sich überzeugt, hilft es ihr, diese Überzeugung an das restliche Scrum-Team zu übertragen und dadurch alle zu motivieren. Des Weiteren sagt Cohn, dass Product OwnerInnen Grenzen ziehen dürfen und auch müssen. Unter Grenzen wird

die Realität gemeint, in der eine Vision realisiert werden kann. Product OwnerInnen ist es in der Folge erlaubt, auch Aussagen zu tätigen, die beispielsweise Deadlines beinhalten. So wäre z.B. eine Aussage wie „Ich brauche dieses Feature bis Ende übernächsten Monats“ zulässig. Solche Aussagen müssen nur realistisch getätigt werden. (Cohn 2010, S. 125f)

Eine gute Product OwnerIn steht dem restlichen Scrum-Team jederzeit zur Verfügung, ist clever in ihrem Business, kommunikativ, entscheidungsfreudig und verfügt über die notwendige Vollmacht innerhalb des Unternehmens Entscheidungen auch treffen zu dürfen. (Cohn 2010, S. 130f)

### 2.2.3 Entwicklerteam

Die optimale Teamgröße des Entwicklerteams umfasst laut Schwaber und Sutherland sieben Mitglieder. Diese Anzahl kann auch um zwei Personen verringert bzw. vermehrt werden. Noch weniger Mitglieder könnte bedeuten, dass zu wenig Manpower vorhanden ist, um anstehende Aufgaben lösen zu können. Die Interaktion zwischen den einzelnen Entwicklerteam-Mitgliedern schwindet bei zu kleinen Teams. (Schwaber & Sutherland 2010, S. 8) (Stichwort: Teamgröße siehe: 2.3.1)

Das Entwicklerteam arbeitet die Tasks aus dem Sprint Backlog (siehe 2.5.2) ab und führt die tatsächlichen Programmierarbeiten aus. (Larman 2003, S. 115) Innerhalb eines Entwicklerteams gibt es verschiedene Rollen und Spezialgebiete der einzelnen Mitglieder. Es gibt beispielsweise SoftwareentwicklerInnen, TesterInnen, DatenbankdesignerInnen, ArchitektInnen und, je nach Produkt, noch weitere Rollen. Alle diese Individuen arbeiten gemeinschaftlich am Produkt und tauschen gegenseitiges Wissen aus (Pichler 2008, S. 13).

Besonders wichtig ist, dass trotz dieser unterschiedlichen Spezialisierungen jedes Entwicklerteam-Mitglied den Titel „Team-Member“ trägt. Es gibt keine anderen Rollenbezeichnungen oder irgendwelche „Unter-Teams“. Dies soll wieder die Gleichberechtigung aller verdeutlichen. (Larman 2003, S. 115) Diese Gleichberechtigung ist im Scrum-Prozess sehr wichtig, deshalb führen neben Larman auch Schwaber und Sutherland an, dass die Titelbezeichnung nur „Team-Member“ ist, und nicht weiter unterschieden wird.

Laut Schwaber und Sutherland sind die gemeinsamen Fähigkeiten aller Entwicklerteam-Mitglieder jedoch wichtiger wie die einzelnen Spezialgebiete. Unter gemeinsamen Fähigkeiten verstehen die beiden die Fähigkeit, eine Anforderung mit guten Arbeitsergebnissen erfüllen zu können und so ein nutzbares Produkt, innerhalb einer Iteration, umsetzen zu können. (Schwaber & Sutherland 2010, S. 8) Schwaber und Sutherland erläutern des Weiteren, dass jede/jeder erforderliche Aufgaben aus ihrem/seinem Bereich (z.B. Informatik) übernehmen können sollte, falls dies in gewissen Situationen notwendig wäre. Muss beispielsweise eine DatenbankdesignerIn, die normalerweise den ganzen Tag damit beschäftigt ist ER-Diagramme zu entwerfen, Programmieraufgaben übernehmen, sollte sie das auch tun. Team-Mitglieder, die sich „zu gut zum Coden“ sind und keine Programmieraufgaben übernehmen wollen, passen nicht gut in ein Entwicklerteam (Schwaber & Sutherland 2010, S. 8).

Das Entwicklerteam legt in Eigenregie fest, was innerhalb der nächsten Iteration, des nächsten Sprints (siehe 2.6.3), an Tasks aus dem Sprint Backlog (siehe 2.5.2) abgearbeitet wird. Das Entwicklerteam hat die Vollmacht, dies selber zu entscheiden. Verlangt eine Product OwnerIn beispielsweise zu viel in einem neuen Sprint, so muss das Entwicklerteam dagegenhalten und mitteilen, dass diese Anforderungen nicht erfüllt werden können. Die Scrum MasterIn soll dabei klarerweise das Entwicklerteam unterstützen. Das Entwicklerteam kann aber auch selber sagen, dass die Anforderungen einfach nicht zu erfüllen sind. (Pichler 2008, S. 13)

Ein weiteres Merkmal von Scrum ist, dass sich das Entwicklerteam selbst organisiert und die Team-Members autonom entscheiden wie der nächste Arbeitsschritt umgesetzt wird. Es braucht somit keine LeiterIn, ChefIn oder ManagerIn. Die Selbstorganisation ist vom Scrum-Prozess vorgegeben und diszipliniert. Das heißt (D.h.) durch diese Selbstorganisation und den disziplinierten Rahmen wird für das Entwicklerteam der Freiraum geschaffen um besser arbeiten zu können. Laut Pichler müssen Entwicklerteams diese Selbstorganisation zu Beginn erlernen. Dabei steht die Scrum

MasterIn unterstützend zur Seite. Hat sich das Entwicklerteam eingespielt und die benötigten Arbeitsabläufe koordiniert und festgelegt, kann jede EntwicklerIn in einem ordentlichen Rahmen arbeiten. (Pichler 2008, S. 15f)

Entwicklerteam-Mitglieder müssen teamfähig sein und über individuelles Fachwissen verfügen. Die Zusammensetzung eines Entwicklerteams soll durch Mitbestimmung der einzelnen potentiellen Mitglieder passieren. Dadurch kann jede EntwicklerIn selber beeinflussen, bei welchem Projekt sie mitarbeiten will. Dies fördert die Motivation und macht mehr Sinn als willkürlich Zuteilungen vorzunehmen. (Pichler 2008, S. 13) Können EntwicklerInnen selber mitentscheiden, bei welchen Projekten sie mitarbeiten wollen, so ist der generelle Wille zur Mitarbeit von Anfang an gegeben und die Arbeit wird besser.

#### 2.2.4 Sonstige Rollen

Der folgende Unterabschnitt beschäftigt sich mit zwei weiteren Rollen. Diese fallen nicht direkt in den Scrum-Prozess, können ihn aber entscheidend beeinflussen, weshalb sie hier genannt werden.

##### 1. „Chickens“

Unter dem Begriff „Chickens“ sind außenstehende Personen gemeint, die nicht Mitglieder des Scrum-Teams sind. Die Rolle eines „Chickens“ könnte beispielsweise eine neue MitarbeiterIn einnehmen, welche innerhalb eines laufenden Sprints (siehe 2.6.3) zum Scrum-Team hinzustößt. Diese Personen können beispielsweise beim Daily-Scrum-Meeting (siehe 2.6.4) teilnehmen, dürfen sich aber nicht zu Wort melden. „Chickens“ ist es erlaubt zuzuhören und dem Prozess beizuwohnen. Sie haben aber nicht die Erlaubnis zu sprechen, sich einzumischen, oder gar Anweisungen zu erteilen. „Chickens“ (außenstehende Personen) dürfen „Pigs“ (Scrum-Team-Mitgliedern) nicht vorschreiben, wie sie ihre Arbeit zu erledigen haben.

##### 2. „Stakeholder“

Die Rolle der Stakeholder umfasst die Personen, welche vom Endergebnis des Produktes direkt profitieren. Dies kann nun beispielsweise eine externe AuftraggeberIn sein, die für das Produkt bezahlt und anschließend das Produkt nutzen kann. In diesem Fall würde der Autor den Stakeholder als „tatsächliche KundIn“ bezeichnen, da ein Geldfluss entsteht. Entwickelt ein Scrum-Team eine neue Buchhaltungssoftware für die unternehmenseigene Abteilung „Finanz & Recht“, des gleichen Unternehmens, so können Mitarbeiter dieser Abteilung als Stakeholder eingestuft werden.

Zusammengefasst kann gesagt werden, dass jede/jeder, die/der vom Produkt schlussendlich profitiert, als Stakeholder eingestuft werden kann. Durch externe Auftragsarbeiten werden Gewinne erwirtschaftet. Durch interne Auftragsarbeiten wird für die dementsprechenden Abteilungen gearbeitet und durch diese Zuarbeit längerfristig ebenfalls ein Plus erzielt. Die intern benötigten Produkte können selber erstellt werden und müssen nicht extern eingekauft werden.

Stakeholder legen zusammen mit der Product OwnerIn die Anforderungen für neue Features fest und können dadurch den Arbeitsaufwand stark beeinflussen.

## 2.3 Scrum - Voraussetzungen

Scrum hat gewisse Voraussetzungen, um funktionieren zu können. Werden diese nicht oder teilweise nicht eingehalten, ist es schwierig, einen ordentlichen Prozessablauf und eine gute Arbeitsweise in Projekten einhalten zu können. Zu diesen Voraussetzungen zählen folgende:

### 2.3.1 Teamgröße

Ein wichtiger Punkt beim Einsatz von Scrum ist die Größe des Teams. Laut Larman sollte ein Team sieben Personen nicht überschreiten. (Larman 2003, S. 120) Anderer Ansicht ist hier Cohn. Er spricht von einer Teamgröße von fünf bis neun Personen. (Cohn 2010, S. 178) Pichler sagt, dass das auch Scrum-Teams mit weniger als fünf Mitglieder Scrum einsetzen kann. Hier muss allerdings vorausgesetzt werden, dass die einzelnen Mitglieder über die notwendigen Kenntnisse und Fähigkeiten verfügen müssen. Mit diesen Fähigkeiten und Kenntnissen müssen verlässlich auslieferbare Produktinkremente geschaffen werden können (Pichler 2008, S. 16).

Es kann festgehalten werden, dass in der Literatur von ähnlich großen Teams die Rede ist. Zusammengefasst scheint eine Teamgröße von fünf bis neun Personen als sinnvoll.

Warum ist eine Teamgröße von fünf bis neun Personen sinnvoll und kann nicht mit einem Entwicklerteam von vier oder 25 Personen gearbeitet werden? Hierfür gibt es mehrere Gründe und daraus resultierende Nachteile:

#### < 5 Mitglieder

1. Durch die Größe wird die Interaktion im Entwicklerteam stark eingeschränkt. Dadurch wird die mögliche Produktivitätssteigerung behindert und abgeschwächt (Schwaber & Sutherland 2010, S. 8).
2. Durch die Größe könnte Women/Man-Power fehlen und benötigte Fähigkeiten nicht vorhanden sein (Schwaber & Sutherland 2010, S. 8).

#### > 9 Mitglieder

1. Durch die Größe steigen der Koordinationsaufwand und die Komplexität stark an. Es wird für die Prozessverantwortlichen (Product OwnerIn, Scrum MasterIn) schwieriger beispielsweise Meetings zu organisieren (Schwaber & Sutherland 2010, S. 8).
2. Durch die Größe steigen außerdem die Kommunikationskosten stark an. Mitglieder müssen mit zu vielen anderen kommunizieren und dies stellt ein zusätzlicher Aufwand dar (Pichler 2008, S. 16).
3. Es besteht die Gefahr, dass sich einzelne Mitglieder zurücklehnen, bzw. in den Hintergrund verschwinden könnten und die anderen die Gesamtarbeit erledigen müssten. (Cohn 2010, S. 179)

### 2.3.2 Teammitglieder & Teamfähigkeit

Eine wichtige Voraussetzung für einen guten Prozessablauf mit Scrum ist das Vorhandensein der benötigten Ressourcen. D.h. es müssen für alle Rollen (vgl. 2.2) passend viele und fachlich kompetente Personen zur Verfügung stehen. Stehen jetzt beispielsweise für ein Projekt drei Product OwnerInnen, vier Scrum-MasterInnen und eine Person für das Entwicklerteam zur Verfügung, ist das sicherlich die falsche Mischung. Die genaue Anzahl von Personen der einzelnen Bereiche wurde bereits beim Kapitel 2.2 erläutert. Dieser Unterabschnitt soll nur kurz aufzeigen, dass die

einzelnen MitarbeiterInnen in richtiger Abstimmung ihrer Eigenschaften, Fähigkeiten und Erfahrungen in einem Team eingesetzt werden müssen. Besonders wichtig ist während des kompletten Scrum-Prozessablaufes die Teamfähigkeit jedes einzelnen Mitglieds eines Scrum-Teams. Im Scrum muss gemäß dem Motto „Eine für alle, alle für eine“ gearbeitet werden. Es gibt nicht „my work“ und „your work“, sondern es gibt nur „our work“. D.h. es steht nicht die Einzelleistung einer Person im Vordergrund, sondern die Gesamtleistung des kompletten Scrum-Teams. Dies muss von allen Teammitgliedern zum Start eines Projektes akzeptiert werden. Deshalb ist „Teamfähigkeit“ eine Grundvoraussetzung und Pflicht im Scrum-Prozessablauf. (Cohn 2010, S. 201) Cohn erwähnt in seinen Ausführungen zu diesem Themenbereich noch einen bereits beim Kapitel 2.1 angesprochenen wichtigen Punkt aus dem Agilen Manifest:

Individuals and interactions over processes and tools

Dieser Satz sagt aus, dass gute Software von großartigen Teams kommt. Aus individuellen Personen entsteht durch die Interaktion Großartiges.

### 2.3.3 Infrastruktur

Scrum hat auch infrastrukturelle Voraussetzungen. Teams, die im Scrum-Prozess arbeiten, sollten dies unbedingt in einem Raum tun. Daher ist, nach der Meinung des Autors, ein Großraumbüro für fünf bis neun Personen mit ausreichend Computerunterstützung für jede einzelne MitarbeiterIn ideal.

Laut Pichler wird durch die Zusammenarbeit auf engem Raum die Kommunikation zwischen den Einzelpersonen gefordert und gefördert. Dies treibt den Prozess weiter voran und es können bessere Ergebnisse erzielt werden. Pichler empfiehlt des Weiteren das Einrichten eines Teambereichs, sollte kein extra Raum zur Verfügung stehen. Sollte beispielsweise in einem Großraumbüro, mit anderen Abteilungen des gleichen Unternehmens, gearbeitet werden, sollten Trennwände aufgestellt werden und ein eigener Bereich für das Scrum-Team geschaffen werden. (Pichler 2008, S. 17) Dadurch kann die Kommunikation zwischen den einzelnen Scrum-Team-Mitgliedern am leichtesten gewährleistet werden.

Für Meetings wie Daily Scrum (siehe 2.6.4) werden täglich Tools benötigt, um arbeiten zu können. Darunter fällt zum Beispiel ein Chart für den Sprint-Burndown (siehe 2.5.4) an der Bürowand. Es werden noch weitere Charts und Pinnwände zum Festhalten der einzelnen Tasks benötigt. Sollten elektronische Tools, beispielsweise zum Festhalten der einzelnen Tasks des Sprint Backlogs (siehe 2.5.2) oder Visualisierungen des Release-Burndown Charts (siehe 2.5.3), eingesetzt werden, sollten die elektronisch erfassten Daten, laut Pichler, zusätzlich ausgedruckt und an der Wand aufgehängt werden. Durch das Sichtbarmachen der Informationen aus den elektronischen Tools, wird eine wichtige Transparenz geschaffen. Zusätzlich fördert dies das Verantwortungsbewusstsein der einzelnen Scrum-Team-Mitglieder (Pichler 2008, S. 19).

Büromaterialien wie Schreibutensilien oder Notizblöcke sind ebenfalls notwendig, um Scrum durchführen zu können. Dies kann der Autor aus eigenen Erfahrungen bestätigen. Die aufgezählten Materialien sind eigentlich nichts Außergewöhnliches, wenn sie aber nicht vorhanden sind, kann kein Scrum durchgeführt werden. Deshalb ist es besonders wichtig, dass darauf geachtet wird, dass auch diese „kleinen“ Voraussetzungen erfüllt werden. Sind beispielsweise alle infrastrukturellen Voraussetzungen erfüllt, es fehlt jedoch ein Whiteboard-Marker, kann nicht gearbeitet werden und der Prozess steht still. Ein still stehender Prozess ist im Scrum besonders schlecht.

### 2.3.4 Continuous Integration

Scrum ist ein fortlaufender, iterativer Prozess und benötigt eine kontinuierliche Integration von neuem Programm-Code in das bestehende System. Laut Larman sollte aus diesem Grund mindestens einmal pro Tag ein „Build“ (Dieser Begriff wird nachfolgend erklärt) des Programm-Codes

erstellt werden (Larman 2003, S. 120). Um einen „Build“ erstellen zu können, muss ein Continuous Integration Server vorhanden sein und mit diesem gearbeitet werden.

Um einen Continuous Integration Server verstehen zu können, muss zuerst der Begriff „Build“ erläutert werden.

**Build** Ein „Build“ ist eine Überprüfung der kompletten Software eines Projektes mithilfe eines Testdurchlaufs. Wird ein „Build“ gestartet, wird die Software je nach Programmiersprache eventuell zuerst kompiliert und damit vorbereitet. Im Anschluss wird die Software mit Tests überprüft. Die Tests kontrollieren dabei die Funktionalitäten der Software auf ihre Richtigkeit. Diese Tests müssen allerdings zuerst von ProgrammiererInnen geschrieben werden (Stichwort Test-Driven-Development (TDD), siehe 2.4.2). Laut Rödиг sind typische Aufgaben während eines solchen Build-Durchlaufs folgende:

- Konfiguration der Umgebung.
- Löschen alter Build-Artefakte.
- Kompilieren der Anwendung.
- Kompilieren von Testprogrammen.
- Durchführen der Testprogramme.
- Statische Code-Überprüfung durch Metrik Tools.
- Generierung von Dokumentation.

(Rödиг 2010, S. 4)

Inwieweit alle von Rödиг aufgezählten Aktionen tatsächlich durchgeführt werden, hängt von der Art der Software ab. Bei einer Ruby-on-Rails-Applikation, welche nur mit der Programmiersprache „Ruby“, und keinen zusätzlichen Bibliotheken, erstellt wird, fällt beispielsweise die Kompilierung weg.

Ein Continuous Integration Server sammelt alle eingetragenen Änderungen an Software-Komponenten. Nach einem Check-In von neuem Sourcecode wird ein neuer Build gestartet und damit ein neuer Durchlauf aller Tests losgelöst. (Keith 2010, S. 211) Ein Continuous Integration Server ist somit zentrale Sammel- und Überprüfungsstelle für neue Software-Komponenten. (Rödиг 2010, S. 6)

Laut Cohn ist seit den frühen 1990iger Jahren ein nächtlicher Build des Produktes als eine der besten Praktiken der Industrie bekannt. (Cohn 2010, S. 162) Doch warum sollte nur einmal pro Nacht ein Build durchgeführt werden? Im Scrum-Prozess checkt jede EntwicklerIn mehrmals täglich neuen Programm-Code in die Applikation ein. Würde dieser Code nur einmal pro Nacht oder gar nur einmal wöchentlich durch vorhandene Tests geprüft werden, könnten sich sehr leicht Fehler einschleichen und die programmierten Teilbereiche beschädigt werden. In der Wirtschaft wird oft ein Continuous Integration Server mit einem weiteren Tool verbunden. Dieses Tool gibt nach einem Build-Durchlauf den Status an die EntwicklerInnen weiter. Durch dieses Feedback weiß eine EntwicklerIn ob alles in Ordnung ist, oder ein Test fehlgeschlagen ist. In einigen Unternehmen wird zusätzlich über LED-Anzeigen oder Ampeln die Farbe von grün (alle Tests laufen erfolgreich durch) auf rot (Tests fehlgeschlagen) umgeschaltet und so an das komplette Entwicklerteam signalisiert, wie der Build verlaufen ist.

Abbildung 1 beinhaltet ein Beispiel für eine „Code-Ampel“. Diese wurde im Praktikumsunternehmen des Autors eingesetzt. Bei fehlerhaften Tests schaltet sie von grün auf rot um und die EntwicklerInnen können leichter erkennen, dass der Build fehlgeschlagen ist.

Der Continuous Integration Server sollte so konfiguriert sein, dass die Tests bei neuen check-ins automatisch anlaufen. Es gibt noch die Möglichkeit, dass die EntwicklerInnen diese Tests händisch



Abbildung 1: Beispiel zusätzliche Signalisierung des Build-Verlauf bei Continuous Integration Server

anstoßen. Cohn rät von dieser Variante jedoch ab, da EntwicklerInnen auch nur Menschen sind. Sie könnten sich womöglich denken, dass sie nur ein paar Codezeilen verändert haben, und ein neuer Build unnötig wäre. Doch gerade hierbei schleichen sich gerne Fehler ein. (Cohn 2010, S. 162) Larman unterstützt die Meinung von Cohn und sieht die laufende und dauernde Überprüfung über einen Continuous Integration Server als die bessere Variante. Mindestens einmal pro Tag muss ein Build durchgeführt werden. (Larman 2003, S. 120)

## 2.4 Scrum - Hilfreiches

Neben den bereits erklärten Voraussetzungen gibt es Hilfreiches zum Einsatz von Scrum. Die folglich erläuterten Aspekte sind kein Muss für Scrum, sie werden jedoch als hilfreich angeführt und oft angewendet. Der Autor spricht sie hier kurz an, damit sich LeserInnen ein besseres Bild machen können.

Des Weiteren sind die folglich beschriebenen Arten zu Programmieren bzw. zu Arbeiten im Rahmen von diversen Lehrveranstaltungen (z.B. Fachspezifisches Projektmanagement und Workflows Web, Web-Technologies 1 + 2, Ausgewählte Kapitel Web) Bestandteil des Unterrichts an der FHS. So erlangen StudentInnen bereits Wissen über hilfreiche Methoden zu Scrum. Dies ist eine wichtige Erkenntnis im Bezug auf die Beantwortung der Forschungsfrage dieser Bachelorarbeit. Der Autor hat im Rahmen von mehreren Projekten an der FHS, sowie im Praktikum, gute Erfahrungen mit den folgenden hilfreichen Tools gemacht:

### 2.4.1 Pair Programming

Beim Pair Programming (PP) wird zu zweit an einem Computer programmiert und gemeinsam gearbeitet. Eine Person tippt auf der Tastatur und die andere schaut dabei über die Schulter. Wird die eine ProgrammiererIn müde oder kann sich aus einem anderen Grund nicht mehr konzentrieren, wird die Tastatur übergeben. Dadurch wird die Konzentration und Zusammenarbeit gesteigert. Die Rollen wechseln mehrmals in der Stunde. Die ursprüngliche Idee dahinter war, dass eine EntwicklerIn sich selber gut kontrollieren kann, jedoch zwei EntwicklerInnen können sich doppelt kontrollieren. PP birgt einige Vorteile in sich. Es kann von der PartnerIn gelernt werden und es findet ein hoher Austausch an Wissen statt. EntwicklerInnen arbeiten disziplinierter und schauen sich gegenseitig auf die Finger. Schlechter Code wird gemeinsam beseitigt und die ProgrammiererInnen können konsequenter arbeiten. Das Endergebnis ist ein gemeinsam programmierter und erarbeiteter Code, welcher keinem vom Pair mehr gehört als dem/der anderem/anderen. (Martin 2003, S. 13)

Aus unternehmerischem Blickwinkel stellt sich natürlich die Frage, warum für die Arbeit einer Person zwei EntwicklerInnen angestellt werden sollen. Dies verdoppelt schließlich einseitig betrachtet

die Kosten des Personalaufwands. Laut Cockburn wird jedoch die Qualität des Programmcodes wesentlich erhöht und Fehler im Code verringert. Des Weiteren werden die technischen Fähigkeiten der ProgrammiererInnen verbessert und die interne Kommunikation gesteigert. Dies führt zu besseren Ergebnissen (Cockburn & Williams 2001). Williams bestätigt die Behauptungen von Cockburn. Er kommt zu dieser Bestätigung durch eine durchgeführte Universitätsstudie. Im Rahmen dieser Studie wurden Software-Engineering-StudentInnen beim Einsatz von PP beobachtet und die Ergebnisse analysiert. (Williams et al. 2000, S. 21ff) Die Ergebnisse spiegeln Cockburns Behauptungen wieder.

### 2.4.2 Test Driven Development

TDD heißt, dass vor dem eigentlichen Sourcecode Tests für diesen geschrieben werden. Diese Tests schlagen zuerst fehl, da noch kein Sourcecode dafür vorhanden ist. Anschließend wird der Sourcecode dafür geschrieben. Bei richtigem Sourcecode müssen die Tests erfolgreich durchlaufen. Der Sourcecode wird nochmals refactored und die Tests müssen immer noch durchlaufen. So ist gewährleistet, dass programmierte (Teil-)Bereiche zu jeder Zeit funktionieren. Ein großer Vorteil von TDD ist, dass der Code beliebig erweitert werden kann. Funktionieren nach der Erweiterung noch alle Testdurchläufe, können sich EntwicklerInnen sicher sein, dass durch neue Features keine alten Bereiche zu Schaden gekommen sind. Das zusätzliche Schreiben von Test-Code kostet Zeit. Durch die gewonnene Sicherheit wird jedoch langfristig, durch sofort vermiedene Fehler, wiederum Zeit gewonnen. (Martin 2003, S. 14) Scrum mit TDD bietet eine starke Sicherheit, dass der Programm-Code schnell und qualitativ hochwertig erstellt wird und auch einwandfrei funktioniert.

### 2.4.3 Story Driven Development

Story-Driven-Development (SDD) ist das Ausformulieren der anstehenden Features in kleinere, verfeinerte „Geschichten“. Diese „Geschichten“ werden auch User Stories genannt und von der Product OwnerIn ausformuliert (Wirdemann 2009, S. 57).

Für Product OwnerInnen bieten diese Stories eine ideale Basis, um Stakeholdern auf nicht-technischer Ebene zu präsentieren, wie deren Anforderungen verstanden wurden. Somit werden eventuelle Missverständnisse zwischen Stakeholdern und den EntwicklerInnen vermieden. Es kommt daher nicht zur Entwicklung und Zeitaufwendung für „unbrauchbaren“ Code, da im Vorhinein das Verständnis der Anforderungen überprüft wird.

Product OwnerInnen können durch den Einsatz von Stories den Product Backlog (vgl. 2.5.1) leichter priorisieren (Cohn 2004, S. 173). EntwicklerInnen können die Stories Schritt für Schritt abarbeiten. Mithilfe von SDD können größere Features in kleinere Teile gebrochen werden und eine iterative Entwicklung ist leichter möglich. Durch diese Aufteilung kann das Scrum-Team den Aufwand besser einschätzen und damit den Sprint leichter planen (Cohn 2004, S. 173).

Zur Abnahme der Features stehen der Product OwnerIn die Stories ebenfalls zur Verfügung. Die Story gibt den Rahmen des Features oder Teilfeatures vor. Dadurch kann die Product OwnerIn leichter feststellen, welche Teile des Sprints bereits fertig gestellt sind, und welche noch fehlen (Cohn 2004, S. 174).

## 2.5 Scrum - Artefakte

Scrum beinhaltet vier Artefakte. Zu diesen Artefakten zählen das „Product Backlog“, „Sprint Backlog“, der „Release-Burndown“ und „Sprint-Burndown“. Diese vier Artefakte sind notwendig, um im Scrum-Prozess arbeiten zu können. Sie stellen dabei Managementtools dar. (Schwaber & Sutherland 2010, S. 4) In den folgenden Unterkapiteln werden diese vier Punkte näher erläutert und vorgestellt.

### 2.5.1 Product Backlog

Das Product Backlog stellt eine Art „Box“ zum Erfassen und Managen von Anforderungen an das Produkt im Scrum-Prozess dar. Diese „Box“ kann nun beispielsweise eine Pinnwand, an der die einzelnen Anforderungen auf Zettel dokumentiert und angepinnt sind, sein. Es besteht auch die Möglichkeit das Product Backlog in elektronischer Form (z.B. Excel, Pivotaltracker, eigene Implementierungen, etc.) zu halten. Wichtig ist nur, dass es ein Product Backlog gibt. Das Product Backlog beinhaltet alle bekannten Anforderungen, welche für das Erreichen des Projektziels notwendig sind. Zu diesen Anforderungen zählen nicht nur Bestandteile neuer Features, sondern auch das Beseitigen von Bugs, oder beispielsweise das Einrichten eines neuen Continuous Integration Servers. Jedes Scrum-Projekt hat ein eigenes Product Backlog (Pichler 2008, S. 27ff). Das Product Backlog wird zwar zu Beginn eines Projektes angelegt, jedoch können die Anforderungen nicht anfangs festgelegt und dann der Reihe nach abgearbeitet werden. Durch den iterativen Prozess kann sich ein Product Backlog während des gesamten Projektverlaufs fortlaufend verändern. Es kommen neue Anforderungen hinzu, bestehende können sich verändern oder ganz wegfallen. Zu Beginn werden die Anforderungen nicht wie beim sequentiellen Prozessablauf (vgl. z.B. wie beim Wasserfallmodell) bis ins kleinste Detail geplant. Sie vervollständigen und erweitern sich inkrementell mit Fortlauf des Projektes. Anforderungen bzw. Einträge im Product Backlog sind priorisiert. Product OwnerInnen priorisieren ihre Product Backlogs nach Wichtigkeit, Nutzen, Kosten und eventuell weiteren unternehmensinternen Gegebenheiten und Bedingungen. Wichtig ist, dass höher priorisierte Anforderungen detaillierter beschrieben werden müssen, als niedriger eingestufte, da sie als erstes abgearbeitet werden. Aus diesem Grund haben Product OwnerInnen die Aufgabe, diese Anforderungen fein detailliert auszuarbeiten. Alle Einträge im Product Backlog sollten geschätzt sein. So können Product OwnerInnen besser bestimmen, welcher Eintrag am meisten wert ist. Für diese Schätzungen werden vom Scrum-Team Punkte vergeben. (Pichler 2008, S. 28) Wie viele Punkte minimal und maximal vergeben werden können, hängt von den Festlegungen des Scrum-Team ab. Dieses kann sich diese Werte selber festlegen, muss sie aber festhalten und kommunizieren. Cohn stimmt Pichler's Ausführungen mit der Zusammenfassung der Schlüsselattribute eines Product Backlogs im Begriff „DEEP“ zu. „DEEP“ sind die Anfangsbuchstaben folgender Begriffe:

Detailed Appropriately, Estimated, Emerged, Prioritized

(Cohn 2010, S. 253f)

Ein sich fortlaufend ändernder Product Backlog mit geschätzten und priorisierten Anforderungen, sowie die wichtigen Anforderungen passend detailliert vorbereitet, stellt somit einen Teil des Grundgerüsts für einen erfolgreichen Scrum-Prozess dar.

### 2.5.2 Sprint Backlog

Das Sprint Backlog ist eine Liste von Aufgaben (Tasks) aus dem Product Backlog. Mithilfe dieser Liste soll in einem Sprint (siehe 2.6.3) ein potenziell lieferbares Inkrement des Produktes entstehen. Die Erfüllung aller Tasks aus dem Sprint Backlog bildet das Ziel eines Sprints. Größere Tasks müssen in kleinere unterteilt werden, so dass ProgrammiererInnen leichter und unabhängiger entwickeln können. Das Sprint Backlog wird während eines Sprints durch das Scrum-Team

ständig verändert. Diese Veränderung ist auf die Entwicklung im Sprint zurückzuführen. ProgrammiererInnen können bei der Arbeit an einzelnen Tasks herausfinden, dass doch mehr, aber auch weniger, Arbeitsschritte als ursprünglich angenommen notwendig sind, um den Task abzuarbeiten. Einzelne dieser Arbeitsschritte können auch mehr oder weniger Zeit, als geplant, in Anspruch nehmen. Sollten neue Arbeitsschritte und Aufgaben auftauchen, werden sie dem Sprint Backlog durch das Scrum-Team hinzugefügt. Der Sprint Backlog darf nur durch Scrum-Team-Mitglieder geändert werden. Der Sprint Backlog ist ein Real-Time-Bild der Arbeit in einem laufenden Sprint. Das Scrum-Team will alle Aufgaben erfüllen, um den Sprint erfolgreich abschließen zu können. (Schwaber & Sutherland 2010, S. 18f)

### 2.5.3 Release-Burndown

Unter Release-Burndown versteht sich ein Graph, der die Summe des verbleibenden geschätzten Aufwands aller Anforderungen im Product Backlog zeitlich abbildet. Unter der zeitlichen Abbildung ist gemeint, dass alle geschätzten Punkte mit ihren Wertungen zusammengerechnet und auf bisherige Sprint-Ergebnisse (Sprint siehe 2.6.3) umgerechnet werden (Schwaber & Sutherland 2010, S. 4) bzw. (Schwaber & Sutherland 2010, S. 17f).

Um dies zu verdeutlichen ein Beispiel:

Im Product Backlog liegen anfangs Anforderungen mit einer Gesamtwertungspunktezahl von 80 Punkten. Ein Scrum-Team hat bereits drei Sprints an einem Projekt gearbeitet. Im ersten Sprint wurden 20, im zweiten 15 und im dritten 5 Wertungspunkte abgearbeitet. D.h. nach drei Sprints liegen noch Anforderungen mit 40 Wertungspunkten im Product Backlog. Aus diesen Zahlen ergibt sich eine durchschnittliche Abarbeitung von 13,333 Wertungspunkten pro Sprint. Das Produkt sollte ursprünglich mit durchschnittlich 20 Punkten pro Sprint bearbeitet werden. Durch die Verzögerungen schaut es nach dem dritten Sprint so aus, als ob sich das Releasedatum des letzten Abschnittes um zwei Sprints nach hinten verschiebt.

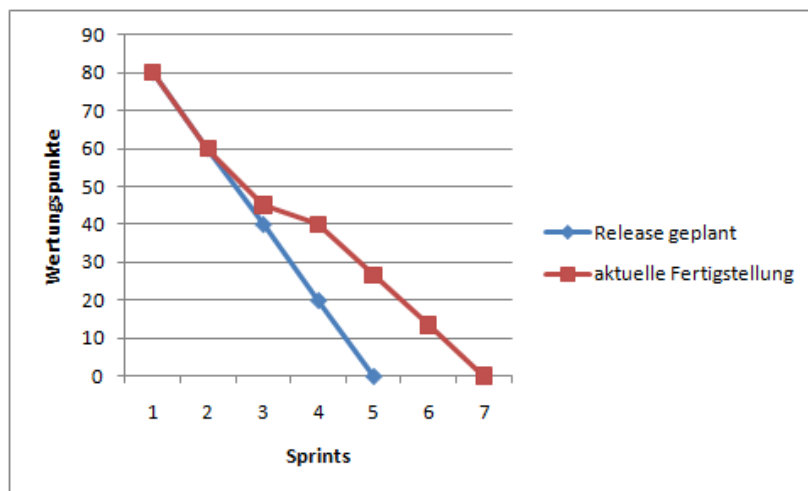


Abbildung 2: Beispiel Release-Burndown

Die Abbildung 2 beinhaltet den Release-Burndown Graphen des obigen Beispiels. Die x-Achse des Diagramms zeigt die Anzahl der Sprints. Die y-Achse beinhaltet die Wertungspunkte. Der Release-Burndown Graph wird beim nächsten Sprint-Ergebnis wieder angepasst. Er stellt eine Art Prognose für das Releasedatum dar.

### 2.5.4 Sprint-Burndown

Der Sprint-Burndown ist wie der Release-Burndown ein Graph. Dieser Graph wird täglich erneuert und angepasst. Er zeigt den aktuellen Sprint-Verlauf an und gibt aus, wie viele Wertungspunkte noch abzuarbeiten sind. Ein perfekter Sprint (siehe 2.6.3) hat einen Sprint-Burndown mit einer kontinuierlichen Kurve und am Ende des Sprints sind alle Wertungspunkte abgearbeitet. (Schwaber & Sutherland 2010, S. 19) Der Sprint-Burndown Graph zeigt dem Scrum-Team bereits während eines Sprint-Verlaufes an, ob es gegen Ende hin knapp werden könnte, alle Punkte abzuarbeiten. So können beispielsweise Personen mit der Rolle der Scrum MasterIn frühzeitig das Entwicklerteam in eine Richtung lenken und motivieren, schneller zu arbeiten, falls der Sprint-Burndown Graph bereits zur Mitte eines Sprints brenzlich aussieht.

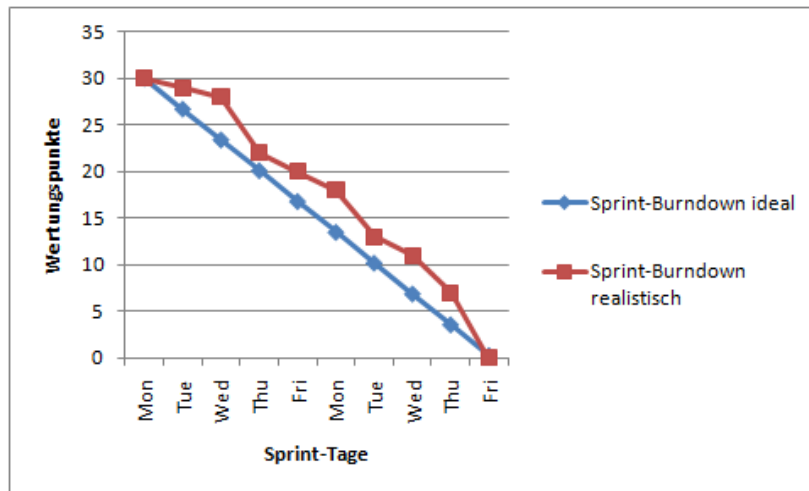


Abbildung 3: Beispiel Sprint-Burndown

Die Abbildung 3 beinhaltet Beispiele für einen Sprint-Burndown Graphen eines 2-wöchigen Sprints. Die x-Achse des Diagramms zeigt den jeweiligen Wochentag eines Sprints. Die y-Achse beinhaltet die Wertungspunkte, die an diesem Tag abgearbeitet wurden. Die kontinuierliche Kurve der Abbildung 3 ist ein Beispiel für einen perfekten Burndown, die zweite Kurve beinhaltet ein realistischeres Beispiel eines Burndown. Beide erreichen aber das Sprintziel.

Die Beispiele aus Abbildung 3 sind an ein Beispiel von Pichler angelehnt. Er zeigt in seinen Ausführungen ein ähnliches Ergebnis. Des Weiteren nennt er Gründe für den realistischen Burndown. Anfangs können Hindernisse auftreten, oder die Aufwandsschätzung durch das Scrum-Team ist falsch ausgefallen. Diese Blockaden legen sich mit der Zeit und der Sprint kann, im besten Fall, noch erfolgreich beendet werden. (Pichler 2008, S. 117f)

## 2.6 Scrum - Timeboxen

Scrum beinhaltet mehrere so genannte Timeboxen. Diese helfen dabei, Regelmäßigkeit herzustellen und einen strukturierten Ablauf sicherzustellen. Zu den Timeboxen gehört das Release-Planning, das Sprint-Planning, der Sprint, das Daily-Scrum Meeting, das Sprint Review und die Sprint Retrospektive. (Schwaber & Sutherland 2010, S. 9) Dieses Kapitel beschäftigt sich mit Erklärungen und Darstellungen zu den jeweiligen Timeboxen.

### 2.6.1 Release-Planning

Mithilfe des Release-Plannings wird ein Plan und Ziele aufgestellt bzw. definiert. Dieser Plan und die Ziele sollten vom Scrum-Team und der übrigen Unternehmensorganisation verstanden und kommuniziert werden können. Diese Ziele beinhalten die am höchsten priorisierten Einträge aus dem Product Backlog mit dem höchsten Business-Value. Des Weiteren beinhaltet der Release-Plan noch Hauptrisiken, welche auftreten können, und die restlichen Features und Funktionalitäten. Mit restlichen Features sind die gemeint, die zwar keinen hohen Business-Value haben, aber sehr wohl zum Produkt dazugehören. Mithilfe des im Release-Planning aufgestellten Plans können wahrscheinliche Release-Termine und Kosten berechnet werden. Die Unternehmensorganisation kann den Fortschritt während des Prozessablaufes beobachten und wenn nötig Sprint-weise Anpassungen vornehmen. (Schwaber & Sutherland 2010, S. 9f) Das Release-Planning ist laut Schwaber und Sutherland ein optionales Meeting. Wird das Meeting jedoch nicht abgehalten, so fehlen die Ergebnisse und damit Voraussetzungen für die nächsten Sprints. Dieses Fehlen wird als Hindernis eingestuft und gelangt als solches in das Product Backlog. Dies führt im Endeffekt dazu, dass das Meeting in einer anderen Form nachgeholt werden muss. Aus diesem Grund macht es Sinn dieses Meeting sofort abzuhalten. In der Scrum-Release-Planung werden, im Gegensatz zu traditionellen Release-Planungen, nur übergreifende Ziele und mögliche Endergebnisse festgelegt. Laut Schwaber und Sutherland benötigt diese Planung nur 15 - 20% der Zeit, welche für eine traditionelle Release-Planung notwendig wäre. Dieser Zeitgewinn wird durch Just-In-Time-Planungen während dem Sprint-Planning (siehe 2.6.2), Sprint Review (siehe 2.6.5) und Daily-Scrum (siehe 2.6.4) jedoch wieder eingebüßt. (Schwaber & Sutherland 2010, S. 9f) Just-In-Time-Planungen sind Planungen, welche in dem Moment getätigt werden, wann sie notwendig sind und nicht früher oder später.

Damit ein Release-Planning funktionieren kann, muss das Product Backlog für den jeweiligen Release priorisiert sein.

Larman bezeichnet das Release-Planning als Pre-game Planning und Staging. Er erweitert mit seinen Ausführungen die von Schwaber und Sutherland. Laut Larman wird in diesem Meeting ebenfalls ein Plan und Ziele festgelegt und etabliert. Die Anforderungen und Features werden aber von allen Stakeholdern eingebracht und im Product Backlog hinterlegt. So soll die Kommunikation der Plan-Etablierung zwischen Scrum-Team und restlicher Unternehmensorganisation noch auf externe Stakeholder erweitert werden, um die Anforderungen so gut wie möglich von Anfang an festhalten zu können. (Larman 2003, S 117)

### 2.6.2 Sprint-Planning

Im Sprint-Planning findet die Planung des Sprints (siehe 2.6.3) statt. Das Scrum-Team erstellt aus Anforderungen des Product Backlogs das Sprint Backlog. Bis zum Ende des Sprints muss das Sprint Backlog abgearbeitet werden und ein lieferbares Produktinkrement entstehen. (Pichler 2008, S. 93) Das Sprint-Planning findet zu Beginn jeden Sprints statt. Beim Sprint-Planning nimmt das gesamte Scrum-Team teil.

Laut Schwaber und Sutherland sollte das Sprint-Planning beispielsweise bei einem einmonatigen Sprint auf acht Stunden zeitlich begrenzt angesetzt werden. Arbeitet ein Scrum-Team im

2-Wochen-Rhythmus muss die Zeit für das Sprint-Planning dementsprechend angepasst werden, und würde nur vier Stunden betragen. Der Autor, der vorliegenden Arbeit, teilt die Meinung von Schwaber und Sutherland und stützt sich dabei auf eigene Erfahrungen. Er arbeitete in einem Unternehmen im 2-Wochen-Sprint-Rhythmus und das Sprint-Planning wurde auf 4 Stunden angesetzt. Im damaligen Fall war diese Zeit richtig bemessen.

Das Sprint-Planning besteht aus zwei Teilen. Diese werden auch Sprint-Planning 1 und Sprint-Planning 2 genannt. Im ersten Teil des Sprint-Plannings stellt die Product OwnerIn der Scrum MasterIn und dem Entwicklerteam alle Anforderungen vor, welche in diesem Sprint erfüllt werden sollen. Hier kann eine zusätzlich neue Priorisierung durch die Product OwnerIn stattfinden. Gründe für eine Neu-Priorisierung könnten beispielsweise neue Erkenntnisse betreffend Umsetzung oder Einwände vom Entwicklerteam sein.

Im zweiten Teil des Sprint-Plannings wird geplant, wie die Anforderungen umgesetzt werden. Hier wird seitens des Entwicklerteams alles Notwendige bis ins Detail durchgesprochen und abgeschätzt bzw. eventuell nochmal neu geschätzt. Die Product OwnerIn steht für eventuell anfallende Fragen, seitens des Entwicklerteams oder der Scrum MasterIn, zur Verfügung. (Schwaber & Sutherland 2010, S. 11f) Schwaber und Sutherland fassen die zwei Teile des Sprint-Plannings in den „What?“- und in den „How“-Part zusammen. Im ersten Teil geht es darum, was konkret erstellt werden muss. Im zweiten Teil wird geklärt, wie die Anforderungen umgesetzt werden können. Larman fasst die zwei Teile des Sprint-Plannings ähnlich zusammen. Im ersten Teil fügt er noch die Anwesenheit von Stakeholdern hinzu. Diese können den Ablauf beeinflussen und es kann dadurch ebenfalls zu Neu-Priorisierungen kommen. Im zweiten Teil hat er die gleichen Ausführungen wie Schwaber und Sutherland. Sollten jedoch zu viele Schätzungen von Anforderungen fehlen, oder falsch geschätzt worden sein, so erweitert Larman dieses Meeting um einen weiteren Planungszyklus. Sprich er würde eine Art „Sprint-Planning 3“ abhalten. (Larman 2003, S. 117)

Der Autor sieht eine Kombination der beiden eben beschriebenen Ansichten als ideal. Aus einer Art „Sprint-Planning 3“ könnte ein „Schätz-Meeting“, ein so genanntes „Estimations“, gemacht werden. Dies würde alleine dazu dienen, neue Anforderungen zu schätzen und alte Schätzungen gegebenenfalls anzupassen.

Beim Sprint-Planning können laut Pichler Fehler auftreten. Erstens könnte es beispielsweise passieren, dass die Scrum MasterIn praktisch die komplette Planung übernimmt, da das Entwicklerteam nicht gewohnt ist, Projektmanagementaufgaben selber zu erledigen. Besonders wichtig ist jedoch, dass das gesamte Scrum-Team, und somit auch das Entwicklerteam, sich selbst organisiert und von keiner Einzelperson direkt abhängig ist. Aus diesem Grund müssen alle Rolleninhaber eines Scrum-Teams am Sprint-Planning teilnehmen und dieses aktiv mitgestalten. Ein weiterer Fehler der auftreten kann, ist die Dominanz eines Teammitglieds. Übernimmt beispielsweise eine bestimmte Programmiererin immer das Wort und lässt keinen anderen mehr sprechen, könnte das komplette Meeting ins Stocken geraten. Deshalb muss die Scrum MasterIn das Sprint-Planning ordentlich moderieren und für Gleichberechtigung unter den Teilnehmern sorgen. Gleiches gilt für die Moderation im Bezug auf unendlich lange Diskussionen. Sie/Er muss dafür sorgen, dass bei Diskussionen das Produkt und die einzelnen Anforderungen im Mittelpunkt stehen und das Meeting nicht zu ausufernd wird und sehr lange, unnötige Diskussionen entstehen. Pichler erwähnt in seinen Ausführungen noch weitere Fehler. Product OwnerInnen müssen am Sprint-Planing teilnehmen. Sie sollen sich jedoch so gut wie möglich zurückhalten und nicht als ProjektleiterInnen agieren. Ansonsten könnte es zu starken Beeinflussungen kommen und das Entwicklerteam inklusive der Scrum MasterIn kann nicht mehr selbst organisierend Entscheidungen treffen. Im Gegenzug zum zu vielen Intervenieren einer Product OwnerIn ist es für die Planung selber sehr schlecht, wenn dieser/diese gar nicht anwesend ist. Ein Sprint-Planning muss immer gut vorbereitet sein. D.h. das Product Backlog muss von der Product OwnerIn soweit vorbereitet sein, dass die Anforderungen vollständig vorhanden sind und damit gearbeitet werden kann. Die einzelnen Anforderungen dürfen dabei nicht zu vage formuliert sein und vor allem nicht zu groß sein. Aus einer großen Anforderung sollten besser mehrere kleinere, detaillierte geformt werden. (Pichler 2008, S. 99ff)

Am Ende des Sprint-Plannings wird für den bevorstehenden Sprint ein so genanntes „Commitment“

seitens des Scrum-Teams abgegeben. Das „Commitment“ drückt aus, was in einem Sprint alles geschafft werden soll. Das „Commitment“ bildet damit ein definiertes Ziel für einen Sprint. (Larman 2003, S. 126) Das Scrum-Team kann nun selber und eigenständig entscheiden, wie dieses Ziel erreicht werden soll. Es muss nur erreicht werden. (Schwaber & Sutherland 2010, S. 11f)

### 2.6.3 Sprint

Ein Sprint ist ein Arbeitszyklus der iterativ abläuft. Ein Sprint wird laut Pichler deshalb auch als Iteration bezeichnet. (Pichler 2008, S. 81) Diese Iteration dauert gewöhnlich 30 Tage. (Larman 2003, S. 117) Die Dauer ist aber je nach Release-Zyklus auch auf zwei Wochen verkürzbar. Zu Beginn jeden Sprints findet das Sprint-Planing statt. Jeden Tag findet im Rahmen des Sprints das Daily-Scrum Meeting statt. Der Sprint wird mit dem Sprint Review und der Sprint Retrospektive abgeschlossen. (Pichler 2008, S. 81f) Zwischen zwei Sprints gibt es keine Pause, Sprints folgen immer direkt aufeinander. (Schwaber & Sutherland 2010, S. 10)

Während eines Sprints bleiben das Sprint-Ziel (Commitment) und die Zusammensetzung des Scrum-Teams unverändert. Sollte beispielsweise eine neue ProgrammiererIn eingestellt werden, so kann sie einem laufenden Sprint als „Chicken“ beiwohnen, um vom Prozessablauf des Scrum-Teams etwas mitzubekommen. Interventionen sind allerdings verboten.

Während des Programmierens in einem Sprints können hilfreiche Techniken wie PP, TDD oder SDD eingesetzt werden. Diese Techniken wurden bereits vorgestellt (siehe 2.4).

Am Ende eines Sprints muss ein fertiges Produktinkrement erstellt worden sein. Die Product OwnerIn muss die Möglichkeit haben, die erfüllten Anforderungen zu überprüfen und abzunehmen. Aus diesem Grund muss das fertig gestellte Produktinkrement am Ende jeden Sprints ein kompletter Teil des Produktes sein. (Pichler 2008, S. 83) Schwaber und Sutherland bezeichnen fertige Inkremente als „done“. Der Begriff „done“ muss für jeden des Scrum-Teams das gleiche bedeuten. (Schwaber & Sutherland 2010, S. 20) So kann beispielsweise ein Scrum-Team mit TDD arbeiten. Hier müssen wahrscheinlich alle erledigten Anforderungen gründlich getestet sein. Ein anderes Scrum-Team jedoch verzichtet auf TDD und setzt auf eine eigens entwickelte Methode. Hier wird die Voraussetzung, dass alles gründlich getestet sein muss, vielleicht nicht zutreffen. Aufgrund dieser unterschiedlichen Ansichtsweisen muss innerhalb eines Scrum-Teams der Begriff „done“ für alle dasselbe bedeuten.

Sprints können vorzeitig abgebrochen werden. Schwaber und Sutherland sehen die Autorität zum Abbruch eines Sprints einzig und alleine bei der Product OwnerIn (Schwaber & Sutherland 2010, S. 11). Pichler verteilt diese Autorität auch auf das restliche Scrum-Team (Pichler 2008, S. 116). Hat nun nur die Product OwnerIn die Autorität dazu, einen Sprint vorzeitig zu beenden, kann das restliche Scrum-Team aber auch externe Stakeholder diese Entscheidung trotzdem maßgeblich beeinflussen. Sprints können aufgrund von groben Hindernissen, ändernden Kundenanforderungen oder nicht mehr aktuellen und somit überflüssigen Anforderungen abgebrochen werden. Die aufgezählten Gründe können beispielsweise durch Änderungen der Marktposition des Unternehmens oder technologischen Änderungen auftreten. Laut Schwaber und Sutherland ist aufgrund der kurzen Dauer eines einzelnen Sprints ein Abbruch jedoch selten sinnvoll und kommt nicht oft vor. (Schwaber & Sutherland 2010, S. 11) Diese Ansichten teilt der Autor dieser Bachelorarbeit. Nach seinen Erfahrungen im Rahmen des Berufspraktikums, während des 5. Semesters des Studiengangs MMT, und den Erzählungen der Scrum-erfahrenen MitarbeiterInnen im Praktikumsunternehmen, sind Sprint-Abbrüche kaum relevant. Natürlich kann es gute und schlechte Sprints geben, aber abgebrochen wird kaum einmal ein Sprint.

### 2.6.4 Daily-Scrum

Schwaber und Sutherland definieren Daily Scrum als ein tägliches 15-minütiges Scrum-Team-Meeting das stets zur selben Zeit und am gleichen Ort stattfindet. (Schwaber & Sutherland 2010, S. 15) Wirdemann nennt in seinen Ausführungen eine fast gleiche Definition vom Daily-Scrum. (Wirdemann 2009, S. 31) Er lässt den Ort des Meetings unbestimmt und gibt nicht vor, dass es immer der gleiche Treffpunkt für das Meeting sein muss. Larman bezeichnet das Daily-Scrum Meeting nur als „Scrum Meeting“ und definiert den Zeitraum des Meetings durchschnittlich auf 15 bis 20 Minuten. (Larman 2003, S. 120f)

Am Daily-Scrum nimmt das komplette Scrum-Team teil und jeder hat Rederecht. Außenstehende dürfen zuhören, sich aber nicht einbringen. Beim Daily-Scrum steht das Scrum-Team in einem Kreis zusammen und jede/jeder einzelne beantwortet die drei folgenden Fragen:

1. Welche Aktivitäten habe ich seit der letzten Daily-Scrum abgeschlossen?
2. Woran plane ich bis zur nächsten Daily-Scrum zu arbeiten?
3. Werde ich in irgendeiner Form an der Ausführung einer Aktivität behindert?

(Pichler 2008, S. 105)

Larman hat diese drei Fragen noch um zwei weitere erweitert:

1. Any tasks add to the Sprint Backlog? (missed, tasks, not new requirements)
2. Have you learned or decided anything new, of relevance to some of the team members? (technical, requirements, ...)

Diese zwei zusätzlichen Fragen wurden laut Larman von Schwaber und Sutherland geprüft und als sinnvoll anerkannt. (Larman 2003, S. 121)

Die Antworten der einzelnen Scrum-Team-Mitglieder müssen aufgrund der kurzen Dauer des Meetings kurz und bündig ausfallen. Die Scrum MasterIn hat die Aufgabe das Daily-Scrum Meeting zu leiten und zu moderieren. Jede SprecherIn soll die gleiche Redezeit bekommen. Probleme sollen angesprochen werden. Nach Problemlösungen wird allerdings außerhalb des Daily-Scrum Meetings gesucht. (Pichler 2008, S. 105)

Laut Wirdemann implementiert das Daily-Scrum Meeting zusammen mit dem Sprint Feedback-Schleifen für das Scrum-Team. D.h. durch die täglichen Meetings innerhalb eines Sprints wird es ermöglicht, die komplette Arbeit des Scrum-Teams für alle transparent und sichtbar zu schalten. Jede ProgrammiererIn bekommt mit, an was die anderen gerade arbeiten, welche Probleme aufgetreten sind, welche Hindernisse im Weg stehen. Durch diese Transparenz findet ein höherer Austausch an Wissen und Erfahrungen statt und EntwicklerInnen können sich gegenseitig leichter helfen. (Wirdemann 2009, S. 31)

Um das Daily-Scrum Meeting leichter durchführen zu können, sind Vorbereitungsarbeiten von Nöten. Vor Beginn des Meetings sollte das Sprint Backlog und der Sprint Burndown aktualisiert werden. So kann während des Meetings leichter der aktuelle Sprint-Verlauf betrachtet, analysiert und besprochen werden. (Pichler 2008, S. 105)

Idealer Ort für das Daily-Scrum Meeting ist laut Larman der Ort, an dem sich alle Tasks und der Sprint-Burndown Graph befinden. (Larman 2003, S. 121) Dies kann beispielsweise in einem Raum vor einem Computer (bei elektronisch geführtem Sprint Backlog) oder vor einer Pinnwand (bei analog geführtem Sprint Backlog) sein.

### 2.6.5 Sprint Review

Das Sprint Review ist ein am Ende jeden Sprints abgehaltenes, informelles Meeting. Das Sprint Review dauert je nach Sprint-Länge zwischen ein und vier Stunden. Für einen einmonatigen Sprint beschränken Schwaber und Sutherland die maximale Länge des Sprint Reviews auf vier Stunden. (Schwaber & Sutherland 2010, S. 13f) Während des Sprint Reviews stellt das Entwicklerteam unter der Leitung der Scrum MasterIn die erledigten Anforderungen vor. Ziel ist es die Arbeitsergebnisse transparent zu machen und zu begutachten. Die Product OwnerIn soll die Möglichkeit haben, durch die Präsentation der Ergebnisse diese auch zu testen, gegebenenfalls abzunehmen und Maßnahmen zur Verbesserung für zukünftige Sprints einzubringen. Alle geleisteten Anforderungen sollen als vollständiges Produktinkrement vorliegen. (Pichler 2008, S. 107)

Am Sprint Review nehmen neben dem kompletten Scrum-Team auch Stakeholder und andere Interessierte teil. Das Team demonstriert die geleistete Arbeit und steht für Fragen zum Produkt zur Verfügung. Wichtig ist, dass die Demonstration des neuen Produktinkrement ohne eine große Vorbereitung und ohne PowerPoint-Präsentation von statten geht. (Wirdemann 2009, S.31) Diese Ansichten teilt Larman in seinen Ausführungen. Die Vorführung hat an einem, dem Live-System möglichst ähnlichen, Test-System zu erfolgen. Es wird der letzte erfolgreiche Build präsentiert und kein älterer Zwischenstand. Es wird somit laufende Software gezeigt und keine theoretischen PowerPoint-Folien. (Larman 2003, S. 120)

Besonders wichtig im Sprint Review ist, dass die Product OwnerIn sich aktiv am Meeting beteiligt. Nicht zufriedenstellende Teilergebnisse müssen ehrlich kommuniziert werden und klar festgehalten werden. Es nützt niemandem etwas, wenn alles schön geredet wird, und am Ende das Ergebnis schlicht weg nicht mit den Anforderungen übereinstimmt. Bei der Präsentation sollten keine Einzelleistungen gelobt werden und einzelne Scrum-Team-Mitglieder hervorgehoben werden. Das Scrum-Team funktioniert nur als Einheit und kann so am besten arbeiten. Ist beispielsweise in einem Sprint etwas besonders Tolles entwickelt worden und sogar die Geschäftsführerin, die normalerweise nie am Sprint Review teilnimmt, nimmt ausnahmsweise daran teil, dann sollte die Präsentation trotzdem im Rahmen des Üblichen ablaufen. Eine extravagant inszenierte Show durch das Entwicklerteam bringt niemandem etwas, da das Sprint Review nicht diesem Zweck dient. Deshalb sollte das Meeting ganz normal ablaufen, da ein Bild des Ergebnisses und nicht die Präsentation im Mittelpunkt steht. (Pichler 2008, S. 110f)

Durch das Feedback von Stakeholdern und KundInnen können Schlüsse für die nächsten Sprints gezogen werden. Diese Schlüsse geben die zukünftig einzuschlagenden Richtungen vor. Laut Wirdemann ist das Sprint Review die Zeit für Kurskorrekturen. (Wirdemann 2009, S.31)

### 2.6.6 Sprint Retrospektive

Die Sprint Retrospektive ist ein Meeting, das direkt nach dem Sprint Review stattfindet. Schwaber und Sutherland setzen für die Sprint Retrospektive eine Dauer von drei Stunden, bei einem einmonatigen Sprint, an. Die Zeit muss wie beim Sprint Review bei kürzerer Sprint-Länge dementsprechend angepasst werden. (Schwaber & Sutherland 2010, S. 14)

Die Sprint Retrospektive kann auch einmal kürzer als die veranschlagte Zeit ausfallen. Dies wäre beispielsweise der Fall, wenn es seitens des Scrum-Teams wenig zu berichten und dadurch zu besprechen gibt. Allerdings sollte die Sprint Retrospektive nicht vernachlässigt werden. Sie ist eines der Werkzeuge zur ständigen Verbesserung des Scrum-Prozesses.

An der Sprint Retrospektive nimmt das komplette Scrum-Team teil. Die Scrum MasterIn hat die Aufgabe, das Meeting vorzubereiten und schlüpft während des Meetings in die Rolle der ModeratorIn. Pichler besagt, dass zusätzliche Personen außerhalb des Scrum-Teams, auf Wunsch des Scrum-Teams, auch an der Sprint Retrospektive teilnehmen können. Dadurch lassen sich beispielsweise Hindernisse durch äußere Einflüsse besprechen und leichter beheben, da außenstehende Personen die Sicht des Scrum-Teams durch das Meeting besser verstehen können. (Pichler 2008, S. 112)

Ziel der Sprint Retrospektive ist es, den Scrum-Prozess innerhalb des Scrum-Teams zu verbessern. Dabei soll die Zusammenarbeit der einzelnen Mitglieder im Vordergrund stehen. Des Weiteren dient sie zur Analyse des letzten Sprints. Hierbei stehen die Beziehung zwischen den Scrum-Team-Mitgliedern und der Prozessablauf im Mittelpunkt. Die zwei folgenden Fragen werden beantwortet: Was hat im letzten Sprint gut geklappt? Wie kann der Prozess verbessert werden?

In der Sprint Retrospektive soll laut Derby und Larsen die ModeratorIn, sprich die Scrum MasterIn, zu Beginn des Meetings alle Mitglieder zuerst willkommen heißen und ihnen das Gefühl geben, dass sie von der Arbeit vorerst abschalten können und ganz frei sprechen können. Es soll eingangs nochmals der Sinn der Retrospektive erläutert werden und jedes Mitglied soll kurz äußern was er/sie sich von der Retrospektive erhofft. (Derby & Larsen 2006, S. 5) Anschließend werden nach den Ausführungen von Derby und Larsen Daten gesammelt. Was hat den einzelnen Mitgliedern im Sprint gut gefallen? Was war weniger gut? Welche Probleme sind entstanden? Welche Probleme sind gelöst worden? Wie wurden diese Probleme gelöst? Die Datensammlungen der einzelnen Mitglieder beinhalten Antworten zu diesen Fragen. Aus den gesammelten Daten wird dann versucht, Erkenntnisse abzuleiten wie tatsächlich vorgegangen wurde und wie dieser Vorgang in Zukunft verbessert werden kann. Diese werden in Gesprächen aufgearbeitet und festgehalten. (Derby & Larsen 2006, S. 11) Das Scrum-Team fällt die Entscheidung was nun konkret gemacht wird, um den Prozess zu verbessern. Zum Schluss der Sprint Retrospektive muss die Entscheidung wie Verbesserungen durchgeführt werden können, festgehalten werden. Derby und Larsen sagen, dass das Festhalten auf großen, und für jeden zu sehenden Charts, Postern oder Ähnlichem erfolgen soll. Dadurch ist das Ergebnis räumlich präsent und die Scrum-Team-Mitglieder haben diese Ergebnisse täglich vor Augen. (Derby & Larsen 2006, S. 13)

## 2.7 Scrum - Zyklus

Mittlerweile sind alle für den Scrum Zyklus erforderlichen Aspekte erläutert worden. Der Scrum Zyklus setzt sich aus den verschiedenen Rollen eines Scrum-Teams, den Timeboxen und den Artefakten zusammen. (Schwaber & Sutherland 2010, S. 4) Verschiedene Voraussetzungen (vgl. 2.3) müssen gegeben sein. Es können hilfreiche Tools (vgl. 2.4) eingesetzt werden.

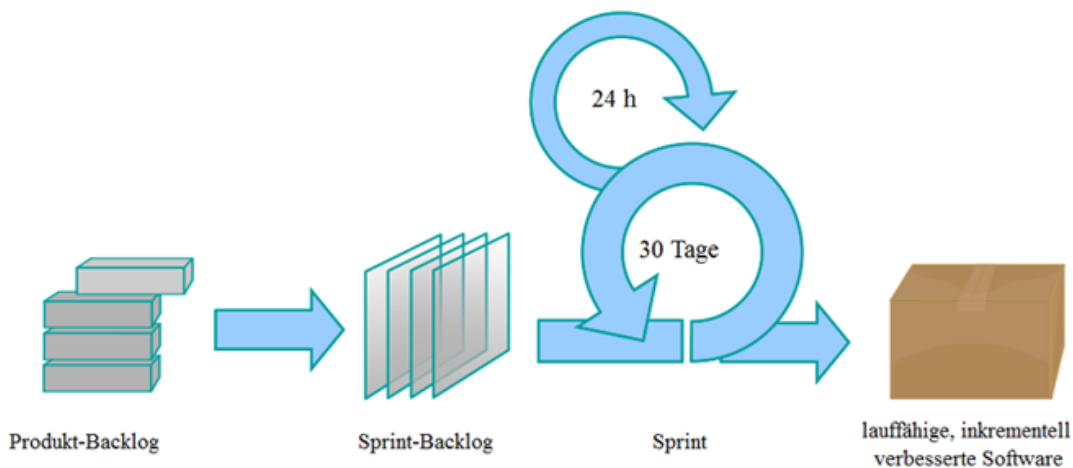


Abbildung 4: Beispiel Scrum Zyklus

Quelle: [http://de.wikipedia.org/wiki/Datei:Scrum\\_process-de.svg](http://de.wikipedia.org/wiki/Datei:Scrum_process-de.svg)

Abbildung 4 beinhaltet ein Beispiel eines Scrum Zyklus. In diesem Beispiel wird der Scrum-Prozess kurz zusammengefasst. Anforderungen werden aus dem Product Backlog ins Sprint Backlog überführt. Der Sprint dauert 30 Tage und die einzelnen Anforderungen aus dem Sprint Backlog werden abgearbeitet. Zu Beginn des Sprints wird das Sprint Planning abgehalten, täglich findet das

Daily-Scrum Meeting statt. Abgeschlossen wird der Sprint mit dem Sprint Review und der Sprint Retrospektive. Zum Schluss muss eine lauffähige, inkrementell verbesserte Software als Ergebnis herauskommen. Am Ende eines Sprints wartet bereits der nächste und der Prozessablauf startet wieder von Neuem.

### 3 Scrum an der Fachhochschule Salzburg

Dieses Kapitel beinhaltet eine analytische Diskussion über den Einsatz von Scrum in Projekten an der FHS. In der vorliegenden Bachelorarbeit werden nur Projekte der Studiengänge MMA und MMT behandelt. Projekte von anderen Studiengängen, wie beispielsweise Informationstechnik & System-Management (ITS) oder Design- & Produktmanagement (DPM), werden nicht mit einbezogen. Die Rahmenbedingungen der FHS spielen dabei eine große Rolle. Aus diesem Grund setzt der Autor auf Leitfäden zu Projekten im Rahmen der Bachelorstudiengänge von MMA und MMT. Zu diesen Projekten zählen die Qualifikationsprojekte. Seitens von MMA wären dies folgende:

- Qualifikationsprojekt 1 „Fachbereichswahl“ (MMA 2011)
- Qualifikationsprojekt 2 „Vertiefung im Fachbereich – Vorbereitung Berufspraktikum“ (MMA 2008)
- Qualifikationsprojekt 3 „Leitfaden QPT3 Bachelor MultiMediaArt Bachelor MultiMedia-Technology Jahrgang B-2008“ (MMA & MMT 2011*d*)

Seitens von MMT wären dies folgende:

- Qualifikationsprojekt 1 „Basisqualifikation“ (MMT 2011*a*)
- Qualifikationsprojekt 2a „Fachspezifische Qualifikation“ (MMT 2010)
- Qualifikationsprojekt 2b „Multimediale Qualifikation QPT 2b “ (MMT 2011*b*)
- Qualifikationsprojekt 3 „Leitfaden QPT3 Bachelor MultiMediaArt Bachelor MultiMedia-Technology Jahrgang B-2008“ (MMA & MMT 2011*d*)

Bei den Leitfäden ist zu beachten, dass der Autor die aktuellsten zum jetzigen Stand als Referenz genommen hat. (Stand 03.05.2011) In den unterschiedlichen Jahrgängen können diese Leitfäden jedoch Unterschiede aufweisen. Die Leitfäden bilden die Rahmenbedingungen, für in Frage kommende Projekte. Für die Masterstudiengänge von MMA und MMT existieren keine Leitfäden. Mit Einführung des MMT-Masterstudienganges sollte für beide Studiengänge ein gemeinsamer Leitfaden entstehen (vgl. B, Zeile 591).

Des Weiteren setzt der Autor in der analytischen Diskussion auf die Aussagen von zwei ExpertInnen, mit denen für diese Bachelorarbeit Interviews geführt wurden. Das erste Interview wurde mit Frau Mag. Jeanny Gucher, Leiterin des Fachbereichs Management & Steuerung an den Studiengängen MMA und MMT, geführt. Das Zweite wurde mit Herrn Mag. (FH) Stefan Randelshofer, der Wissenschaftlicher Mitarbeiter und Lehrender an der FHS, sowie zertifizierter Scrum Master ist, abgehalten. Diese zwei Personen werden aufgrund ihrer Erfahrungen und Tätigkeiten im Bezug auf Scrum, an der FHS, sowie in der Wirtschaft, als ExpertInnen angesehen. Am Wichtigsten für die Einstufung als ExpertInnen im Bezug auf die Beantwortung der Forschungsfrage ist, nach Meinung des Autors der vorliegenden Bachelorarbeit, dass diese beiden Personen das Umfeld der FHS kennen und deshalb gut einschätzen können.

Das erste Unterkapitel dieses Kapitels befasst sich mit den Gegebenheiten an der FHS. Welche Projekte werden an der FHS erstellt? In welchem zeitlichen Rahmen findet die Ausarbeitung dieser Projekte statt? Was ist an der FHS besonders zu beachten?

In den nachfolgenden Unterkapiteln wird Scrum gemäß der Literatur auf die Rahmenbedingungen der FHS herunter gebrochen. Ist Scrum gemäß Erkenntnissen aus der Literatur eins-zu-eins auf die FHS übertragbar? Welche Abänderungen müssen eventuell gemacht werden? Wie kann am besten vorgegangen werden, um Scrum sinnvoll einzusetzen? Ist ein sinnvoller Einsatz überhaupt möglich? Welche Personen müssen welche Verantwortlichkeiten übernehmen?

### 3.1 Gegebenheiten an der FHS

Dieses Kapitel beschäftigt sich mit spezifischen Gegebenheiten an der FHS.

#### 3.1.1 Projekttypen durch Ausbildungsschwerpunkte

In den Studiengängen MMA und MMT werden viele unterschiedliche Typen von Projekten realisiert. Diese hängen von den Ausbildungsschwerpunkten von MMA und MMT ab. Es existieren beispielsweise Projekte in folgenden Themenbereiche:

- Audio (MMA)
- Film (MMA)
- Mediendesign (MMA)
- 3D-Animation (MMA)
- Web & Communities (MMT)
- Augmented Reality & Game (MMT)

Diese Themenbereiche werden auch des Öfteren gemischt und es werden gemeinsam Projekte erstellt. Beispielsweise wird seitens der Studiengangleitung empfohlen im Rahmen des QPT2b von MMT mit MMA-Studierenden zusammen zu arbeiten. (MMT 2011b, S. 3). In anderen Projekten, wie dem QPT3, ist die Zusammenarbeit mit verschiedenen Fachbereichen Pflicht. (MMA & MMT 2011d, S. 3)

Durch diese vorgegebenen Rahmenbedingungen kann festgehalten werden, dass durch die Kombination von verschiedenen Themenbereichen bzw. Fachbereichen eine hohe Anzahl an verschiedenen Projekttypen entsteht.

Für den Einsatz von Scrum sind, laut Gucher und Randelshofer, nur Projekte mit einem hohen Software-Anteil geeignet (vgl. A.1, Zeile 26ff bzw. A.2, Zeile 411ff). Diese Meinung teilt Pichler in seinen Ausführungen. In seiner Definition von Scrum bezieht er Projekte ohne Software-Anteil nicht mit ein. Für ihn ist Scrum ein agiles Managementframework zur Entwicklung von Software. (Pichler 2008, S. 1) Daraus schließt der Autor, dass Projekte mit einem geringen bis gar keinem Softwareanteil, wie beispielsweise Film-Projekte, nicht für den Einsatz von Scrum geeignet sind.

#### 3.1.2 Projekttypen nach Abläufen

An der FHS werden verschiedene Projekte mit verschiedenen Abläufen erarbeitet. Dieser Abschnitt spricht die Projekttypen nach ihren Abläufen kurz an, da sie später bei der Rollenverteilung und Stakeholder-Zuteilung für die entsprechenden Unterschiede sorgen.

- klassisches Studentenprojekt:

StudentInnen haben eine interessante Projektidee inklusive Konzept und setzen die Idee schließlich um (vgl. B, Zeile 561). Am Ende präsentieren sie das Ergebnis, stellen das Projekt der Öffentlichkeit zur Verfügung und schreiben in ihrem eigenen Blog einen kurzen Bericht darüber.

D.h. StudentInnen entwickeln aus eigenen Ideen eigenständig Projekte im Rahmen des Studiums.

- FHS internes Projekt:

Die Studiengänge der FHS benötigen immer wieder eigene Plattformen oder Tools um sich der Öffentlichkeit gegenüber besser präsentieren zu können oder intern praktische Anwendungen zur Verfügung zu haben. Aus diesem Grund werden teilweise Projekte intern von MMA oder MMT Lehrenden vergeben. Die Projektinitiative geht bei diesen Projekten von den dementsprechenden Lehrenden aus. Die Erarbeitung geschieht durch StudentInnen (vgl. B, Zeile 563).

- FHS Forschungsprojekt:

Aus den Forschungsprojekten der Studiengänge MMA und MMT ergeben sich auch Projektaufträge. Die ProjektauftraggeberInnen sind in diesen Fällen die zuständigen Forschungsbeauftragten seitens der Studiengänge. Die Projektinitiative geht von den Forschungsbeauftragten aus und wird durch diese an StudentInnen übertragen. StudentInnen erarbeiten das Projekt und werden durch die Forschungsbeauftragten betreut (vgl. B, Zeile 565f).

- Auftragsprojekt:

**Allgemein** Ein Projekt, welches von externen AuftraggeberInnen an die Studiengänge und von den Studiengängen an die StudentInnen übergeben wird, wird als Auftragsprojekt bezeichnet. Solche Auftragsprojekte müssen inhaltlich in den Lehrplan der StudentInnen passen (vgl. B, Zeile 568f). D.h. das Auftragsprojekt muss, einem dem Lehrplan entsprechenden, qualitativen Anspruch haben. Des Weiteren müssen verwendete Werkzeuge, zeitliche Rahmenbedingungen und der Projektumfang mit den dementsprechenden Inhalten des Lehrplans übereinstimmen.

**Ablauf** Bei kommerziellen Projekten gibt es meist folgenden Ablauf: StudentInnen erstellen im Unterricht ein Konzept oder einen Prototyp. Diese Arbeit wird von der betreuenden Lehrperson beurteilt (vgl. B, Zeile 577f). Die Konzepte bzw. Prototypen werden entweder zu üblichen Marktpreisen bezahlt, oder im Rahmen eines Wettbewerbes mit Preisgeldern belohnt. Im Anschluss können die StudentInnen für die Umsetzung direkt von der externen Firma angestellt oder beauftragt werden. Diese Anstellung kann beispielsweise im Rahmen eines Praktikums, eines Sommer- oder Nebenjobs erfolgen. Wichtig ist, dass die StudentInnen dafür ganz normal und gemäß der Wirtschaft üblichen Preisen entlohnt werden (vgl. B, Zeile 580f).

### 3.1.3 Projektumfang

Im Rahmen der Ausbildung an der FHS führen Studierende Projekte mit unterschiedlich großem Umfang durch. Manche Projekte können in einem Zeitraum von einer bis zwei Wochen abgeschlossen werden, für andere wiederum benötigen die Studierenden mehrere Monate zur Fertigstellung. Der Autor sieht nach einer Analyse der Leitfäden der verschiedenen Qualifikationsprojekte folgende Projekte als relevant, um vom Umfang her, mit Scrum bearbeitet zu werden:

1. QPT2-Projekte

- QPT2 MMA

Das QPT2 von MMA verläuft für fast alle Fachbereiche über ein komplettes Semester. Ausgenommen ist hier der Fachbereich „Mediendesign“. In diesem Fachbereich ist die Abgabe früher und das Projekt muss innerhalb von zwei Monaten umgesetzt werden (MMA 2008, S. 2). Im Durchschnitt ergibt sich ein zeitlicher Umfang von drei Monaten für die Umsetzung (Semesterbeginn bis Ende April (Mediendesign), Semesterbeginn bis Mitte Juni (Audio, 3D-Animation, Film; Semesterbeginn ist mit Ende Februar bzw. Anfang März festgelegt).

- QPT2b MMT

Das QPT2b von MMT findet innerhalb eines zeitlichen Rahmens von vier Monaten statt. Hier wird von Ende Februar bzw. Anfang März bis Anfang Juni am Projekt gearbeitet (MMT 2011*b*, S. 3).

Im QPT2b-Leitfaden von MMT wird den Studierenden empfohlen, mit dem Studiengang MMA zusammenzuarbeiten. (MMT 2011*b*, S. 3) Der QPT2-Leitfaden von MMA besagt, dass das Projekt in Einzel- oder Gruppenarbeiten durchgeführt werden kann. (MMA 2008, S. 1) Daraus schließt der Autor, dass in jedem Fall eine generelle Zusammenarbeit einen Sinn ergibt.

Durch eine Zusammenarbeit zwischen MMA und MMT erreichen die einzelnen QPT2-Projekte inhaltlich größere Projektumfänge und können Voraussetzungen wie richtige Teamgrößen (vgl. 2.3.1) erreichen.

Zusammengefasst sind, nach der Meinung des Autors, für das QPT2 von MMA und QPT2b MMT die zeitlichen Rahmenbedingungen passend, um im Scrum-Prozess bearbeitet zu werden. Um die inhaltlichen Projektumfänge erreichen zu können, muss ein QPT2 Projekt von MMA die Zusammenarbeit mit einem QPT2b Projekt von MMT suchen.

## 2. QPT3-Projekte

Für die QPT3-Projekte existiert ein gemeinsamer Leitfaden für die Studiengänge MMA & MMT. Aus diesem ist zu entnehmen, dass ein QPT3-Projekt einen zeitlichen Umfang von knapp zehn Monaten einnimmt (Beispiel 20.07.2010 bis 30.05.2010). Sollte seitens der Studierenden erst die zweite oder dritte Abgabemöglichkeit wahrgenommen werden, erhöht sich die Dauer um eine bzw. zwei Wochen (MMA & MMT 2011*d*, S. 2).

Im Rahmen der QPT3-Projekte ist die Zusammenarbeit mit verschiedenen Fachbereichen Pflicht. (MMA & MMT 2011*d*, S. 3) Durch die zeitlichen Rahmenbedingungen von QPT3-Projekten und der Pflicht der Zusammenarbeit mit verschiedenen Fachbereichen (Stichwort: Teamgröße (vgl. 2.3.1) sieht der Autor im Einsatz von Scrum in QPT3-Projekten einen Sinn. Der Autor teilt damit die Meinung von Randelshofer. Dieser sieht im Einsatz von Scrum in QPT3-Projekten jedenfalls einen Sinn (vgl. A.2, Zeile 451).

## 3. Masterprojekte

Zurzeit (Stand 16.06.2011) gibt es keine Leitfäden für Masterprojekte der Studiengänge MMA & MMT. Es soll allerdings ein gemeinsamer Leitfaden entstehen (vgl. B, Zeile 591).

Masterprojekte sind laut Jellinek ähnlich zu den QPT3-Projekten. Der zeitliche Ablauf ist jedoch anders gestalten (vgl. B, Zeile 593). Masterprojekte finden über einen Zeitraum von vier Semestern statt.

Durch die zeitlichen Rahmenbedingungen und die Tatsache, dass Masterprojekte eine Steigerung von QPT3-Projekten darstellen sollten, sieht der Autor bei diesen Projekten den Einsatz von Scrum als sinnvoll an.

Ein Einsatz von Scrum bei einem QPT1-Projekt (MMA vgl. (MMA 2011), MMT vgl. (MMT 2011*a*)) sieht der Autor als nicht relevant an. Diese Projekte haben einen zeitlichen Rahmen von knapp einem Monat (MMA siehe (MMA 2011, S. 4), MMT siehe (MMA 2011, S. 6). Durch diese kurze Zeitdauer ist ein Einsatz von Scrum nicht sinnvoll. Des Weiteren sind die QPT1-Projekte in Einzelarbeit (MMA siehe (MMA 2011, S. 2), MMT siehe (MMA 2011, S. 3) zu erarbeiten. Teamprojekte werden laut den QPT1-Leitfäden nur in Ausnahmesituationen genehmigt. Voraussetzungen, wie beispielsweise die richtige Teamgröße (vgl. 2.3.1), können bei Einzelprojekten nicht erfüllt werden.

### 3.1.4 Potenzielle Scrum-Projekte

In diesem Unterkapitel soll die Anzahl an potenziellen Scrum-Projekten in den Studiengängen MMA und MMT ermittelt werden. Dabei wird die Berechnung in Bezug auf die Projekte durchgeführt, welche den richtigen Projektumfang (vgl. 3.1.3) und die passende Teamgröße (vgl. 3.3.1) erreichen können. Die resultierende Anzahl spiegelt die Projekte wider, die für die spätere Bestimmung von infrastrukturellen Voraussetzungen ausschlaggebend sind.

Zurzeit (Stand 17.06.2011) gibt es folgende Anzahl an StudentInnen pro Jahrgang in den Studiengängen MMA und MMT (vgl. B, Zeile 586ff). Der zukünftige MMT-Masterstudiengang wird bereits im Vollausbau berücksichtigt.

1. MMA-Bachelor: 60 - 65 Personen (3 Jahrgänge)
2. MMA-Master: 45 Personen (2 Jahrgänge)
3. MMT-Bachelor: 30 - 35 Personen (3 Jahrgänge)
4. MMT-Master: 20 Personen (2 Jahrgänge)

**Hinweis zur Berechnung** Die Bachelor-Jahrgänge werden jeweils einmal mitgerechnet, da während eines Semesters nicht 2 Jahrgänge des gleichen Studiengangs gleichzeitig am gleichen QPT-Typ arbeiten können. Beispielsweise ist es nicht möglich, dass die MMT-Bachelorjahrgänge 2008 und 2009 gleichzeitig am QPT3 arbeiten. Die Masterstudiengänge werden im Vollausbau einbezogen, da die Projekte über die gesamte Dauer des Master-Studiums erarbeitet werden (vgl. B, Zeile 593ff) und somit beide Jahrgänge gleichzeitig an den Projekten arbeiten können.

Projekt	MMA + MMT	Durchschnitt Team	Ergebnis
QPT2	60	4	15
QPT2b	30	2	15
QPT3	90	6	15
Masterprojekt	130	6	21,6
Gesamt			66,6

Tabelle 1: Potenzielle Scrum Projekte FHS

Tabelle 1 beinhaltet die Anzahl an potenziellen Scrum-Projekten innerhalb eines Semesters an der FHS in den Studiengängen MMA und MMT. Von diesen 66,6 Projekten sind jedoch nicht alle tatsächlich Scrum-fähige Projekte. Einige haben keinen hohen Software-Anteil oder es fehlen andere Voraussetzungen für den Einsatz von Scrum. Der Autor nimmt an, dass von diesen 66,6 Projekten ca. 35 Projekte tatsächlich mit Scrum durchgeführt werden könnten.

### 3.1.5 Vorwissen

MMA- und MMT-StudentInnen haben unterschiedliches Vorwissen betreffend dem Thema Agile Entwicklungsmethoden und Scrum. MMT-Studierende erlernen die Grundlagen zu diesem Themengebiet beispielsweise in der Integrierten Lehrveranstaltung „Software-Engineering“. In dieser Lehrveranstaltung liegt laut Curriculum ein Schwerpunkt auf dem Themengebiet der „Agilen Softwareentwicklung“. (MMT 2008) Seitens MMA-Studierenden sind dem Autor keinerlei Lehrveranstaltungen bekannt, welche dieses Themengebiet abdecken. Gucher ist der Meinung, dass das vorhandene bzw. fehlende Vorwissen betreffend Scrum kein Problem darstellt. Dadurch, dass sowie so verschiedene Fachbereiche aufeinander stoßen, ist auch innerhalb des Teams, fachlich gesehen, unterschiedliches Vorwissen vorhanden. Gucher sieht den Einsatz von Scrum als eine Management-Logik. Die iterative Entwicklung findet in der Folge sowohl im Design-, als auch im Programmierbereich statt (vgl. A.1, Zeile 217 - 227). Generell ist Gucher der Meinung, dass Scrum grundsätzlich

nicht schwer zu erlernen ist (vgl. A.1, Zeile 35). So können StudentInnen mit noch fehlendem Vorwissen zum Thema Scrum diese Defizite schnell beseitigen und mit den anderen StudentInnen mit Vorwissen gleichziehen.

### 3.1.6 Hindernisse und Ablenkungen

An der FHS arbeiten MMA- und MMT-StudentInnen nicht nur an einem Projekt. Es gibt verschiedene Lehrveranstaltungen und in jeder wird von den Studierenden Leistung verlangt. Es muss gleichzeitig an mehreren Projekten gearbeitet werden, eventuell stehen noch Klausuren und Präsentationen an. Des Weiteren sind nicht alle Studierenden jeden Tag von morgens bis abends an der FHS anwesend. Hier fehlt somit die Vollzeitanzwesenheit, welche beispielsweise in einem Unternehmen gegeben ist.

Durch diese Hindernisse und Ablenkungen kann der Scrum-Prozessablauf gestört und beeinflusst werden. Randelshofer empfiehlt genau in solch stressigen Situationen die grundsätzliche Idee von Scrum anzuwenden (vgl. A.2, Zeile 506). Dadurch, dass das Scrum-Team beispielsweise den Sprint selber plant, können eventuell anstehende Klausuren oder andere Hindernisse eingeplant werden. Haben StudentInnen während der nächsten zwei Wochen zum Beispiel vier Klausuren, drei Präsentationen und noch eine Abgabe eines anderen Projektes, so muss ein anstehender Sprint mit dementsprechend weniger Anforderungen aus dem Product Backlog befüllt werden. Hierbei spielt auch die richtige Priorisierung eine entscheidende Rolle. Stehen in den nächsten zwei Wochen die Weihnachtsferien bevor, muss seitens des Scrum-Teams alles vorbereitet werden, (z.B. Eintrag eines analogen Sprint Backlogs in eine elektronische Form) um auch in Abwesenheit von der FHS weiterentwickeln zu können.

Gucher geht über dies hinaus und findet diese Hindernisse und Ablenkungen vorteilhaft (vgl. A.1, Zeile 155ff). Später in der realen Wirtschaft werden StudentInnen eventuell mit der gleichen Situation Vorlieb nehmen müssen und gleichzeitig bei mehreren Projekten mitarbeiten. Die Situation an der FHS ist für Gucher aufgrund dessen bereits eine gute Vorbereitung für das spätere Berufsleben.

Der Autor teilt die Meinung von Randelshofer, sieht allerdings das Problem, dass durch Hindernisse die einzelnen Sprint-Ergebnisse stark voneinander abweichen können. In einem Sprint könnten beispielsweise Anforderungen mit 20 Wertungspunkten abgearbeitet werden, im nächsten Sprint werden nur drei Wertungspunkte erledigt. Durch diese starken Schwankungen könnte das Scrum-Team den Überblick, über die noch bis zum endgültigen Release zu erledigenden Aufgaben, verlieren. Des Weiteren könnten sich die Prognosen des Release-Burndowns, von Sprint zu Sprint, um mehrere Wochen verschieben. Das ist für die Planung sicherlich nicht ideal und muss seitens des Scrum-Teams von Anfang an mit eingerechnet werden.

### 3.1.7 Entlohnung

**Regel** In der Wirtschaft werden Scrum-Team-Mitglieder mit Geld, Prämien, Firmenautos, Firmenwohnung, etc., für erledigte Arbeiten entlohnt bzw. bezahlt. Wie diese Entlohnung ausfällt ist von den jeweiligen Qualifikationen, Umfang der Arbeit und Arbeitgebern abhängig. Getane Arbeiten und abgeschlossene Projekte können seitens der Arbeitnehmer als Referenzprojekte verwendet werden.

An der FHS werden abgeschlossene Projekte benotet und StudentInnen bekommen die dementsprechenden Beurteilungen. Die Arbeitsergebnisse können als Referenzprojekte ins Portfolio einer StudentIn aufgenommen werden. So haben StudentInnen die Möglichkeit durch Projekte an der FHS eine sehr gute Projekte-Sammlung mit eigenen Werken zu erlangen, sie werden dafür jedoch nicht bezahlt. Durch die im Studium erstellten Werke können sich StudentInnen für Jobs bewerben und dafür das Portfolio angeben.

**Ausnahme** Bei Projekten vom „Projekttyp nach Abläufen“ „Auftragsprojekt“ hat die Entlohnung gemäß der Wirtschaft zu erfolgen (vgl. 3.1.2, Abschnitt „Auftragsprojekt“). Dies ist sehr wichtig, da ansonsten die Entlohnung von Leistungen von AbsolventInnen der FHS in Zukunft zu niedrig eingeschätzt werden könnten. Dies würde zu einer generellen Verminderung der Entlohnung führen.

Eine weitere Möglichkeit StudentInnen zu entlohnen bzw. zu belohnen bilden Preisgelder. Diese können im Rahmen von Wettbewerben gewonnen werden.

## 3.2 Scrum - Rollen - Situation an der FHS

Aus der Literatur sind bereits die Rollen „Scrum MasterIn“, „Product OwnerIn“, das „Entwickler-team“ und sonstige Rollen, wie beispielsweise „Chickens“ oder „Stakeholder“, bekannt. (vgl. 2.2). Können diese Rollen deckungsgleich für die FHS adaptiert werden? Wie können sie sinnvoll besetzt werden? Welche Abänderungen müssen gemacht werden? Gibt es noch zusätzliche Rollen an der FHS? Dieses Unterkapitel geht in den folgenden Abschnitten auf die Situation an der FHS ein.

### 3.2.1 Scrum Master

An der FHS werden noch keine expliziten Scrum MasterInnen ausgebildet. Laut Gucher werden im Schwerpunkt „LEAD“, des zukünftigen Master-Studiengangs MMT (ab Wintersemester 2011), und im Schwerpunkt „Steuerung“, des Master-Studiengangs MMA, agile Entwicklungsmethoden vermehrt unterrichtet. Hier wird mit Scrum gearbeitet (vgl. A.1, Zeile 76). Die Rolle der Scrum MasterIn sollte in Folge eine StudentIn mit einem dieser fachlichen Schwerpunkte übernehmen. Dies bestätigt Randeshofer ebenfalls (vgl. A.2, Zeile 329). Diese Personen verfügen über das benötigte Grundwissen über Scrum, um ein Scrum-Team, fachlich gesehen, leiten zu können.

Wichtig ist, dass eine zukünftige Scrum MasterIn auch vom Scrum-Team akzeptiert wird und sie diese Rolle überhaupt ausführen will.

Scrum MasterInnen sind die ChefInnen über den Prozess und den Prozessablauf. An der FHS könnten Scrum MasterInnen, laut Randeshofer, über die Motivation und die Begeisterung der einzelnen Scrum-Team-Mitglieder, den Prozessablauf fördern und in einem guten Rahmen ablaufen lassen (vgl. A.2, Zeile 295f). Eine Scrum MasterIn muss jeden einzelnen darauf hinweisen, dass ein an der FHS durchgeführtes Projekt für ein gutes Portfolio, für spätere Bewerbungen, sehr wichtig ist. Die im Rahmen einer Ausbildung erstellten Projekte sind gerade in der jetzigen Zeit, in der multimedialen Branche, ein entscheidender Punkt für eine eventuelle Anstellung in einem Unternehmen. Diese Begeisterung des Einzelnen und somit des kompletten Scrum-Teams muss im Vordergrund stehen.

Der Autor sieht die Aufrechterhaltung des Prozessablaufes im Rahmen von längerfristigen Projekten als leichter an. Unter längerfristigen Projekten sind beispielsweise Masterprojekte, welche über vier Semester gehen, gemeint (vgl. B, Zeile 593f). Hier können StudentInnen Projekte bereits so auslegen, dass sie für eventuelle Startup-Unternehmen, nach erfolgreicher Absolvierung des Master-Studiums, verwendet werden können. Dies würde praktisch einer viersemestrigen Vorbereitung auf eine Unternehmensgründung gleich kommen. In solchen Projekten könnte die Scrum MasterIn mithilfe der Motivation einer Firmengründung arbeiten. Das Projekt könnte ja beispielsweise in einem späteren Unternehmen sehr erfolgreich werden und wirtschaftlich rentabel sein. Durch solche Visionen und Zukunftspläne könnte die Scrum MasterIn das restliche Scrum-Team motivieren und dadurch die Freude an der Arbeit steigern und im Endeffekt das Ergebnis verbessern.

Das Beseitigen von Hindernissen, seitens der Scrum MasterIn, wird im Rahmen der FHS womöglich nicht immer ganz leicht sein. Hier kommt es selbstverständlich darauf an, um welche Art von Hindernissen es sich handelt. Bei einem falsch konfigurierten Continuous Integration Server beispielsweise, muss eine Scrum MasterIn in der Lage sein, die Konfiguration wieder richtig zu

stellen oder die Richtigstellung zu veranlassen. Dadurch kommen Scrum MasterInnen in Kontakt mit LektorInnen und anderen FHS-Angestellten. Folglich müssen Scrum MasterInnen das gewisse Gespür haben und mit Autoritätspersonen (z.B. LektorInnen) kommunizieren können. Diese Kommunikation ist für einen reibungslosen Prozessablauf sehr wichtig.

In einer Firma muss eine Scrum MasterIn auch mit Autoritätspersonen kommunizieren können. Der Unterschied zur Situation an der FHS ist, dass sie in einer Firma ebenfalls als Autoritätsperson eingestuft werden kann. An der FHS ist dies nicht so leicht möglich. Scrum MasterInnen in MMA- und MMT-Projekten sind in erster Linie StudentInnen und keine Autoritätspersonen. Innerhalb des Scrum-Teams nehmen sie zwar die Rolle der Scrum MasterIn ein, sollten sie aber beispielsweise mit einem Problem betreffend einem Server zur IT-Abteilung gehen und die Problembeseitigung veranlassen wollen, so sind sie für die MitarbeiterInnen der IT-Abteilung „nur“ StudentInnen.

### 3.2.2 Product Owner

Dieses Unterkapitel beschäftigt sich mit der Rolle der Product OwnerIn an der FHS. Es ist in drei Unterabschnitte gegliedert. Der Abschnitt „Rollenbesetzung“ diskutiert verschiedene Möglichkeiten, wie die Rolle der Product OwnerIn besetzt werden könnte. Im zweiten Abschnitt „Entscheidungsfindung“ stellt der Autor dar wie Product OwnerInnen an der FHS zur Entscheidungsfindung kommen können. Der letzte Bereich dieses Unterkapitels beschäftigt sich mit der Kommunikation einer Product OwnerIn. In diesem Abschnitt wird auf jeden der „Projekttypen nach Abläufen“ (vgl. 3.1.2) eingegangen. Durch diese verschiedenen Typen muss eine Product OwnerIn mit unterschiedlichen KundInnen bzw. AuftraggeberInnen als Stakeholder kommunizieren. Wie diese Kommunikation im Rahmen der FHS stattfinden könnte, wird im Abschnitt „Kommunikation“ erläutert und diskutiert.

**Rollenbesetzung** Für Randelshofer ist die Rolle der Product OwnerIn die schwierigste Rolle innerhalb des Setting der FHS (vgl. A.2, Zeile 369). Um sie besetzen zu können, sollte laut Randelshofer, folgendermaßen vorgegangen werden:

Hinter dem Produkt steht eine Vision. Innerhalb des Projekt-Teams wird eine Person bestimmt, welche diese Vision im Rahmen hält und dafür sorgt, dass nach dieser vorgegangen wird. Diese Person wird Product OwnerIn. Die Entscheidung zur Auswahl dieser Rolle muss einstimmig passieren. Wichtig ist, dass hier die Tatsache, dass die Rolle der Product OwnerIn eine Projektmanagementaufgabe ist, beachtet wird. Personen, die gerne programmieren und aktiv am Entwicklungsprozess mitarbeiten und zusätzlich die Vision womöglich am besten vertreten können, sind nicht unbedingt gute Product OwnerInnen. Sie könnten zu sehr bei der technischen Umsetzung intervenieren.

Der Autor ist derselben Meinung wie Randelshofer und würde eine solche Rollenbesetzung befürworten. Über dies hinaus sieht der Autor in StudentInnen, welche den Fachbereich „LEAD“ des zukünftigen MMT-Masterstudiengangs (Stand 24.05.2011) oder den Fachbereich „Steuerung“ des MMA-Masterstudiengangs belegen, potenzielle KandidatInnen für die Rolle der Product OwnerIn. So könnte beispielsweise eine „LEAD“-StudentIn aus dem Masterstudiengang in einem QPT3 aus dem Bachelor-Studiengang die Rolle der Product OwnerIn übernehmen. „LEAD“-StudentInnen können dadurch Erfahrungen im Projektmanagement sammeln und besitzen zusätzlich noch ein gutes technisches Verständnis, welches die Kommunikation sicherlich erleichtert.

**Entscheidungsfindung** Product OwnerInnen müssen die endgültigen Entscheidungen treffen, sodass der Scrum-Prozess nicht zum Stillstand kommen kann. Aus diesem Grund würde der Autor vorschlagen, dass Product OwnerInnen die täglichen Entscheidungen alle alleine und selbstständig treffen sollten. Sollten größere Entscheidungen anstehen, würde der Autor ein Team-Meeting abhalten. In diesem Meeting können die einzelnen zur Verfügung stehenden Möglichkeiten mit dem kompletten Scrum-Team diskutiert werden und Lösungsvorschläge erarbeitet werden. Dadurch kann eine Product OwnerIn auf die Bedürfnisse der einzelnen Scrum-Team-Mitglieder eingehen

und somit leichter Entscheidungen treffen, welche dem gesamten Projekt-Team als sinnvoll und positiv erscheinen. Es kann sich jede/jeder in einem solchen Team-Meeting einbringen und ihre/seine Vorstellungen erläutern. Dadurch fühlt sich keine/keiner übergangen und jede Meinung kann kund getan werden. Unterschiedliche Meinungen können offen und ehrlich besprochen und abgewägt werden. Um die Entscheidung jedoch tatsächlich zu fällen und zu einem Ergebnis zu kommen, ist die Rolle der Product OwnerIn besetzt. Nach Meinung des Autors sind mit dieser Vorgehensweise die Ausführungen aus der Literatur auf die Gegebenheiten an der FHS am besten angewandt.

**Kommunikation** Die Product OwnerIn bildet die Kommunikationsschnittstelle zwischen dem Entwicklerteam und den Stakeholdern (vgl. 2.2.2). Im Rahmen der FHS können je nach „Projekttyp nach Abläufen“ verschiedene Stakeholder (vgl. 3.1.2) und somit verschiedene Kommunikationsarten bzw. Kommunikationswege in Erscheinung treten. In der Folge werden die AuftraggeberInnen bzw. KundInnen der Stakeholder diskutiert:

- klassisches Studentenprojekt:

Beim klassischen Studentenprojekt tritt der Fall ein, dass die Scrum-Team-Mitglieder eigentlich zugleich auch ihre eigenen KundInnen sind (vgl. 3.2.4, Abschnitt „Stakeholder“, Punkt „klassisches Studentenprojekt“). In diesem Fall müssen die Product OwnerInnen mit dem Entwicklerteam und der Scrum MasterIn auf zwei verschiedene Arten kommunizieren. Die Kommunikation muss aus der Kundensicht und aus der „Projekt-Team-Mitglied-Sicht“ geführt werden. Wichtig ist dabei die Rollenabgrenzung. Werden die Rollen vermischt (z.B. Entwicklerteam-Mitglied - KundIn), kann nicht mehr gut gearbeitet werden. Denken EntwicklerInnen beispielsweise nur technisch, können technisch gute Projekte entstehen. Werden diese Projekte allerdings nicht aus der Kundensicht betrachtet und bearbeitet, könnten sie unbrauchbar werden. Die Achse dieser Kommunikation muss die Product OwnerIn halten und dafür sorgen, dass die Rollen abgegrenzt werden und eingehalten werden.

Der Autor würde als Lösung für das Problem des Rollenkonflikts vorschlagen, dass die Product OwnerInnen getrennte Meetings für jede Ansicht (EntwicklerIn - KundIn) mit dem Scrum-Team abhalten. In den ersten Meetings werden beispielsweise die Anforderungen festgehalten (vgl. „Product Backlog 2.5.1“). Diese Anforderungen werden in Stories aus der Kundensicht verfasst (Stichwort: SDD, vgl. 2.4.3). Wichtig ist dabei, dass über technische Details kein Wort verloren wird. Dieses erste Meeting gehört nur der Kundensicht.

Sind die Anforderungen im Product Backlog eingepflegt, kann die Product OwnerIn diese richtig priorisieren. Im normalen Scrum-Verlauf, beispielsweise im Sprint-Planning, kann die Kommunikation aus der „Projekt-Team-Mitglied-Sicht“ geführt werden. In diesen Meetings werden technische Details für die Umsetzung besprochen und geplant (vgl. 2.6.2).

Durch diese Vorgangsweise ist gewährleistet, dass keine Ansicht zu kurz kommt und das Problem des Rollenkonflikts, nach der Meinung des Autors, am einfachsten und besten aus der Welt geschaffen werden kann.

- FHS internes Projekt:

Bei diesem „Projekttyp nach Abläufen“ ist die LektorIn, welche das interne Projekt ins Leben ruft, AuftraggeberIn und somit KundIn (vgl. 3.2.4, Abschnitt „Stakeholder“, Punkt „FHS internes Projekt“). Die Product OwnerIn muss in der Folge die Anforderungen für den Product Backlog gemeinsam mit dieser LektorIn erarbeiten.

Aus eigenen Erfahrungen des Autors geht zwar meistens die Initiative von FHS internen Projekten von LektorInnen der Studiengänge MMA oder MMT aus, allerdings geben diese nur gewisse Rahmenbedingungen vor. Die StudentInnen verfügen also über gewisse Gestaltungsfreiheiten bei der Ausarbeitung der Projekte. Aufgrund dessen würde der Autor vorschlagen,

dass beim ersten Meeting nicht nur die LektorIn als KundIn anwesend ist, sondern das komplette Scrum-Team ebenfalls in die Kundenrolle schlüpft und am Meeting teilnimmt. Dadurch können die grundlegenden Anforderungen für den späteren Product Backlog gemeinsam diskutiert und erarbeitet werden. Wichtig ist, dass dieses Meeting frei von technischen Diskussionen bleibt.

Bei den nachfolgenden Meetings zwischen der KundIn (LektorIn) und der Product OwnerIn sollte das restliche Scrum-Team, nach der Meinung des Autors, nicht mehr anwesend sein. Die grundlegenden Anforderungen sind ja bereits geklärt worden. Sollte es in nachfolgenden Meetings doch noch größere Anforderungsänderungen geben, könnte die Product OwnerIn nochmals ein gemeinsames Meeting einberufen (LektorIn und Scrum-Team anwesend). Nach der Meinung des Autors ist dies aber von Fall zu Fall seitens der Product OwnerIn abzuwägen.

- Forschungsprojekt:

Da beim Forschungsprojekt die Forschungsbeauftragte als AuftraggeberIn und KundIn auftritt (vgl. 3.2.4, Abschnitt „Stakeholder“, Punkt „Forschungsprojekt“) und diese Person, nach der Meinung des Autors, ähnlich als eine LektorIn zu behandeln ist (oft ist die Forschungsbeauftragte eine LektorIn) können die gleichen Kommunikationswege wie beim obigen Punkt „FHS internes Projekt“ gewählt werden.

Der einzige Unterschied zum obigen Punkt ist, nach der Meinung des Autors, die Gestaltungsfreiheit der StudentInnen. Diese ist bei Forschungsprojekten eingeschränkter als bei FHS internen Projekten. Die Product OwnerIn muss aus diesem Grund öfters den Kontakt und die Kommunikation zur Forschungsbeauftragten herstellen. Die im Laufe des Projektes gewonnenen Erkenntnisse betreffend dem Forschungsgebiet müssen laufend mit der Forschungsbeauftragten ausgetauscht werden.

- Auftragsprojekt:

Bei Auftragsprojekten ist die externe Firma AuftraggeberIn und KundIn (vgl. 3.2.4, Abschnitt „Stakeholder“, Punkt „Auftragsprojekt“). Diese Aufträge werden von externen Firmen an die Studiengänge und anschließend an die StudentInnen übergeben (vgl. 3.1.2, Abschnitt „Auftragsprojekt“). Durch diese Übergabe schaltet sich eine Person des Studiengangs in die Kommunikation mit ein. Beispielsweise könnte diese Person die betreuende Lehrperson des Projektes sein. Diese Person muss bei der Kommunikation auch berücksichtigt werden.

Die Product OwnerInnen müssen somit über die zuständige Person des Studiengangs mit den externen AuftraggeberInnen die Kommunikation aufbauen. Der Autor schlägt hierfür ein gemeinsames Meeting vor. Bei diesem Meeting nehmen die Product OwnerIn, die entsprechende Person des Studiengangs und die externen AuftraggeberInnen teil. Bei diesem Meeting werden grundlegende Positionen besprochen und Anforderungen festgelegt. In der Folge sollte, nach Meinung des Autors, dieses Meeting wiederholt werden. Ständige Rückmeldungen und Austausch zwischen externen AuftraggeberInnen, Person des Studiengangs und Product OwnerIn sind wichtig.

Die hier beschriebenen Kommunikationsmöglichkeiten und Kommunikationswege können nicht in allen Projekten zu 100% umgesetzt werden. Es kann immer Ausnahmen geben. Der Autor denkt aber, dass durch die beschriebenen Arten zu kommunizieren eine gute Basis für gute Projekte geschaffen werden kann.

Die restlichen Stakeholder, welche bereits im Abschnitt „Stakeholder“ (vgl. 3.2.4, Abschnitt „Stakeholder“) erläutert wurden, dürfen nicht vergessen werden. Ihre Teilnahme an gewissen Meetings wurde bereits angesprochen (z.B. Teilnahme am Sprint Review, siehe 3.6.5).

### 3.2.3 Entwicklerteam

Die Rolle des Entwicklerteams ist an der FHS leichter zu besetzen als die Rolle der Scrum MasterIn und Product OwnerIn. Das fachliche Know-How wird im Rahmen der FHS in den dementsprechenden Lehrveranstaltungen unterrichtet und ist folglich vorhanden. Unter dem fachlichen Know-How sind beispielsweise Programmierfähigkeiten zu verstehen. Die einzelnen Entwicklerteam-Mitglieder bringen individuelles Fachwissen und Spezialisierungen in das Scrum-Team mit. Die Ausprägung des Wissens und die unterschiedlichen Erfahrungen der einzelnen Mitglieder sind an der FHS gleich wie in der Wirtschaft. Eine EntwicklerIn hat beispielsweise ein hohes Wissen bei der Programmierung von automatisierten Tests, eine andere hat als Spezialgebiet das Datenbankdesign. Ein Scrum-Team an der FHS hat somit die gleichen Voraussetzungen wie ein Team in einem richtigen Unternehmen. Einige Entwicklerteam-Mitglieder verfügen bereits über Vorwissen zum Thema Scrum, andere nicht. Dies wurde bereits im Punkt „Vorwissen 3.1.5“ diskutiert.

Die Selbstorganisation des Teams muss zuerst erlernt werden. Da die Scrum MasterIn vermutlich nicht über die nötige Erfahrung verfügt, einem Entwicklerteam diese Selbstorganisation beizubringen, wäre nach Meinung des Autors, das Hinzuziehen einer erfahrenen Scrum MasterIn, für den Anfang, sinnvoll. Es könnte zum Beispiel eine StudentIn aus dem Fachbereich „Steuerung“ des MMA-Masters, welche bereits Projekte als Scrum MasterIn durchgeführt hat, der tatsächlich vom Team gewählten Scrum MasterIn eines QPT3s zur Unterstützung zur Seite stehen. Hier könnte nach der Meinung des Autors eine Art „Mentor“-Programm entstehen.

Zu Beginn hat die Scrum MasterIn viele Aufgaben und noch keine Erfahrungen. Hier hilft die „externe“ Scrum MasterIn beispielsweise dem kompletten Scrum Team, Selbstorganisation zu lernen. Hat das Entwicklerteam und die wirkliche Scrum MasterIn Erfahrungen gemacht, kann die „externe“ Scrum MasterIn wieder abgezogen werden. Diese Vorgehensweise bringt Vorteile für beide Seiten mit sich. Die neue Scrum MasterIn kann vom Wissen der erfahrenen Scrum MasterIn profitieren und diese kann sich zusätzlich noch Coaching-Fähigkeiten aneignen.

Grundsätzlich kann gesagt werden, dass geeignete Personen für das Entwicklerteam an der FHS vorhanden sind (vgl. A.2, Zeile 353).

### 3.2.4 Sonstige Rollen

An der FHS gibt es, ebenso wie gemäß der Literatur, sonstige Rollen, auf welche nun eingegangen wird. Der Autor ist der Meinung, dass die Stakeholder-Zuteilung sehr stark vom „Projekttyp nach Abläufen“ abhängig ist (vgl. 3.1.2).

**Chickens** Chickens wurden bereits erklärt (vgl. Chickens 1). Die Situation an der FHS ist laut der Meinung des Autors komplexer, als es zuerst den Anschein macht. Es gibt einige Interessengruppen, welche sich als außenstehende Personen für Projekte interessieren. Dazu zählen zum Ersten MitstudentInnen des gleichen Jahrgangs wie StudentInnen einer Projektgruppe. Diese sind sicherlich interessiert an neuen Projekten und Errungenschaften von MitstudentInnen. Dadurch kann das Wissen von einzelnen Studierenden an den eigenen Jahrgang vermittelt und übergeben werden. Es werden neue Lösungsansätze demonstriert und dies regt wiederum MitstudentInnen an, diese neuen Lösungswege selber auszuprobieren und sich in der Folge weiter zu entwickeln. Des Weiteren zählt der Autor zusätzlich noch StudentInnen von jüngeren Jahrgängen dazu. Diese schauen zu ihren Vorgängern auf und können sich durch deren Leistungen ausmalen, was sie selber alles noch erlernen werden und in welche Richtung das entsprechende Studium läuft. So können jüngere Generationen ihre eigene, zukünftige Entwicklung besser einschätzen und beurteilen. Dies hilft ihnen, sich leichter darauf einzustellen und den Fokus auf fachlich interessante Bereiche, von Beginn an, setzen zu können.

**Stakeholder** An der FHS gibt es jede Menge potenzielle Stakeholder bzw. KundInnen. Hier spielt der „Projekttyp nach Abläufen“ eine entscheidende Rolle (vgl. 3.1.2). Folgende Stakeholder könnten, nach der Meinung des Autors generell in Projekten vorkommen:

- Studiengangleitung

Das Interesse der Studiengangleitungen liegt in guten Endergebnissen von Projekten der Studiengänge MMA und MMT. Mit diesen Projekten können sich die Studiengänge der Öffentlichkeit präsentieren und dadurch das Interesse an den Studiengängen für neue BewerberInnen wecken. Zur Studiengangleitung zählt der Autor die StudiengangleiterIn und die Administration.

- FachbereichsleiterIn

Die FachbereichsleiterIn leitet einen Fachbereich und ist an den Projekten aus dem Fachbereich besonders interessiert. Sie/Er möchte gute Ergebnisse von Projekten von Studierenden des Fachbereiches sehen.

- Betreuende Lehrperson

Die direkte AnsprechpartnerIn von Projektgruppen wird als betreuende Lehrperson bezeichnet. Sie betreut das Projektteam und steht ihm zur Seite. Des Weiteren ist sie für die Beurteilung des Endergebnisses verantwortlich.

- Fachliche Betreuungsperson

Unter der fachlichen Betreuungsperson versteht der Autor Personen, welche die Projekte in speziellen Fachgebieten und Bereichen betreuen. Implementiert beispielsweise ein Scrum-Team einen Algorithmus für ein 3D-Game mit einem hohen mathematischen Anteil, wäre das hinzuziehen einer Mathematik-LehrerIn als fachliche Betreuungsperson von Vorteil.

- Forschungsbeauftragte

Forschungsbeauftragte sind Personen der Studiengänge, die an einem Forschungsprojekt arbeiten bzw. ein Forschungsprojekt betreuen oder leiten.

- ProjektmitarbeiterInnen

Unter ProjektmitarbeiterInnen versteht der Autor Mitglieder des Scrum-Teams selber. Diese können als ihre eigenen KundInnen auftreten.

- EndnutzerInnen

Die Gruppe der EndnutzerInnen könnten beispielsweise MitstudentInnen darstellen. Das Interesse dieser Gruppe liegt in der tatsächlichen Endnutzung des Projektes nach dessen Fertigstellung. EndnutzerInnen könnten aber auch Personen außerhalb der FHS sein. Erstellt ein Scrum-Team beispielsweise eine Social Community, könnten Familie, Freunde und Bekannte von StudentInnen auch als EndnutzerInnen eingestuft werden.

Um eine generelle Zuteilung von Stakeholder, im Rahmen dieser Bachelorarbeit, durchführen zu können, hat sich der Autor auf die aufgezählten Stakeholder beschränkt. Es kann immer wieder Projekte geben, die noch andere Interessengruppen haben. Erstellt ein Scrum-Team beispielsweise ein neues Wiki für die Studiengänge MMA und MMT, so muss beim Login in dieses Wiki die Sicherheit bei der Passwortübertragung des Fachhochschule (FH)-Passwortes einer StudentIn gewährleistet sein. In diesem Fall hat wahrscheinlich die IT-Abteilung der FHS ein Interesse daran, dass hier alles mit rechten Dingen abläuft und die Übertragung sicher stattfindet. In diesem Falle wäre somit die IT-Abteilung ein potenzieller Stakeholder.

Der Autor würde abhängig vom „Projekttyp nach Abläufen“ folgende Personen als Stakeholder zuteilen:

**Hinweis** Er nimmt dabei an, dass jeder Stakeholder eine andere Person ist. Wäre die betreuende Lehrperson beispielsweise zugleich die FachbereichsleiterIn, stehen beide als Stakeholder aufgelistet da.

- klassisches Studentenprojekt:  
Studiengangleitung, FachbereichsleiterIn, betreuende Lehrperson, eventuell fachliche Betreuungsperson, EndnutzerInnen, ProjektmitarbeiterInnen als KundInnen
- FHS internes Projekt:  
LektorIn seitens des Studiengangs, der internes Projekt benötigt, als AuftraggeberIn und KundIn (vgl. B, Zeile 563f), Studiengangleitung, FachbereichsleiterIn, betreuende Lehrperson, eventuell fachliche Betreuungsperson, EndnutzerInnen. Die AuftraggeberIn und betreuende Lehrperson wird in den meisten Fällen die gleiche Person sein.
- Forschungsprojekt:  
Forschungsbeauftragte als AuftraggeberIn und KundIn (vgl. B, Zeile 565f), Studiengangleitung, FachbereichsleiterIn, betreuende Lehrperson, eventuell fachliche Betreuungsperson, EndnutzerInnen.
- Auftragsprojekt:  
Externe Firma als AuftraggeberIn und KundIn, Studiengangleitung, FachbereichsleiterIn, betreuende Lehrperson, eventuell fachliche Betreuungsperson, EndnutzerInnen.

Die hier, seitens des Autors, getätigten Stakeholder-Zuteilungen unterscheiden sich nur minimal. Das liegt daran, dass beispielsweise die Studiengangleitungen und FachbereichsleiterInnen ein Grundinteresse an den Projekten von StudentInnen der Studiengänge haben und deshalb immer zugeteilt werden können. Projekt-Teams benötigen betreuende Lehrpersonen und in manchen Fällen fachliche Betreuungspersonen, aus diesem Grund wurden sie ebenfalls immer zugeteilt. Die Projekte sollen der Öffentlichkeit präsentiert werden und für StudentInnen positive Auswirkungen haben, dadurch sind bei jedem Projekttyp die EndnutzerInnen als Stakeholder zugeteilt. Sie verwenden die Projekte nach der Veröffentlichung.

Hauptunterschiede sieht der Autor in den AuftraggeberInnen bzw. KundInnen. Diese sind von Projekttyp zu Projekttyp unterschiedlich angeführt.

Diese Stakeholder-Zuteilungen treffen natürlich nicht in jedem Fall zu 100% zu. Da zu Projektbeginn oft nur schwer einzuschätzen ist, wie sich der Projekt-Verlauf entwickelt, würde der Autor anfangs eher zu viele Stakeholder mit einbeziehen und mit Verlauf des Projektes, beispielsweise durch Usability-Tests, Umfragen, Analysen oder Ähnliches, falsche Stakeholder aussortieren und durch eventuell anfangs nicht mit einbezogene Interessengruppen nach besetzen.

Abschließend ist noch festzuhalten, dass trotz möglichem regen Interesse und vielen verschiedenen Stakeholdern, ein Scrum-Team innerhalb des Scrum-Prozessablaufes, hauptsächlich mit den direkt betreuenden und beurteilenden Lehrpersonen zu tun hat. Dies bestätigt Randelshofer in seinen Aussagen - er würde die betreuende Lehrperson in jedem Fall als Stakeholder ansehen (vgl. A.2, Zeile 339ff).

### 3.3 Scrum - Voraussetzungen - Situation an der FHS

Aus der Literatur sind bereits die Voraussetzungen für den Einsatz von Scrum erläutert worden (siehe 2.3). Sind diese Voraussetzungen an der FHS gegeben? Welche Schritte sind notwendig, um fehlende Voraussetzungen zu beseitigen? Dieses Unterkapitel geht in den folgenden Absätzen auf die Situation an der FHS ein.

#### 3.3.1 Teamgröße

Die Teamgröße an der FHS hängt vom Umfang bzw. der Art der Projekte ab. Es gibt Projekte, welche alleine durchgeführt werden müssen bzw. können und Projekte die im Team gestaltet werden müssen bzw. können. Fest steht, dass gemäß Literatur eine Teamgröße von fünf bis neun Personen, einzuhalten ist (vgl. „Teamgröße 2.3.1“).

Bei folgenden Projekten könnte unter gewissen Umständen diese Anzahl Teammitglieder in einem Projekt erreicht werden:

- Qualifikationsprojekt 2 MMA:

Der Leitfaden für das Qualifikationsprojekt 2 des Studiengangs MMA besagt, dass sowohl Einzel- als auch Gruppenarbeiten möglich sind. Es wird jedoch empfohlen bei größer angelegten Projekten mit dem Studiengang MMT zusammen zu arbeiten. (MMA 2008) Wird beispielsweise ein solches Projekt mit einem Qualifikationsprojekt 2b von MMT kombiniert und zusammengearbeitet, könnte die benötigte Anzahl für die richtige Teamgröße erreicht werden.

- Qualifikationsprojekt 2a MMT:

Der Leitfaden für das Qualifikationsprojekt 2a des Studiengangs MMT besagt, dass mindestens 2, maximal 3 Studierende von MMT an einem solchen Projekt beteiligt sein müssen. Eventuell sollten noch StudentInnen von MMA hinzugezogen werden. (MMT 2010) Der Autor zieht in der Folge den Schluss daraus, dass sich ein QPT2a in Zusammenarbeit mit MMA-StudentInnen zu einem größeren Projekt mit der erforderlichen Teamgröße entwickeln könnte.

- Qualifikationsprojekt 2b MMT:

Beim Qualifikationsprojekt 2b von MMT wird von einer Mindestanzahl von zwei und einer Maximalanzahl von vier ProgrammiererInnen seitens MMT gesprochen. Bei diesem Projekt liegt ein starker Fokus auf der Teamarbeit. Aus diesem Grund ist eine Zusammenarbeit mit dem Studiengang MMA sehr sinnvoll und wird empfohlen. (MMT 2011b) Die Kombination mit einem QPT2 von MMA bietet sich hier an, um die Teamgröße erreichen zu können.

- Qualifikationsprojekt 3 MMA & MMT:

Das Qualifikationsprojekt 3 ist ein Teamprojekt und es müssen laut Leitfaden die einzelnen Mitglieder aus verschiedenen Fachbereichen und Studiengängen kommen. (MMA & MMT 2011d) Dieses Projekt hat auch vom Umfang her die Voraussetzungen um in einem Team von fünf bis neuen Personen bearbeitet zu werden. Die passende Teamgröße ist bei diesem Projekt somit grundsätzlich gewährleistet.

- Masterprojekt MMA & MMT:

Masterprojekte stellen eine weitere Steigerung dar und sind aus diesem Grund auch in der passenden Teamgröße zu erarbeiten. Für MMA & MMT liegt leider noch kein expliziter Leitfaden vor, aus dem die Bestimmungen entnehmbar wären.

Der Autor ist aufgrund eigener Erfahrungen der Meinung, dass erst in einem QPT2 (MMA) und QPT2b (MMT), QPT3 und in Masterprojekten sinnvoll mit einer Teamgröße von fünf bis neun Personen gearbeitet werden kann und sich überhaupt so viele Mitglieder finden lassen. Das QPT2a von MMT stuft er noch als zu klein ein.

Des Weiteren ist festzuhalten, dass die richtigen Teamgrößen nicht in allen Projekten gegeben sind, welche unter grundsätzlich Scrum-„fähige“ Projekte eingestuft werden. Es kann auch QPT3 Projekte geben, in denen nur drei StudentInnen zusammenarbeiten. Andererseits kann es auch QPT2a Projekte geben, bei denen genügend Ressourcen zur Verfügung stehen. Die hier getätigten Aussagen sind auf die Analyse der Leitfäden und eigenen Erfahrungen seitens des Autors gestützt, werden aber nicht immer zu 100% zutreffen.

### 3.3.2 Teammitglieder & Teamfähigkeit

Die Situation der einzelnen Scrum-Rollen an der FHS wurde bereits erläutert. Die, für Scrum, so wichtige Teamfähigkeit, der einzelnen Mitglieder, wird in den Studiengängen MMA und MMT seit den ersten Semestern als wichtiger Punkt angesehen. In Lehrveranstaltungen wie „Selbstmanagement“ oder „Projektmanagement“ findet dieses Thema Platz. Im Scrum-Prozess ist aber natürlich mehr nötig, als nur ein theoretisches Hintergrundwissen. Der Autor ist der Meinung, dass eine nicht-teamfähige Person sich durch den Prozessablauf trotzdem im Scrum-Team etablieren kann. Durch die Selbstorganisation des Teams wird viel kommuniziert, diskutiert und besprochen. Durch diesen Austausch können sich die Studierenden besser in die Situation integrieren. Scrum fordert und fördert die Teamfähigkeit.

Mit den jetzigen Statuten aus den Leitfäden (Stand 09.06.2011) ist das Verlassen eines Teams, bzw. der Wechsel in ein anderes, nur eher schwierig durchzuführen. Es sind enge Zeitpläne vorgegeben und diese müssen, seitens der Studierenden, eingehalten werden. Ein Teamwechsel macht, nach der Meinung des Autors, bei Projekten, welche kürzer als ein halbes Jahr dauern (z.B. QPT2a, QPT2b MMT), wenig bzw. keinen Sinn. Hier ist die Zeit zu kurz, um in der Mitte der Projektdauer ein Team zu verlassen und zu einem neuen hinzuzustoßen. Die StudentInnen haben keine Zeit sich im neuen Team an die Gegebenheiten anzupassen und müssen zuerst mit dem Projekt vertraut werden. Dadurch fallen sie den anderen Mitgliedern zur Last und können somit kaum am Projekt mitarbeiten. Das Team, das von einem Mitglied verlassen wird, könnte Probleme bekommen die entstandene Personallücke zu schließen, da alle anderen Ressourcen bereits aufgeteilt sind.

Bei QPT3- oder Masterprojekten sieht der Autor diese Situation anders. Hier wäre ein Wechsel möglich, sollte sich eine Person in einem Team einfach nicht zurecht finden können. Die Dauer des QPT3s mit zehn Monaten (vgl. Leitfaden (MMA & MMT 2011*d*, S.2)) und die Dauer von Masterprojekten über das gesamte Masterstudium hinweg (vgl. B, Zeile 593), bieten den Freiraum, solche Wechsel durchführen zu können. Ein Wechsel selbst sollte im ersten Drittel, bis maximal der Hälfte der Projektzeit vollzogen werden können. Bei einem Wechsel zu einem späteren Zeitpunkt sieht der Autor wieder das selbe Problem wie vorhin bereits beschrieben.

Der QPT3-Leitfaden beinhaltet einen Punkt „Notfallsituationen“. Dieser behandelt Problemfälle wie die fehlende Termintreue oder Nichteinhaltung der erforderlichen Qualität von einzelnen Teammitgliedern (MMA & MMT 2011*d*, S. 10). Nach Meinung des Autors sollte dieser Bereich erweitert werden. Die Möglichkeit eines Teamwechsels, innerhalb des ersten Projektdrittels bis maximal der Hälfte der Projektzeit, sollte mit einbezogen werden.

Generell ist noch festzuhalten, dass nicht wegen jeder Kleinigkeit sofort ein Teamwechsel durchgeführt werden soll. In einem Scrum-Team kann es immer wieder zur Problemen und Meinungsverschiedenheiten kommen. Diese sollten aber im Prozess abgearbeitet werden. Für genau solche Problemfälle stehen dem Scrum-Team Meetings wie die Sprint Retrospektive oder das Daily-Scrum-Meeting zur Verfügung.

### 3.3.3 Infrastruktur

Um den Einsatz von Scrum in Projekten der Studiengänge MMA & MMT sinnvoll gestalten zu können, müssen infrastrukturelle Voraussetzungen geschaffen werden. Es müssen die benötigten Räume, Computerunterstützung, Utensilien, etc. vorhanden sein (vgl. „Infrastruktur 2.3.3“).

Falls nicht durch andere Quellen angegeben, hat der Autor durch eigene Erkundungen die benötigten Informationen zu folgenden Abschnitten zusammengetragen.

#### Räume inklusive Ausstattung

- Räume inklusive Ausstattung - Ist-Zustand

Zurzeit (Stand 17.06.2011) stehen für die Studierenden folgende Räume zur Benutzung zur Verfügung:

##### 1. Seminarräume

Die Seminarräume können seitens der Studierenden gebucht werden. Dies kann der Autor aus eigenen Erfahrungen bestätigen. Allerdings ist festzuhalten, dass diese Räume oft durch Lehrveranstaltungen besetzt sind. Aus den Erfahrungen des Autors haben Lehrveranstaltungen eine höhere Priorität. StudentInnen sollten frühzeitig an die Buchung eines solchen Raumes denken, da sie ansonsten oft schon vergeben sind.

Diese Räume verfügen, außer einem Computer am Vortragepult, über keinerlei Computerausstattung. W-LAN-Verbindungen sind sehr wohl vorhanden. Dadurch können StudentInnen ihre eigenen Laptops mit Internet in diesen Räumen verwenden. Des Weiteren sind in den Seminarräumen Whiteboards, Flipcharts, Flipchart-Papier, Pin-Leisten und teilweise mobile Pinnwände vorhanden.

Zu den Seminarräumen zählen folgende Räumlichkeiten:

- U SE 333
- U SE 354
- U SE 355
- U SE 356
- U SE 357

##### 2. Labore

Die Labore verfügen über eine Computerausstattung (MMA & MMT 2011*a*). Es sind zusätzlich Whiteboards, Flipcharts, Flipchart-Papier sowie Pin-Leisten verfügbar. Zu den Laboren zählen folgende Räume:

- U LB 315

Das Labor U LB 315 wird auch Multimedia-Lab genannt. Dieser Raum steht hauptsächlich für Lehrveranstaltungen zur Verfügung. Sollte er einmal nicht besetzt sein, kann er eventuell gebucht werden (MMA & MMT 2011*a*).

- U LB 351

Der Labor-Raum U LB 351 ist das MMT-Lab und steht hauptsächlich für Lehrveranstaltungen zur Verfügung. Bei diesem Labor gilt das Gleiche wie bei U LB 315. Falls der Raum einmal frei wäre, könnte er ebenfalls gebucht werden (MMA & MMT 2011*a*).

- U LB 363

Das Labor U LB 363 ist das Game-Labor. Er steht für Game-Projekte zur Verfügung. Dieser Raum ist buchbar. Größere Game-Projekte, welche beispielsweise in den Masterstudiengängen entwickelt werden, haben eine größere Chance diesen Raum zur Verfügung gestellt zu bekommen (MMA & MMT 2011*a*).

### 3. Sonstige Räume

Es stehen für MMA- und MMT-StudentInnen noch sonstige Räume zur Verfügung. Diese Räume verfügen über keine Computerunterstützung und bieten des Weiteren keine Utensilien (z.B. Whiteboards, Flipcharts). W-LAN steht zur Verfügung.

Zu diesen Räumen zählen folgende: (MMA & MMT 2011a)

(a) Terrarium

Das Terrarium ist ein öffentlich zugänglicher Raum. Er ist nicht versperrbar und kann nicht fix gebucht werden.

(b) U LB 360

Dieser Raum beinhaltet Projekt-Kojen. Diese Projekt-Kojen können von Studierenden wochen- und monatsweise gebucht werden. Zutritt zu diesem Raum haben nur Personen mit den entsprechenden Berechtigungen. Diese können mit ihren FHS-Zugangskarten in den Raum gelangen. Der Raum ist somit nicht öffentlich zugänglich (MMA & MMT 2011c).

(c) U LB 362

Dieser Raum ist ein Lagerraum des MMA Fachbereiches „Mediendesign“. Im Sommersemester 2011 wurde er für ein Game-Projekt benutzt. Daraus schließt der Autor, dass dieser Raum über Sonderregelungen von StudentInnen gebucht werden kann.

(d) U 56

Der U56 ist ein kleiner Kellerraum (MMA & MMT 2011a). Dieser Raum kann von Studierenden wochen- und monatsweise gebucht werden (MMA & MMT 2011b).

Zusätzlich zu den durch Quellen oder eigene Erkundungen bestätigten Räumen können StudentInnen noch weitere Räumlichkeiten an der FHS buchen. Hier muss zuerst angeführt werden, dass die Studiengänge MMA und MMT im 3. Stock des FHS-Gebäudes untergebracht sind (Stand 29.06.2011). Andere Stockwerke sind durch andere Studiengänge und Personen besetzt. Der Autor kann durch eigene Erfahrungen sagen, dass aber auch Räume beispielsweise im 2. oder 4. Stock des FHS-Gebäudes tageweise gebucht werden können. Durch frühzeitiges nachfragen konnte hier die Projektgruppe des Autors während des QPT3-Projektes beispielsweise Räume im 2. und 4. Stock für Projektstage buchen. Diese Räume waren meistens ähnlich zu den Seminarräumen ausgestattet. Da diese Räume jedoch nicht den direkten Räumlichkeiten der beiden Studiengänge MMA und MMT unterliegen, sind sie nicht in der obigen Liste angeführt.

Werden alle Räume der obigen Liste zusammengezählt, kann festgehalten werden, dass grundsätzlich zwölf Räume für MMA- MMT-StudentInnen zur Verfügung stehen. Davon sind acht Räume mit Whiteboards, Flipcharts, etc. ausgestattet.

- Räume inklusive Ausstattung - Soll-Zustand

Scrum-Projekte sollten innerhalb eines Raumes stattfinden (siehe 2.3.3). Im Abschnitt „Anzahl potenzieller Scrum-Projekte 3.1.4“ wurde eine Anzahl von 66,6 Projekten, welche mit Scrum erarbeitet werden könnten, errechnet. Diese Anzahl reduziert sich laut Meinung des Autors auf 35 Projekte. Die Gründe dafür wurden bereits erläutert (siehe 3.1.4).

Wird davon ausgegangen, dass an jedem dieser 35 Projekte Vollzeit gearbeitet wird, und jedes Projekt einen eigenen Raum benötigt, ergibt sich daraus ein Raumbedarf von 35 Räumen. Durch bereits erläuterte Gegebenheiten an der FHS (vgl. Abschnitt „Hindernisse und Ablenkungen 3.1.6“) kann jedoch nicht davon ausgegangen werden, dass alle diese 35 Projekte in Vollzeitarbeit erstellt werden und somit auch nicht täglich einen Raum benötigen. Des Weiteren gibt es Scrum-Meetings, die nach der Meinung des Autors, mit mehreren Projektgruppen gemeinsam durchführbar sind (vgl. 3.6.5). Diese Gegebenheiten bzw. Faktoren vermindern den effektiven Raumbedarf.

Auf der anderen Seite benötigen nicht nur StudentInnen, welche an Scrum-Projekten arbeiten, Räumlichkeiten. Film-Projekte beispielsweise benötigen für Dreh-Szenen oder die Postproduktion mitunter auch Platz um die Arbeiten erledigen zu können. Dies kann der Autor aus eigenen Erfahrungen bestätigen. Des Weiteren benötigen StudentInnen für andere Projekte auch Räumlichkeiten. Das QPT2a von MMT beispielsweise muss in Teamarbeit erstellt werden (MMT 2010, S. 2), ist aber nicht als Scrum-Projekt eingestuft (vgl. 3.1.3). Durch die verpflichtende Teamarbeit müssen die betroffenen StudentInnen auch zusammensitzen und miteinander arbeiten bzw. kommunizieren. Dies geht in einem gemeinsamen Raum viel einfacher. Dadurch wird der effektive Raumbedarf erhöht.

Durch die dargelegten Faktoren (Raumbedarfsverminderung - Raumbedarfserhöhung) ist es schwierig, den tatsächlichen Raumbedarf und somit den „Soll-Zustand“ festzusetzen.

Der Autor stützt sich auf die Ausführungen aus der Literatur und setzt deshalb für eine optimale Lösung einen Raumbedarf von 35 Räumen fest. Jedes Projekt bekommt einen eigenen Raum und kann darin arbeiten. Ihm ist dabei bewusst, dass dies an der FHS zurzeit (Stand 29.06.2011) wohl kaum erfüllbar ist. Die optimale Lösung stuft er trotzdem so ein, da ein eigener Raum in der Literatur als so wichtig angesehen wird (vgl. Ausführungen von Pichler: (Pichler 2008, S. 17))

Für eine minimale Lösung sieht der Autor einen Raumbedarf von 18 Räumen an. Ein Scrum-Team sollte zumindest die Möglichkeit haben das „MMFly-Scrum“ Meeting durchführen zu können (vgl. 3.6.4) und zumindest die Hälfte der Woche in einem Projektraum zusammenarbeiten zu können.

Die „Sonstigen Räume“ sind ohne Utensilien wie beispielsweise Whiteboards oder Pinnwände ausgestattet. Hier macht eine Anschaffung dieser Utensilien, nach der Meinung des Autors, einen Sinn. In diesen Räumen wird hauptsächlich an Projekten gearbeitet und in der Folge werden die Utensilien unbedingt benötigt.

- Räume inklusive Ausstattung - Lösungsansatz

Der im Soll-Zustand festgelegte Raumbedarf der minimalen Lösung von 18 Räumen ist, nach der Meinung des Autors, bereits schwierig an der FHS bereit zu stellen (vgl. Punkt „Ist-Zustand“ mit 11 zur Verfügung stehenden Räumen). Die optimale Lösung sieht für jedes Projekt einen eigenen Raum vor und kommt auf einen Raumbedarf von 35 Räumen. Die optimale Lösung scheint somit noch weit schwieriger erfüllbar zu werden.

Damit die Räume fair und sinnvoll verteilt werden können, schlägt Gucher eine Art Bewerbung für einen Raum vor. Projekte, bei denen es grundsätzlich Sinn macht, im Scrum-Prozess zu arbeiten, könnten sich um einen Raum bewerben (vgl. A.1, Zeile 130). Die Raumbewerbungen könnten analysiert und dementsprechend Räume zugeteilt werden.

Der Autor würde des Weiteren vorschlagen, dass mehrere Projektgruppen in einem Raum arbeiten könnten. Hier ist eine richtige Zeiteinteilung und Abstimmung seitens der ProjektleiterInnen bzw. Scrum MasterInnen gefragt. Wenn diese die Meetings innerhalb mehrerer Projektgruppen richtig koordinieren, wäre eine Art „Schichtbetrieb“ möglich. Die eine Projektgruppe arbeitet beispielsweise von acht Uhr morgens bis zwölf Uhr mittags und die zweite Projektgruppe hat erst um dreizehn Uhr ihr Daily-Scrum Meeting.

Sollten trotz guter Zeiteinteilung und Koordination seitens der Projektverantwortlichen, zu wenige Räume zur Verfügung stehen, kann aus Sicht des Autors, auch zeitgleich in einem Raum gearbeitet werden. Wird hier zu Beginn ein Verhaltenskodex festgelegt (z.B. gewisse Lautstärke muss bei Gesprächen eingehalten werden, etc.) und alle Anwesenden halten sich an diesen Kodex, kann auch ohne Probleme gemeinsam mit unterschiedlichen Projektgruppen in einem Raum gearbeitet werden. Der Autor hat während seines QPT3s mit dieser Arbeitsweise gute Erfahrungen gemacht. Während dieser Zeit war des Öfteren der komplette Fachbereich „Web & Communities“ in einem Raum anwesend. In der Folge arbeiteten drei bis vier unterschiedliche Projektgruppen gemeinsam in einem Raum.

Sollte sich trotzdem nicht für jede Projektgruppe ein Raum ausgeben, wäre dies laut Randelshofer auch zu lösen. StudentInnen könnten ihre Scrum-Utensilien, wie z.B. ein Burndown-Chart, in einem Campus-Zimmer an der Wand aufhängen und dort die Meetings abhalten (vgl. A.2, Zeile 485).

Für die fehlenden Utensilien wie Whiteboards und Pinnwände wäre, nach der Meinung des Autors, eine Investition seitens der FHS am sinnvollsten. Diese Utensilien können für mehrere Projekte wieder verwendet werden. Dadurch macht eine Investition auch langfristig gesehen einen Sinn. Sollte die FHS eine solche Investition nicht tätigen können, könnten Studierende auf elektronische Tools ausweichen. Ein solches elektronisches Tool wird im Kapitel „Scrum - Artefakte - Situation an der FHS 3.5“ angesprochen. Dieses könnte von den StudentInnen verwendet werden.

### Computerunterstützung

- Computerunterstützung - Ist-Zustand

Zurzeit (Stand 17.06.2011) stehen drei Computer-Labore für Projekte zur Verfügung (MMA & MMT 2011a). Hierbei handelt es sich um die Labore, die bereits im vorigen Abschnitt „Räume“ angesprochen worden sind.

Die Computer in diesen Laboren sind zurzeit (Stand 28.06.2011) nur mit Windows-Betriebssystemen ausgestattet. Dies hat der Autor durch Erkundungen in Erfahrung gebracht. Hier würde, nach der Meinung des Autors, ein Dual-Boot mit beispielsweise einem Windows und einem Linux-Betriebssystem Sinn ergeben. Aus eigenen Erfahrungen weiß der Autor, das z.B. im Fachbereich „Web & Communities“ von MMT mit einem Linux-Betriebssystem sehr viel leichter entwickelt werden kann, als unter Windows.

Die bereits im Abschnitt „Räume“ angesprochene Verfügbarkeit dieser drei Computer-Labore ist nicht sehr hoch. Einzig das Game-Labor steht direkt für Projekte zur Verfügung. Die anderen Labore sind hauptsächlich für die Lehre vorhanden (vgl. 3.3.3, Abschnitt „Räume“, Punkt „Labore“).

Für die Entwicklung von Spielen im Fachbereich „Augmented Reality & Game“ von MMT steht zurzeit ein Asset-Server für etwaige 3D-Model-Dateien zur Verfügung.

- Computerunterstützung - Soll-Zustand

Es sind drei Computer-Labore vorhanden und es werden bis zu 35 Projekte im Scrum-Prozess durchgeführt. Diese Projekte haben alle einen hohen Software-Anteil, da ansonsten der Einsatz von Scrum wenig Sinn ergeben würde. In der Folge benötigen diese Projektgruppen Computer um arbeiten zu können. Wollen jetzt 35 unterschiedliche Scrum-Teams in drei Computer-Laboren ihre Projekte entwickeln, wird es schwierig. Zusätzlich stehen diese Labore hauptsächlich für die Lehre zur Verfügung (vgl. (MMA & MMT 2011a)). Da die meisten StudentInnen aber bereits über eigene Laptops verfügen, sieht der Autor in dieser Situation kein Problem. In den restlichen Räumen steht zusätzlich noch W-LAN zur Verfügung. StudentInnen können somit an ihren eigenen Rechnern arbeiten und benötigen nicht unbedingt eine Computerunterstützung seitens der FHS.

Der Autor findet die Anzahl an vorhandenen Computer-Laboren passend gewählt. Allerdings fehlt, nach der Meinung des Autors, ein Computer-Labor mit Macs. Es entwickeln und arbeiten viele StudentInnen mit Macs, deshalb sollte noch ein zusätzliches Labor mit Mac-Computern angeschafft werden.

Des Weiteren sollten die bestehenden Computer-Labore mit Dual-Boot ausgerüstet werden und über ein Windows- und ein Linux-Betriebssystem verfügen. Dadurch haben StudentInnen mehr Möglichkeiten und Freiheiten zwischen verschiedenen Betriebssystemen zu wählen. Dies könnte unter Umständen in gewissen Projekten ein sehr großer Vorteil sein.

- Computerunterstützung - Lösungsansatz

Die Computer-Labore verfügen über so genannte „Labor-ManagerInnen“. Der Job einer Labor-ManagerIn wird in der Regel durch Studierende besetzt. Diese sind dafür zuständig, dass die richtige Software auf den Computern installiert ist und führen deshalb gelegentlich Installationen oder Updates durch. Nach der Meinung des Autors könnten diese die Labore mit Dual-Boot ausstatten.

### Server und Services

- Server und Services - Ist-Zustand

Zurzeit (Stand 17.06.2011) steht der MMA+MMT-Projektserver für die Studierenden zur Verfügung (MMA & MMT 2011a). Dieser Server soll eine Kommunikationsplattform für MMA- und MMT-Studierende, LektorInnen und AbsolventInnen sein. Er dient als Archiv für Werke und Projekte. Des Weiteren bietet der MMA+MMT-Projektserver für die StudentInnen Infrastrukturen zur Entwicklung von Projekten. Es stehen Webspaces und Datenbanken zur Verfügung. Diese können auf Wunsch von den StudentInnen bestellt werden (MMA & MMT 2011e). Dieser Server wird seitens der Studiengänge MMA und MMT selbst gewartet und betreut. Dafür stehen zurzeit eine LektorIn sowie eine StudentIn (als Nebenjob) zur Verfügung. Es können auf Wunsch Webspaces mit PHP oder Rails, Wordpress, MySQL, etc. eingerichtet werden (vgl. (MMA & MMT 2011a) und vgl. (MMA & MMT 2011e)).

Des Weiteren gibt es an der FHS ein zentraler Server für git- und svn-Repositories. Hier können Projekte mit den Versionskontrollsystemen „git“ und „svn“ gespeichert bzw. abgelegt werden (MMA & MMT 2011a).

Unter „kirk.multimediatechnology.at“ sind viele virtuelle Server eingerichtet. Diese werden hauptsächlich für die Lehrveranstaltung „Web Operations“ verwendet. Des Weiteren werden sie auch noch für Unity Server verwendet (MMA & MMT 2011a).

Die Studiengänge MMT und MMA verfügen über weitere Ressourcen. Bei der Hostingplattform „Github“ steht für die FHS ein Organisationsaccount zur Verfügung. Dies ermöglicht Studierenden über diesen Account Projekte zu hosten und zu entwickeln.

Für größere Datenmengen verfügen die Studiengänge MMA und MMT über einen gemeinsamen Amazon S3 Account (MMA & MMT 2011a). Dieser Account wird beispielsweise für das Ablegen der Mediendateien des MMA- und MMT-Portfolios (Online zu finden unter: <http://portfolio.mediacube.at/>) verwendet. Dies kann der Autor aus eigenen Erfahrungen bestätigen, da er an der Entwicklung des Portfolios im Rahmen seines QPT3-Projektes, beteiligt gewesen ist. Der Amazon S3 Account nimmt dem MMA+MMT-Projektserver viel an Last ab.

- Server und Services - Soll-Zustand

An der FHS entwickeln die StudentInnen zunehmend mit Versionskontrollsystemen. Dies kann der Autor aus eigenen Erfahrungen bestätigen. In manchen Projekten ist die Abgabe in „git-“, „svn-“, oder anderen Repositories Pflicht. Dies wäre beispielsweise im QPT1 von MMT der Fall (siehe (MMT 2011a, S. 4)). Ein Repository-Server ist bereits vorhanden. Die Bestellung eines passenden Repositories ist Aufgabe der Studierenden. Diese müssen darauf achten, dass sie zu Projektbeginn bereits über ein Repository verfügen und sollten deshalb früh genug ein Repository bestellen.

An der FHS existiert noch kein Continuous Integration Server (Stand 05.05.2011). Bereiche des bereits angesprochenen MMA+MMT-Projektserver könnten zu einem Continuous Integration Server erweitert werden.

Der MMA+MMT-Projektserver kann, nach der Meinung des Autors, sehr gut als Production-Server verwendet werden. In Kombination mit dem Amazon S3 Account der Studiengänge

(Stichwort: große Datenmengen) sieht der Autor somit an der FHS den Production-Server als gegeben.

Als Staging-Server könnte der MMA+MMT-Projektserver ebenfalls eingesetzt werden.

Für die Wartung und Einrichtung von Repositories und Software werden Personen benötigt. Diese sind an der FHS jedoch bereits vorhanden (siehe Ist-Zustand: LektorIn, StudentIn (als Nebenjob)). Aus diesem Grund sieht der Autor hier keinen weiteren Personalbedarf.

Zusammengefasst kann festgehalten werden, dass an der FHS fast alle benötigten Servertypen und Services vorhanden sind. Einziges Problem könnte werden, dass sich das meiste auf dem MMA+MMT-Projektserver befindet und dieser überlastet wird. Durch Auslagerungen auf den Organisationsaccount Github und den Amazon S3 Account kann die Last ein wenig vom MMA+MMT-Projektserver abgewälzt werden. Durch die steigende Anzahl an Projekten und Archivmaterial wird der Server aber zunehmend ausgelastet.

Der Autor sieht in einer Anschaffung eines zweiten Servers, mit den gleichen Eigenschaften wie der MMA+MMT-Projektserver, trotz der zunehmenden Auslastung, wenig Sinn. Nach der Meinung des Autors ist ein Ausbau des bestehenden Servers besser, als eine Neuanschaffung. Hier teilt der Autor die Meinung von den MMA und MMT Verantwortlichen. Laut diesen muss hier langfristig gedacht werden (vgl. (MMA & MMT 2011e), Abschnitt „Ziele“, Punkt „Prinzipien“.

- Server und Services - Lösungsansatz

Der noch fehlende Continuous Integration Server an der FHS könnte, wie bereits angesprochen, in den MMA+MMT-Projektserver inkludiert werden. Nach der Meinung des Autors, könnte beispielsweise die StudentIn, welche bei der Wartung der Server-Infrastruktur mithilft, auf dem Projektserver einen „Hudson“-Server einrichten. Ein „Hudson“-Server ist ein Opensource Continuous Integration Server. (Rödig 2010, S. 13)

Um die generelle hohe Last vom MMA+MMT-Projektserver wegzubekommen, sollte nach der Meinung des Autors der Github-Account der FHS besser ausgenutzt werden. Nach den Erkenntnissen des Autors existiert dieser Github-Account seit knapp einem Semester und ist noch nicht in reger Benutzung. Vereinzelt LehrveranstaltungsleiterInnen hosten Übungsprojekte auf diesem Account. Ein tatsächliches Studentenprojekt ist, zum derzeitigen Stand (28.06.2011), noch nicht auf diesem Account gehostet.

Des Weiteren könnten Studierende eigene Github-Accounts eröffnen und ihre Projekte dort hosten. BenutzerInnen stehen auf Github bis zu 300 MB Speicherplatz gratis zur Verfügung. Dies kann der Autor aus eigenen Erfahrungen bestätigen. Einzige Voraussetzung für den gratis Speicherplatz ist die Veröffentlichung des Programm-Codes. Die Projekte müssen somit öffentlich sichtbar sein. Dies stellt für den Autor allerdings kein Problem dar. Manche Projekte müssen sowieso unter einer Opensource Software Lizenz veröffentlicht werden. Dies ist beispielsweise beim QPT2b von MMT Pflicht (siehe (MMT 2011b, S. 6).

Eine weitere Möglichkeit zur Entlastung des MMA+MMT-Projektserver wäre der Einsatz von „Heroku“. Heroku ist eine Cloud-Applikation für Ruby. D.h. es können Ruby on Rails Applikationen über diese Plattform ins Netz gestellt werden (Heroku 2011b). Diese Plattform ist generell kostenpflichtig. Es gibt allerdings die Möglichkeit kleinere Applikationen (max. 5 MB groß) gratis online zu stellen. Laut dem Anbieter ist dies für Staging- oder Testing-Vorgänge gedacht (Heroku 2011a). Für den Autor könnte dies zur Folge haben, dass der MMA+MMT-Projektserver im Staging-Bereich entlastet werden könnte.

Sollte der MMA+MMT-Projektserver tatsächlich vollkommen ausgelastet sein, würde der Autor als Notfallsituation folgendes vorschlagen: Es gibt im Internet sehr viele Webspaces-Gratis-Anbieter (z.B. „Funpic“ (Funpic 2011)). Diese bieten meistens Webspaces, FTP-Zugang, PHP und MySQL-Datenbanken. In Ausnahmefällen könnten StudentInnen auf solche Gratis-Anbieter ausweichen. Der Autor sieht diese Möglichkeit allerdings nur in Ausnahmefällen und wenn sonst keine andere Möglichkeit vorhanden ist. Die Gratis-Anbieter werden oft durch

Werbung finanziert und schalten diese automatisch auf die veröffentlichten Projekte. Dies wirkt, nach der Meinung des Autors, unprofessionell und für EndnutzerInnen ist die Werbung mitunter sehr störend.

### 3.4 Scrum - Hilfreiches - Situation an der FHS

Hilfreiche Tools wie PP, TDD und SDD wurden bereits vorgestellt (siehe 2.4).

Die Situation an der FHS bezüglich dieser Tools ist folgende:

MMT-Studierende des Fachbereichs „Web & Communities“ bekommen in Lehrveranstaltungen wie „Web-Technologies“ bereits Einblicke in TDD oder SDD. Dies kann der Autor aus eigener Erfahrung bestätigen. Diese Einblicke und eventuelle Erfahrungen aus dem Berufspraktikum, während des 5. Semesters, könnten beim Einsatz von Scrum, beispielsweise in einem QPT3-Projekt, hilfreich sein. Im Fachbereich „Augmented Reality & Game“ ist dem Autor aus Erzählungen von Studierenden dieses Fachbereichs bekannt, dass diese hilfreichen Tools noch nicht gelehrt bzw. eingesetzt wurden. Hier wäre es sicherlich sinnvoll, beide Fachbereiche gleich zu behandeln und die gleiche Art von Wissen weiter zu geben. Seitens des Studiengangs MMA ist dem Autor der Einsatz von solchen hilfreichen Tools nicht bekannt. Hier stellt sich nur die Frage, ob im Designbereich überhaupt solche Tools notwendig sind. Beispielsweise bei der Entwicklung eines Audiotracks machen Tests weniger Sinn. Bei der Entwicklung eines Frontends für eine Website, mit JavaScript und CSS, machen beispielsweise Integrationstests sehr wohl einen Sinn und es könnte TDD eingesetzt werden.

### 3.5 Scrum - Artefakte - Situation an der FHS

Dieses Unterkapitel beschäftigt sich mit der Situation an der FHS betreffend Scrum Artefakten (siehe 2.5).

Die Artefakte an und für sich können im Rahmen der FHS in Scrum-Projekten ohne Einschränkungen zum Einsatz kommen. Einziges Problem ist die Findung eines passenden Ortes für die Artefakte. Studierende müssen geeignete Plätze zur Anbringung der Artefakte finden. Dieser Punkt hat somit wieder mit der Voraussetzung des Vorhandenseins einer passenden Infrastruktur zu tun.

Randelshofer schlägt für dieses Problem die Verwendung von elektronischen Tools vor (vgl. A.2, Zeile 479). Zum jetzigen Zeitpunkt (Stand 04.05.2011) würde dafür ein passendes Tool namens Pivotaltracker zur Verfügung stehen. Mithilfe dieses Tools können sogar alle vier Artefakte (Product Backlog, Sprint Backlog, Release-Burndown und Sprint-Burndown) in elektronischer Form geführt und abgebildet werden. Dieses Tool steht, zum jetzigen Zeitpunkt, gratis zur Verfügung. Das Unternehmen Pivotal Labs Inc., welches den Pivotaltracker entwickelt, möchte in Zukunft für die Benutzung Geld verlangen. Für Bildungseinrichtungen, wie die FHS, bleibt das Tool jedoch weiterhin gratis. (Sepulveda 2011). Außer dem Pivotaltracker besteht auch die Möglichkeit, die Scrum-Artefakte beispielsweise in Form von Excel-Tabellen zu führen. Diese Excel-Tabellen könnten, zum Beispiel in einem geteilten Dropbox-Ordner, für jeden zugänglich gemacht werden. Sind die infrastrukturellen Voraussetzungen geschaffen, können die Artefakte auch analog geführt werden.

Der Autor kann aus eigenen Erfahrungen sagen, dass das Führen von elektronischen Artefakten seine Vorteile gegenüber analogen hat. Bei einem Scrum-Team mit StudentInnen aus verschiedenen Bereichen Österreichs und dem Ausland ist, beispielsweise zu unterrichtsfreien Zeiten, die Flexibilität von elektronisch geführten Artefakten höher. Jede/jeder kann, von jedem beliebigen Ort aus, auf die Artefakte zugreifen und es kann auch, bei Abwesenheit von der FHS, weiterentwickelt werden.

Dieser Meinung bzw. Ansicht gegenüber stehen die Ausführungen von Scrum Artefakten gemäß der Literatur. Die Literatur besagt, dass Scrum in einem Raum stattfinden soll und alle dafür benötigten Rahmenbedingungen dafür geschaffen werden müssen, dazu zählen auch die Artefakte. Schwaber und Sutherland empfehlen in ihren Ausführungen die Verwendung von analogen Artefakten. Die Artefakte sollen händisch verfasst und großflächig an den Bürowänden aufgehängt werden. Scrum-Teams nehmen große und für jeden sichtbare Artefakte an den Bürowänden besser auf, als beispielsweise Excel-Tabellen (Schwaber & Sutherland 2010, S. 19).

In welcher Form nun die Artefakte an der FHS geführt werden sollten, ist nach der Meinung des Autors, stark von den infrastrukturellen Gegebenheiten abhängig. Ist ein Projektraum vorhanden, macht ein analoges Führen Sinn. Steht dem Scrum-Team kein Raum zur Verfügung ist das elektronische Format zu bevorzugen.

Generell ist noch festzuhalten, dass ein analoges Product Backlog bei Bedarf auch in ein elektronisches Tool überführt werden kann. Dies ist zwar mit Arbeit verbunden, kann aber an der FHS auch Sinn machen. Steht beispielsweise eine unterrichtsfreie Zeit bevor, wird das Product Backlog ins elektronische Format übertragen. Nach der unterrichtsfreien Zeit wird der analoge Stand angepasst. Dadurch ist ein guter Übergang geschaffen und es kann auch unabhängig von den Räumlichkeiten an der FHS weiterentwickelt werden.

### 3.6 Scrum - Timeboxen - Situation an der FHS

Dieses Unterkapitel beschäftigt sich mit der Situation an der FHS betreffend Scrum Timeboxen (siehe 2.6). Können alle Meetings gemäß Literatur stattfinden? Welche Änderungen müssen gegebenenfalls gemacht werden? Welche zusätzliche Personen sollten an den verschiedenen Meetings teilnehmen? Wie sollen die Meetings gestaltet werden? Laut Randelshofer ist es generell möglich, im Rahmen der FHS alle Scrum Timeboxen durchzuführen (vgl. A.2, Zeile 309). Es müssen jedoch teilweise Modifikationen stattfinden.

#### 3.6.1 Release-Planning

Das Release-Planning sollte, aus der Sicht des Autors, in die Konzeptions-Phase bzw. Pre-Production Phase der Projekte fallen. Hier werden erste Überlegungen und Ideen aus verfassten Exposé konkretisiert und weiter aufgearbeitet. Aus diesem Grund sieht der Autor dieses Meeting eher als ein Meeting über eine längere Phase hin an, als über ein klares Meeting von beispielsweise fünf Stunden an einem Nachmittag. An diesem Meeting sollen alle Interessengruppen teilnehmen. Darunter fällt das Scrum-Team inklusive der betreuenden Lehrperson. Handelt es sich um ein fachübergreifendes Projekt, sollten betreuende Lehrpersonen aller Fachbereiche teilnehmen.

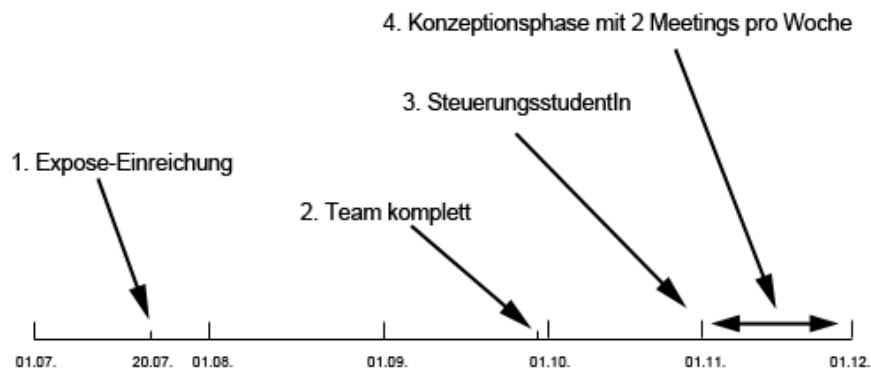


Abbildung 5: Beispiel Release-Planning

Die Abbildung 5 zeigt ein Release-Planning an der FHS am Beispiel eines QPT3-Projekts im Studienjahr 2010/11. Zuerst muss von den StudentInnen ein Exposé abgegeben werden. In diesem Beispiel handelt es sich um ein Exposé für ein QPT3-Projekt. Es wird am 20.07.2010 abgegeben. Die Studiengangleitung akzeptiert das Exposé und somit die Projektidee. Die StudentIn, welche das Exposé verfasst hat, macht sich auf die Suche nach Projekt-Mitgliedern. Am 30.09.2010 steht das komplette Team. Es setzt sich mit Studierenden von MMA und MMT zusammen. Das Team möchte das Projekt gerne im Scrum-Prozessablauf umsetzen und entscheidet sich deshalb am 01.11.2010 für die Aufnahme einer Master-StudentIn aus dem MMA-Schwerpunkt „Steuerung“. Das Projekt wird von Lehrpersonen der jeweiligen Fachbereiche betreut. Ab dem 01.11.2010 beginnt die Konzeption bzw. Pre-Production Phase des Projektes. Diese dauert bis zum 30.11.2010. Innerhalb dieses Abschnittes finden zwei gemeinsame Meetings pro Woche mit allen Projekt-Mitgliedern und den beiden Betreuungspersonen statt. In diesen Meetings wird grob festgelegt, was am Ende der Projektentwicklungszeit für ein Ergebnis herauskommen soll. Mit Ende dieser Phase ist das Release-Planning beendet. Die Zeiträume und Daten zu diesem Beispiel sind an den QPT3-Leitfaden angelehnt (vgl. (MMA & MMT 2011d))

Voriges Beispiel soll die Ansicht des Autors verdeutlichen. Das Release-Planning besteht somit an der FHS aus mehreren, regelmäßig stattfindenden Meetings. Wichtig ist, dass zu Beginn dieser Phase, seitens des Scrum-Teams, bereits alle Scrum Rollen eingeteilt und festgelegt worden sind. Somit kann jede/jeder einzelne bereits vor Beginn der eigentlichen Umsetzung seine Rolle „lernen“.

Diese Lösung der Umsetzung des Release-Plannings ist an Larman's Ansichten angelehnt (vgl. 2.6.1). Dieser bezeichnet das Release-Planning als Pre-game Planning und Staging. Aus diesem Pre-game Planning hat der Autor die Pre-Production Phase abgeleitet.

### 3.6.2 Sprint-Planning

Das Sprint-Planning kann fast zu 100% gemäß der Literatur übernommen werden. Das komplette Scrum-Team muss an diesem Meeting teilnehmen. Der Autor würde zusätzlich bei den ersten zwei bis drei Sprints eine außenstehende, Scrum-erfahrene Person zur Beobachtung hinzufügen. Diese sollte sich inhaltlich nicht einmischen, aber der eigentlichen Scrum MasterIn zur Seite stehen und eventuell bei der Moderation unter die Arme greifen. Anfangs ist dies für eine neue Scrum MasterIn sicherlich nicht so einfach und es könnten bereits beschriebene Fehler auftreten (vgl. 2.6.2).

### 3.6.3 Sprint

Ein Sprint kann grundsätzlich fast zu 100% gemäß der Literatur übernommen werden. An der FHS gibt es jedoch Hindernisse und Ablenkungen, welche bereits im Punkt 3.1.6 angesprochen wurden. Können Scrum-Team-Mitglieder diese Hindernisse mit guten Einteilungen und richtiger Priorisierung umgehen, können Sprints gemäß der Literatur abgehalten werden.

### 3.6.4 Daily-Scrum

Das Daily-Scrum Meeting ist laut Randelshofer das einzige Meeting welches, im Rahmen der FHS, schwierig durchzuführen ist (vgl. A.2, Zeile 316ff). Hier spielen die unterschiedlichen Stundenpläne von Scrum-Team-Mitgliedern eine Rolle. Es ist nicht jede/jeder täglich um die gleiche Zeit anwesend. Bei Projekten mit Studierenden aus mehreren Fachbereichen und beiden Studiengängen (MMA und MMT) wird dies noch aufwendiger zu organisieren. Randelshofer schlägt zur Lösung dieses Problems eine Modifikation dieses Meetings vor. Laut Literatur sollte es täglich abgehalten werden, Randelshofer schlägt aber ein „MMFly-Scrum“ Meeting vor. D.h. das Daily-Scrum Meeting wird nur jeden Montag, Mittwoch und Freitag abgehalten. So verringert sich die Anzahl der Tage, an denen alle Scrum-Team-Mitglieder an der FHS anwesend sein müssen. Trotz dieser Verringerung erfahren Mitglieder im 2-Tages-Rhythmus den aktuellen Stand der anderen. Die Transparenz, welche durch das Daily-Scrum Meeting entsteht, ist dadurch zwar eingeschränkt, doch der Autor sieht diesen Kompromiss als sinnvoll an. Durch den 2-Tages-Rhythmus können Studierende neben dem Scrum-Projekt noch andere anstehende Arbeiten erledigen und sich die Zeit wahrscheinlich besser einteilen. Kann eine ProgrammiererIn trotzdem nicht am Meeting teilnehmen, könnte sie per Skype oder anderen Chat-Programmen, auch in Abwesenheit, am Meeting teilhaben. Hierbei eignen sich elektronische Formen der Scrum Artefakte gut, da auch die über Chatprogramme dem Meeting zugefügten Scrum-Team Mitglieder, am eigenen Rechner, den aktuellen Stand abfragen können. Dadurch wird die externe Teilnahme zusätzlich noch erleichtert.

Am Daily-Scrum Meeting nehmen, gemäß Literatur, alle Scrum-Team-Mitglieder teil. Die Scrum MasterIn hat während des Daily-Scrum Meetings die Aufgabe dieses zu moderieren (siehe 2.6.4). Der Autor würde anfangs zusätzlich noch eine mit Scrum erfahrene Person zur Unterstützung der Scrum MasterIn hinzufügen. Diese könnte bei Problemen betreffend der richtigen Moderation die Scrum MasterIn unterstützen.

### 3.6.5 Sprint Review

Das Sprint Review kann zu 100% gemäß der Literatur übernommen werden. Die Literatur besagt, dass an diesem Meeting das komplette Scrum-Team und Stakeholder teilnehmen (siehe 2.6.5).

Der Autor würde es als sinnvoll ansehen, wenn das Sprint Review im Rahmen von Begleitenden Lehrveranstaltungen zu den Projekten (z.B. bei einem QPT3-Projekt in der Lehrveranstaltung „Interdisziplinäre multimediale Qualifikation“) stattfindet. Führen StudentInnen des gleichen Jahrgangs beispielsweise mehrere Projekte mit Scrum durch, könnten die Längen der Sprints der unterschiedlichen Projekte, aufeinander abgestimmt werden. Endet bei mehreren Projekten ein Sprint zugleich, steht zum gleichen Zeitpunkt das Sprint Review an. Die Sprint Reviews könnten an ein und demselben Tag, in den gleichen Räumlichkeiten, an der FHS, durchgeführt werden. Dadurch sind bei allen Sprint Reviews zum einen die Scrum-Teams anwesend, und zum anderen aber auch MitstudentInnen und Betreuungspersonen. MitstudentInnen und Betreuungspersonen können den präsentierenden Scrum-Teams Feedback geben und damit in die Rollen der Stakeholder schlüpfen.

Zusätzlich ist der Autor der Meinung, dass eine Teilnahme von FachspezialistInnen, beim Sprint Review, notwendig ist. Unter FachspezialistInnen wäre beispielsweise bei einem Ruby-on-Rails Projekt, die Lehrperson, welche den Studierenden die dementsprechenden Lehrveranstaltungen mit dem dementsprechenden fachlichen Input liefert, gemeint. Wird innerhalb eines Sprints zum Beispiel ein komplizierter mathematischer Algorithmus in ein Projekt eingebaut, wäre die Teilnahme einer Mathematik-LektorIn von Vorteil. Diese FachspezialistInnen können eventuell Feedback zu ganz speziellen Bereichen des Sprint-Ergebnisses geben.

### 3.6.6 Sprint Retrospektive

Die Sprint Retrospektive ist ein wichtiges Werkzeug zur Verbesserung des Scrum-Prozessablaufs (vgl. 2.6.6). Aus diesem Grund ist der Autor der Meinung, dass dieses Meeting im Rahmen der FHS in den ersten zwei bis drei Sprints von einer erfahrenen Scrum MasterIn geleitet werden sollte. Hat die eigentliche Scrum MasterIn eines Projektes diese Erfahrungen noch nicht, so sollte eine externe Person hinzugezogen werden. Nur die unterstützende Hilfe einer externen Person könnte hier, nach Ansichten des Autors, zu wenig sein. Anfangs sollte die externe Person das Meeting leiten, nach zwei bis drei Sprints sollte das Zepter an die eigentliche Scrum MasterIn des Teams übergeben werden. Die externe Person sollte zur Unterstützung der Moderation, aber trotzdem beim Meeting anwesend sein.

Der Autor ist dieser Meinung, da die Verbesserung des Prozessablaufs im Vordergrund steht. Laut Ausführungen von Derby und Larsen gibt es alleine für die Datensammlung während einer Sprint Retrospektive schon acht verschiedene Varianten wie vorgegangen werden kann. (Derby & Larsen 2006, S. IX) Dies könnte bei Scrum MasterInnen-Anfänger zu einer Überforderung führen.

## 4 Fahrtenplan

Dieses Kapitel beantwortet die Forschungsfrage mit einer Aufbereitung der Ergebnisse aus den Erkenntnissen aus dem Kapitel 3 „Scrum an der Fachhochschule Salzburg“. Die Aufbereitung wird in Form eines Fahrtenplans ausgedrückt. Dieser soll zukünftigen StudentInnen als Grundlage dienen, wie vorgegangen werden sollte, um Scrum in Projekten der Studiengänge MMA und MMT, an der FHS, sinnvoll einsetzen zu können. Zusätzlich soll er für die Studiengangleitung und verantwortlichen Personen seitens der beiden Studiengänge eine Palette an Vorschlägen mitbringen. Aus diesen Vorschlägen sollte hervorgehen, welche nötigen Schritte noch vollzogen werden müssen, um Scrum sinnvoll an der FHS einsetzen zu können.

Der Autor teilt den Fahrtenplan in eine „ideale“ bzw. „teure“ und in eine „mittlere“ bzw. „Kompromiss“-Variante ein. Die Grundvoraussetzungen für diese beiden Varianten sind jedoch die selben und werden deshalb im folgenden Abschnitt festgehalten:

### Grundvoraussetzungen

- „Projekttyp durch Ausbildungsschwerpunkte“: Das bearbeitete Projekt sollte einen hohen Software-Anteil beinhalten. Dies ist wichtig für einen sinnvollen Einsatz von Scrum als Projektmanagement-Prozess (vgl. 3.1.1). Es sollte vor dem Projektstart überprüft werden, ob das Projekt einen hohen Software-Anteil hat. Hat es beispielsweise einen hohen Software-Anteil, diesen jedoch erst ab dem vierten Monat des Projekt-Verlaufes, kann so geplant werden, dass ab dem vierten Monat im Scrum-Prozess gearbeitet wird.
- Projektumfang: Das Projekt sollte den richtigen Projektumfang haben (vgl. 3.1.3). Unter passenden Projekten versteht der Autor folgende: QPT2 (MMA), QPT2b (MMT), QPT3 (MMA und MMT) und Masterprojekte (MMA und MMT).
- „Projekttyp nach Abläufen“: Der Typ des Projektes kann einer von allen im Punkt „Projekttypen nach Abläufen 3.1.2“ aufgezählten Punkte einnehmen. Dies funktioniert allerdings nur unter der Voraussetzung, dass Gegebenheiten, wie ein hoher Software-Anteil und der richtige Projektumfang, gesetzt sind.

### 4.1 Ideale Variante

Dieses Unterkapitel beinhaltet die ideale Variante des Fahrtenplan wie vorgegangen werden kann, um Scrum sinnvoll an der FHS einsetzen zu können. Die folgenden Punkte ergeben nur einen Sinn, wenn die Grundvoraussetzungen, welche im vorigen Abschnitt erläutert wurden, eingehalten werden bzw. gegeben sind.

#### 1. Besetzung von Scrum-Rollen:

##### (a) Scrum MasterIn:

Die Rolle der Scrum MasterIn sollte von einer MMA bzw. MMT MasterstudentIn mit Schwerpunkt „Steuerung“ bzw. „LEAD“ besetzt werden. Diese StudentIn sollte diese Rolle übernehmen wollen und hat bereits dementsprechende Grundkenntnisse und Vorwissen in Scrum und im Projektmanagement. Zur Unterstützung sollte trotzdem noch eine erfahrene Scrum MasterIn zur Verfügung stehen. Diese sollte anfangs beispielsweise die Sprint Retrospektive leiten. Des Weiteren sollte sie eine unterstützende Position einnehmen. Sie sollte, der vom Team gewählten Scrum MasterIn, beim Umsetzen der Pflichten einer Scrum MasterIn (vgl. 2.2.1) helfen. Im Speziellen sollte sie dabei behilflich sein, dem Team Selbstorganisation beizubringen. Der Prozessablauf und die Weiterentwicklung des Teams stehen im Vordergrund, deshalb sollte die Scrum MasterIn anfangs weniger Programmierarbeit übernehmen. Sie sollte darauf achten bzw.

sich darauf konzentrieren, dass die Selbstorganisation und der Umgang mit Scrum als Projektmanagement-Werkzeug im Team funktionieren. Weitere Aufgabenbereiche sind die Beobachtung der aktiven Mitarbeit aller Mitglieder sowie jederzeit für Fragen zu Verfügung zu stehen.

(b) Product OwnerIn:

Die Rolle der Product OwnerIn sollte von einer MMA bzw. MMT MasterstudentIn mit Schwerpunkt „Steuerung“ bzw. „LEAD“ besetzt werden. Die Wahl der Rollenbesetzung sollte innerhalb des kompletten Scrum-Teams einstimmig passieren. Die gewählte Person sollte hinter dem Projekt stehen. Sie sollte acht geben, dass die Projekt-Vision von allen Team-Mitgliedern eingehalten wird (vgl. 3.2.2). Tägliche Entscheidungen sollten von dieser Person getroffen werden. Für größere Entscheidungen kann die Product OwnerIn ein Meeting einberufen. An diesem Meeting sollten alle Scrum-Team-Mitglieder teilnehmen. Dieses Meeting dient zur Diskussion und Einbringung aller Ideen innerhalb des Teams. Die endgültige Entscheidung sollte jedoch die Product OwnerIn treffen (vgl. 3.2.2). Die Kommunikation der Product OwnerIn sollte für jeden „Projekttyp nach Abläufen“ dementsprechend ausgeführt werden (vgl. 3.2.2, Abschnitt „Kommunikation“).

(c) Entwicklerteam:

Das Entwicklerteam sollte durch unterschiedliche, für das Projekt benötigte, StudentInnen mit dementsprechend benötigten Fähigkeiten zusammengestellt werden. Die ideale Variante sieht vor, dass alle Mitglieder des Entwicklerteams teamfähig sind. Des Weiteren wäre von Vorteil, wenn einige StudentInnen des Entwicklerteams bereits Erfahrungen im Scrum-Prozess gemacht haben. Hier könnten beispielsweise Erfahrungen aus den Berufspraktiken eine positive Auswirkung haben. StudentInnen, welche noch keine Erfahrungen sammeln konnten, können von Beginn weg von den erfahreneren Projekt-Mitgliedern lernen. Dadurch kann sich die Selbstorganisation des Teams, von Anfang an, schneller weiter entwickeln und es können in der Folge bessere Projektergebnisse erzielt werden.

(d) Sonstige Rollen:

Die Rolle der „Chickens“ nehmen am besten MitstudentInnen des gleichen Jahrgangs und StudentInnen jüngerer oder älterer Jahrgänge des selben Studiengangs ein (vgl. Sonstige Rollen an der FHS 3.2.4 Abschnitt „Chickens“). Wer die Stakeholder-Rolle einnimmt, hängt stark vom „Projekttyp nach Abläufen“ ab. Dies wurde bereits im Punkt „Sonstige Rollen an der FHS 3.2.4, Abschnitt Stakeholder“ diskutiert. In jedem Fall wird die Stakeholder-Rolle von den direkt betreuenden und schlussendlich für die Projekt-Bewertung zuständigen Lehrpersonen eingenommen. Zusätzliche Stakeholder sind hauptsächlich vom „Projekttyp nach Abläufen“ abhängig (vgl. 3.1.2).

2. Scrum-Voraussetzungen schaffen:

(a) Teamgröße:

In der Literatur wird eine Teamgröße von fünf bis neun Personen als optimal angesehen (vgl. 2.3.1). Welche Gegebenheiten an der FHS zur Erreichung einer solchen Teamgröße erfüllt werden sollten, wurde bereits im Punkt „Teamgröße an der FHS 3.3.1“ diskutiert. Aus der dortigen Diskussion erreichen QPT2- (MMA) bzw. QPT2b- (MMT), QPT3- und Master-Projekte die benötigte Teamgröße. Somit sollte es sich beim Projekt, welches mit Scrum umgesetzt werden soll, um ein solches Projekt handeln.

(b) Teammitglieder & Teamfähigkeit:

Beim Projektstart sollte dem kompletten Team erstmal Grundsätzliches über den Scrum-Prozessablauf erläutert werden. Dies sollte am besten bereits vor dem Projektstart getan werden, sodass sich alle darauf einstellen können. Hat sich ein Team für den Einsatz von Scrum entschieden, sollte die Selbstorganisation des Teams in den Mittelpunkt gerückt werden. Anfangs nicht-teamfähige Personen können sich dadurch besser einstellen

und die Teamfähigkeit erlernen. Scrum fordert und fördert die Teamfähigkeit. Durch diesen Prozess wird das komplette Projektteam zusammengeschweißt und der Wissensaustausch und die Zusammenarbeit können dadurch besser funktionieren und vor allem laufend verbessert werden.

- (c) Infrastruktur Es sollte für jedes Scrum-Projekt in der idealen Variante ein Projektraum zur Verfügung stehen. Des Weiteren sollten in diesem Raum benötigte Utensilien wie Whiteboards, Pinnwände, etc. vorhanden sein (siehe 3.3.3, Abschnitt „Räume inklusive Ausstattung“).

Jede Person des Scrum-Teams sollte die Möglichkeit haben, an einem passenden Computer zu arbeiten. D.h. eine Mac-Entwicklerin sollte an der FHS die Möglichkeit haben, an einem Mac-Computer zu arbeiten. Dies sollte ebenso für eine Linux- oder eine Windows-UserIn gelten. Dies ist natürlich eine sehr große Investition für die FHS. In der idealen Variante sollte die Entwicklungsumgebung jedoch perfekt auf die jeweilige EntwicklerIn angepasst bzw. ausgerichtet sein.

Für alle Projekte sollten die passenden Server-Strukturen gegeben sein. D.h. es sollte ein Continuous Integration Server, ein Staging-Server und ein Production-Server zur Verfügung stehen. Für diese Server-Typen könnte der MMA+MMT-Projektserver verwendet werden (vgl. 3.3.3, Abschnitt „Server und Services“). Des Weiteren sollte das Scrum-Team die Möglichkeit haben, mit Versionskontrollsystemen, wie beispielsweise „git“, oder „svn“, zu arbeiten. In der Folge sollte ein Repository-Server zur Verfügung stehen. Die einzelnen Server sollten mit der Software, welche die Projekte benötigen, ausgestattet sein.

### 3. Scrum-Artefakte:

Die Scrum-Artefakte sollten in der idealen Variante analog geführt werden. D.h. einer Projektgruppe steht ein Projektraum zur Verfügung. In diesem Projektraum hängt an der Wand eine Pinnwand mit dem aktuellen Sprint-Burndown. Zusätzlich zum analogen Führen der Artefakte empfiehlt der Autor die Daten in gewissen Situationen (z.B. bevorstehende unterrichtsfreie Zeit von zwei Wochen) ins digitale Format zu übertragen (vgl. 3.5).

Um eine solche Übertragung durchführen zu können, sollten bereits im Vorhinein Vorkehrungen getroffen werden. Es sollte ein Tool für die elektronische Erfassung zur Verfügung stehen. Im Punkt „Scrum - Artefakte - Situation an der FHS 3.5“ wurde bereits ein solches Tool angesprochen und vorgestellt. Ist ein passendes Tool vorbereitet, sollte die Übertragung ins elektronische Format frühzeitig passieren. Hier ist seitens der Studierenden zu beachten, dass beide Formate auf demselben Stand bleiben sollten. Dies sollte zumindest bis zum Zeitpunkt „X“ so sein. Ab dem Zeitpunkt „X“ arbeitet jeder des Teams sowieso nur mehr mit der elektronischen Variante. Zeitpunkt „X“ könnte beispielsweise der Tag vor einer zweiwöchigen unterrichtsfreien Zeit sein. Nach der unterrichtsfreien Zeit darf nicht darauf vergessen werden, den analogen Stand wieder anzupassen.

### 4. Scrum-Timeboxen:

- (a) Release-Planning:

Das Release-Planning sollte als Meeting über eine längere Phase hinweg geführt werden (vgl. 3.6.1). Es sollten pro Woche zwei Meetings abgehalten werden in denen das Projekt besprochen wird. An diesen Meetings sollten neben dem kompletten Scrum-Team auch betreuende Lehrpersonen teilnehmen. Das gesamte Release-Planning sollte idealerweise in der Konzeptions- bzw. Pre-Production Phase des Projektes stattfinden. So haben Studierende noch genügend Zeit, eine Grundidee aus einem ersten Exposé, gemeinsam mit den betreuenden Lehrpersonen, weiter zu entwickeln. Hier sollten technische Grundsätze genauso angesprochen werden, wie die spätere Vermarktung des Produktes. Ein zusätzlicher Workshop-Tag zu Beginn der Konzeptions-Phase würde den Prozess sicherlich vorantreiben.

## (b) Sprint-Planning:

Das Sprint-Planning sollte gemäß der Literatur durchgeführt werden (vgl. 2.6.3). Zusätzlich zum Scrum-Team sollte anfangs zur Unterstützung der Moderation der Scrum MasterIn eine erfahrene Scrum MasterIn am Sprint-Planning teilnehmen.

## (c) Sprint:

Ein Sprint sollte gemäß der Literatur durchgeführt werden. Etwaige Hindernisse und Ablenkungen sollten seitens der Studierenden durch eine richtige Planung und Priorisierung umgangen werden können. Für größere Ablenkungen, wie beispielsweise eine Phase in der jede/jeder des gesamten Scrum-Teams eine Bachelor- oder Masterarbeit zu schreiben hat, sollten von Anfang an Zeitpuffer einkalkuliert werden. Dadurch kann vermieden werden, dass das Projekt aus dem Ruder läuft, Abgabefristen verstreichen oder das gesamte Scrum-Team den Überblick über noch bevorstehende Arbeiten verliert. Ein Sprint sollte idealerweise in den gemäß der Literatur dafür vorgesehenen Räumlichkeiten stattfinden.

## (d) Daily-Scrum:

Optimal wäre es, wenn das Daily-Scrum gemäß der Literatur durchgeführt werden könnte. Aufgrund von bereits erläuterten Problemen ist das an der FHS leider nicht möglich (vgl. 3.6.4). Aus diesem Grund wäre die Modifikation in ein „MMFly“-Scrum Meeting die ideale Lösung dieses Meeting im Rahmen der FHS abzuhalten („MMFly“-Scrum vgl. 3.6.4). An diesem Meeting sollten alle Scrum-Team-Mitglieder teilnehmen. Ist eine/einer einmal verhindert, sollte die Teilnahme beispielsweise über Skype oder andere Chat-Programme stattfinden. Dadurch sind trotzdem alle Mitglieder beim Meeting dabei. Sollte eine/einer einmal einfach überhaupt nicht am Meeting teilnehmen können, dann ist das nun mal so. In der realen Wirtschaft gibt es auch Fälle, bei denen ProgrammiererInnen, beispielsweise aufgrund von Krankheit, das Daily-Scrum Meeting ausfallen lassen müssen.

## (e) Sprint Review:

Das Sprint Review sollte gemäß der Literatur durchgeführt werden. An diesem Meeting sollten neben dem Scrum-Team noch die passenden Stakeholder (vgl. 3.2.4, Abschnitt „Stakeholder“) teilnehmen. EndnutzerInnen als Stakeholder könnten beispielsweise von MitstudentInnen dargestellt werden. Des Weiteren ist die Teilnahme von Fachspezialisten am Sprint Review ideal. Diese könnten den StudentInnen fachliches Feedback und Tipps liefern, sodass diese besser weiterarbeiten können. Das Sprint Review sollte im Rahmen einer begleitenden Lehrveranstaltung zum jeweiligen Projekt stattfinden (vgl. 3.6.5).

## (f) Sprint Retrospektive:

Die Sprint Retrospektive sollte gemäß Literatur durchgeführt werden. Einziger Unterschied zur Literatur ist dabei, dass die Leitung dieses Meetings anfangs von einer externen erfahrenen Scrum MasterIn übernommen werden sollte. Nach zwei bis drei Sprints kann die eigentliche Scrum MasterIn die Leitung übernehmen. Die Externe sollte jedoch noch beratend zur Seite stehen. Die Sprint Retrospektive ist für die Prozessverbesserung sehr wichtig und ausschlaggebend. Um hier die Professionalität gewährleisten zu können, sollte mit der beschriebenen externen erfahrenen Person gearbeitet werden. Somit wäre die Sprint Retrospektive an der FHS am idealsten gelöst.

## 4.2 Kompromiss-Variante

Dieses Unterkapitel beinhaltet die Variante des Fahrtenplans, bei dem Kompromisse eingegangen werden. Die folgenden Punkte ergeben, wie bei der idealen Variante, nur einen Sinn, wenn die Grundvoraussetzungen, welche bereits erläutert wurden, eingehalten werden. Die Variante mit Kompromissen beinhaltet dieselben Punkte wie die ideale Variante. Es werden jedoch Abstriche gemacht und wie gesagt Kompromisse eingegangen.

### 1. Besetzung von Scrum-Rollen:

#### (a) Scrum MasterIn:

Die Rolle der Scrum MasterIn sollte wie bei der idealen Variante besetzt werden. Steht die in der idealen Variante angesprochene externe erfahrene Scrum MasterIn zur Unterstützung nicht zur Verfügung, kann auf diese verzichtet werden. Die restlichen Punkte sind der idealen Variante zu entnehmen. Sollten dem Scrum-Team zu wenige EntwicklerInnen zur Verfügung stehen, sollte nach Meinung des Autors, die Scrum MasterIn trotzdem anfangs weniger Programmierarbeiten übernehmen. Ansonsten könnte die Scrum MasterIn überfordert sein und die Kontrolle über das Entwicklerteam verlieren. Ein kompletter Verlust der Kontrolle über das Entwicklerteam und den Projektfortschritt schadet dem Projekt vermutlich weit mehr, als weniger implementierte Features, welche die Scrum MasterIn erarbeiten hätte können.

#### (b) Product OwnerIn:

Die Rolle der Product OwnerIn sollte grundsätzlich wie bei der idealen Variante besetzt werden. Steht keine StudentIn des Fachbereiches „Steuerung“, oder „LEAD“ zur Verfügung, kann die Rolle von einer Person eines anderen Fachbereiches übernommen werden. Diese sollte jedoch zumindest über ein wenig Erfahrung im Bereich „Projektmanagement“ verfügen. Die restlichen Punkte sind der idealen Variante zu entnehmen. Insbesondere die Entscheidungsfindung und die Kommunikation sollten so gestaltet werden, wie sie in den entsprechenden Punkten, in der idealen Variante, angeführt sind.

#### (c) Entwicklerteam:

Die Besetzung des Entwicklerteams sollte wie in der idealen Variante vollzogen werden. Sollten nicht alle Mitglieder teamfähig sein, wird dies durch den Scrum-Prozess ausgeglichen. Durch die Selbstorganisation des Entwicklerteams können sich nicht-teamfähige Personen die Teamfähigkeit leichter aneignen.

#### (d) Sonstige Rollen:

Die Rollenvergabe der sonstigen Rollen sollte grundsätzlich wie bei der idealen Variante funktionieren. MitstudentInnen stehen sicherlich jederzeit zur Verfügung und können deshalb so gut wie immer die Rolle der „Chickens“ einnehmen. Bekommt ein Projekt einmal weniger Aufmerksamkeit und dadurch weniger verschiedene Stakeholder, sollte zumindest die für die Betreuung und Beurteilung zuständige Lehrperson als Stakeholder mit einbezogen werden.

### 2. Scrum-Voraussetzungen schaffen:

#### (a) Teamgröße:

Die Teamgröße sollte gemäß der Literatur eingehalten werden. Für zu kleine oder zu große Teams macht der Einsatz von Scrum keinen bzw. sehr wenig Sinn (siehe 2.3.1). Aus diesem Grund kann bei diesem Punkt auf nichts aus der idealen Variante verzichtet werden. Die dortigen Vorschläge sollten angewendet werden.

#### (b) Teammitglieder & Teamfähigkeit:

Siehe ideale Variante.

## (c) Infrastruktur

Sollten nicht für alle Projekte Räumlichkeiten zur Verfügung stehen, sollte es einen Bewerbungsprozess für Räume geben. Der Bewerbungsprozess für Räume wurde bereits im Punkt „Infrastruktur (an der FHS) 3.3.3, Abschnitt Lösungsansatz“ erläutert. Des Weiteren schlägt der Autor vor, Räume unter mehreren Projekt-Gruppen aufzuteilen. Die richtige Aufteilung ist eine Einteilung- und Koordinationsangelegenheit der Scrum MasterInnen. Diese sollten dafür sorgen, dass der Prozessablauf ungehindert fortfahren kann.

Können fehlende Utensilien wie Whiteboards oder Pinnwände seitens der FHS nicht angeschafft werden, sollten die Vorhandenen unter den Projekten aufgeteilt werden. Der Autor schätzt eine solche Anschaffung jedoch als langfristig sinnvoll ein und hofft, dass diese Investition in neue Utensilien seitens der Verantwortlichen der FHS vollzogen wird.

Sollte es seitens der FHS nicht möglich sein, jeder EntwicklerIn einen passenden Computer mit dem von ihr bevorzugten Betriebssystem zur Verfügung stellen, sollte seitens der Scrum MasterIn frühzeitig für Ersatz gesorgt werden. Es sollte zu Beginn eines Projektes die Entwicklungsumgebung abgeklärt werden. Dadurch können sich alle Scrum-Team-Mitglieder darauf einstellen. Der Autor kann aus eigenen Erfahrungen behaupten, dass dies ganz gut klappt. Während seines QPT3s haben sich alle Programmierer der QPT-Gruppe des Autors zu Beginn auf ein Betriebssystem geeinigt und mit diesem wurde entwickelt. Des Weiteren hat mittlerweile fast jede StudentIn bereits einen eigenen Laptop und entwickelt mit diesem.

Sollte es nicht möglich sein innerhalb des Settings der FHS benötigte Server, wie Production-, Staging-, oder Repository-Server zur Verfügung zu stellen, oder diese Server zu ausgelastet sein, so empfiehlt der Autor auf Github (Organisationsaccount der FHS), Github (Privataccount von StudentInnen), Heroku oder andere Gratis-Anbieter auszuweichen (vgl. 3.3.3, Abschnitt „Server und Services“, Punkt „Lösungsansatz“).

Ein Continuous-Integration Server sollte auch in der Kompromiss-Variante auf jeden Fall zur Verfügung stehen. Ist ein Ausweichen auf die bereits angesprochenen Gratis-Anbieter nicht möglich, schlägt der Autor folgendes vor: StudentInnen könnten sich eigene Server für die kontinuierliche Integration einrichten. Aus eigenen Erfahrungen ist ihm bekannt, dass einige Studierende über eigene Server verfügen. Die Konfiguration eines Hudson-Servers sollte grundsätzlich nicht allzu schwer sein. Das dafür notwendige Wissen wird teilweise an der FHS gelehrt. Wird hier seitens der StudentInnen zusammengearbeitet, könnten die notwendigen Server eigenständig angeschafft und konfiguriert werden.

## 3. Scrum-Artefakte:

Die Scrum-Artefakte sollten in der Variante mit Kompromissen grundsätzlich in elektronischem Format geführt werden. Steht beispielsweise kein Projektraum zur Verfügung, kann auf die elektronische Erfassung zugegriffen werden. Als Möglichkeit die Artefakte trotz nicht vorhandenem Projektraum analog zu führen, sieht der Autor in „mobilen“ Plattformen. So könnten Studierende beispielsweise ein Sprint-Burndown-Chart auf einer tragbaren Pinnwand oder auf einem Plakat anbringen. Diese Pinnwand oder das Plakat wird von einer StudentIn mit nach Hause genommen und zu sämtlichen Meetings wieder mitgebracht. Hier ist die Kreativität der StudentInnen gefragt.

## 4. Scrum-Timeboxen:

## (a) Release-Planning:

Der grundsätzliche Ansatz der idealen Variante ist beim Release-Planning in jedem Fall zu bevorzugen. Sind Meetings zweimal die Woche über einen längeren Zeitraum nicht möglich, da alle Studierenden beispielsweise gerade ein Berufspraktikum absolvieren, so würde der Autor folgendes vorschlagen:

Zu Beginn des Projektes werden zwei bis drei Workshop-Tage innerhalb einer Woche abgehalten. An diesen Terminen haben alle Team-Mitglieder und die betreuenden Lehrpersonen anwesend zu sein. Nach diesen Workshops kann jede StudentIn und BetreuerIn wieder ihren/seinen Weg gehen. Anschließend treffen sich die StudentInnen einmal pro Woche online und sprechen über das Projekt. Diese Meetings könnten beispielsweise über Skype oder ähnliche Chatprogramme ablaufen. Hat ein Projektteam neun Mitglieder, wird ein Gruppen-Video-Skype unter Umständen etwas unübersichtlich und verwirrend. In diesen Fällen würde der Autor auf andere Medien zugreifen. Gemeinsame Google-Docs-Dokumente, Dropbox Ordner, Facebook-Seiten, etc. könnten zum Informationsaustausch genutzt werden. Wichtig ist nur, dass das Projekt weiter verfolgt wird, und nicht nach den zwei bis drei Workshop-Tagen ein Monat lang überhaupt nichts mehr passiert. Betreuende Lehrpersonen sollten während dieser Phase regelmäßig durch die ProjektleiterIn den aktuellen Stand des Projektes mitgeteilt bekommen. Dies könnte über E-Mails passieren.

(b) Sprint-Planning:

Das Sprint-Planning sollte grundsätzlich wie in der idealen Variante funktionieren. Steht die dort angesprochene erfahrene Scrum MasterIn, welche für Unterstützung sorgen sollte, nicht zur Verfügung, kann im Notfall auf diese verzichtet werden. Die restlichen Punkte sollten der idealen Variante entnommen werden.

(c) Sprint:

Ein Sprint sollte gemäß der idealen Variante durchgeführt werden. Hier steht die richtige Priorisierung und Planung im Vordergrund. Hindernisse und Ablenkungen sollten so umgangen werden. Sollte kein Projektraum an der FHS zur Verfügung stehen, sollte das Scrum-Team trotzdem ein- bis zweimal pro Woche in einem Raum zusammen arbeiten. Hier könnten extra Räume an der FHS gebucht werden. Eine weitere Möglichkeit sind Projekträume im privaten Umfeld. Ein Studenten-WG-Wohnzimmer eines Scrum-Team-Mitglieds könnte beispielsweise einmal pro Woche als Projektraum genutzt werden. Die restlichen Punkte sollten der idealen Variante entnommen werden.

(d) Daily-Scrum:

Eine Modifikation des Daily-Scrum Meetings zu einem „MMFly“-Scrum Meeting, wie es in der idealen Variante angedacht ist, sollte zumindest angewendet werden. Im Scrum-Prozessablauf ist es besonders wichtig, dass jedes Entwicklerteam-Mitglied über den Stand der Arbeit der Anderen Bescheid weiß. Durch den starken Informationsaustausch kommen etwaige Probleme schneller ans Tageslicht, der Prozess wird gefördert und die Entwicklung ist besser (siehe 2.6.4). Aufgrund dessen sieht der Autor in der Modifikation zum „MMFly“-Scrum Meeting die einzige Möglichkeit, an der FHS, dieses Meeting möglichst nah, gemäß der Literatur, abzuhalten. Eine weitere Modifikation auf nur mehr ein oder zwei Meetings pro Woche wäre für den Scrum-Prozess nicht gut. Sollten StudentInnen an der FHS nicht anwesend sein, müssen diese Meetings über Skype oder andere Chatprogramme abgehalten werden.

(e) Sprint Review:

Das Sprint Review sollte auf jeden Fall, wie in der idealen Variante beschrieben, abgehalten werden. Durch dieses Meeting bekommen EntwicklerInnen sehr viel Feedback, was wiederum sehr wichtig für die zukünftige Arbeit ist. Am wichtigsten ist für den Autor, dass dieses Meeting in einem ordentlichen Rahmen abläuft. Aus diesem Grund sollte es auf jeden Fall im Rahmen einer verpflichtenden Lehrveranstaltung an der FHS abgehalten werden. Durch das Herzeigen von Ergebnissen seitens der StudentInnen werden auch andere Studierende inspiriert. Es wird gemeinsam über den Programm-Code und die Design-Arbeiten gesprochen und ein starker Austausch an Wissen und Ideen findet statt. Dies fördert die Gemeinschaft und den Austausch, was wiederum die Endergebnisse verbessert.

## (f) Sprint Retrospektive:

Die Sprint Retrospektive sollte gemäß der Literatur durchgeführt werden. Steht eine, in der idealen Variante beschriebene, externe Scrum MasterIn nicht zur Verfügung, kann Notfalls auf diese verzichtet werden. Wichtig ist aber, dass die Sprint Retrospektive ernst genommen wird. Aus eigenen Erfahrungen kann der Autor sagen, dass dies oft nicht der Fall ist. Hier ist seitens der Scrum MasterIn unbedingt darauf zu achten, dass die Sprint Retrospektive ordentlich abläuft. Womöglich können Abstriche bei den einzelnen Punkten gemacht werden, wie die Sprint Retrospektive abläuft (siehe 2.6.6), da die Scrum MasterIn anfangs mit den zu vielen Möglichkeiten eines Ablaufs überfordert sein könnte. Die Scrum MasterIn sollte aber zumindest darauf achten, dass miteinander gesprochen, diskutiert und analysiert wird. Findet dies anfänglich in einem chaotischen Rahmen statt, ist es immer noch besser, als wenn ganz auf die Sprint Retrospektive verzichtet wird.

**Anmerkung** Es muss festgehalten werden, dass die ideale Variante zu bevorzugen ist. Ist dies in gewissen Punkten (z.B. aufgrund von zu hohen Infrastruktur-Kosten, etc.) nicht möglich, kann auf die Variante mit Kompromissen zugegriffen werden. Eine Kombination beider Fahrtenplan-Varianten in den unterschiedlichen Punkten ist auch möglich.

## 5 Fazit

Zusammenfassend kann im Rahmen der vorliegenden Bachelorarbeit gesagt werden, dass mit Einhaltung der im Kapitel 4 „Fahrtenplan“ dargelegten Punkte ein Einsatz von Scrum in MMA- und MMT-Projekten, an der FHS, sinnvoll gestaltet werden kann. Die Forschungsfrage wie der Einsatz von Scrum funktionieren kann, ist somit beantwortet. Für die Erfüllung aller notwendigen Aspekte und Voraussetzungen sind nicht nur StudentInnen, sondern auch Verantwortliche seitens der beiden Studiengänge, zuständig. Die ersten StudentInnen, welche an der FHS Scrum einsetzen, haben/hatten sicherlich noch Pionierarbeiten für zukünftige Projekte zu leisten. Durch eine gute Zusammenarbeit von Studierenden und Lehrenden kann für zukünftige Projekte die Basis geschaffen werden, Scrum leichter einzusetzen.

Festzuhalten ist, dass der Autor die Erforschung des Themengebiets „Scrum in MMA- und MMT-Projekten“ zum jetzigen Zeitpunkt (Stand 01.07.2011) abgeschlossen hat. Mit der Einführung des MMT Masterstudiengangs, ab dem Wintersemester 2011, kann sich das Ergebnis der Forschungsfrage selbstverständlich ändern. Die ExpertInnen aus den Interviews haben den neuen Schwerpunkt „LEAD“ des MMT Masterstudiengangs bereits angesprochen. Wie dieser Schwerpunkt allerdings in der Praxis aussieht und funktioniert, ist schwer bzw. noch kaum zu sagen.

Als Forschungsausblick würde der Autor deshalb interessant finden, wie die Situation in fünf Jahren an der FHS aussieht. Wurden bereits Scrum-Projekte gemäß des Ergebnisses dieser Bachelorarbeit erfolgreich durchgeführt? Inwieweit hat der neue MMT Masterstudiengang die Situation an der FHS im Bezug auf Scrum verändert? Wurden bereits alle notwendigen Voraussetzungen geschaffen um Scrum zu 100% gemäß der Literatur an der FHS einzusetzen? Um dies festzustellen, müsste eine StudentIn das Forschungsgebiet zu einem späteren Zeitpunkt nochmal neu aufrollen.

Des Weiteren wäre interessant, ob mithilfe des aufgestellten Fahrtenplans, auch Projekte aus anderen Studiengängen der FHS durchführbar sind. Dies könnte beispielsweise durch eine Untersuchung von Projekten der Studiengänge ITS und DPM belegt werden.

## Literatur

- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J. & Thomas, D. (2001), 'Manifesto for Agile Software Development'. Stand: 2001, Aufruf am: 19.04.2011.  
**URL:** <http://www.agilemanifesto.org/>
- Cockburn, A. (2002), *Agile Software Development*, Addison-Wesley Professional, Boston, MA 02116, USA.
- Cockburn, A. & Williams, L. (2001), 'The Costs and Benefits of Pair Programming'. Stand: 2001, Aufruf am: 28.04.2011.  
**URL:** <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.26.9064&rep=rep1&type=pdf>
- Cohn, M. (2004), *User Stories Applied For Agile Software Development*, Addison-Wesley, Boston, MA 02116, USA.  
**URL:** <http://my.safaribooksonline.com/book/software-engineering-and-development/agile-development/0321205685/firstchapter>
- Cohn, M. (2010), *Succeeding with Agile - Software Development Using Scrum*, Addison-Wesley Professional, Boston, MA 02116, USA.
- Derby, E. & Larsen, D. (2006), *Agile Retrospectives: Making Good Teams Great*, Pragmatic Bookshelf.
- Funpic (2011), 'Shared Database'. Stand: 29.06.2011, Aufruf am: 29.06.2011.  
**URL:** <http://www.funpic.de/>
- Gloger, B. (2010), *Your Scrum Checklist*, bor!sgloger Wien. New York. Recife. Baden-Baden.  
**URL:** <http://www.infoq.com/resource/minibooks/scrum-checklists/en/pdf/Scrum%20CheckList%202011.pdf>
- Heroku, I. (2011a), 'Shared Database'. Stand: 30.06.2011, Aufruf am: 30.06.2011.  
**URL:** <http://www.heroku.com/pricing#0-0>
- Heroku, I. (2011b), 'What is Heroku?'. Stand: 30.06.2011, Aufruf am: 30.06.2011.  
**URL:** <http://about.heroku.com/>
- Highsmith, J. (2001), 'History: The Agile Manifesto'. Stand: 2001, Aufruf am: 26.06.2011.  
**URL:** <http://www.agilemanifesto.org/history.html/>
- Keith, C. (2010), *Agile Game Development with Scrum*, Addison-Wesley Professional, Boston, MA 02116, USA.
- Larman, C. (2003), *Agile & Iterative Development - A Manager's Guide*, Addison-Wesley Professional, Boston, MA 02116, USA.
- Martin, R. C. (2003), *Agile Software Development principles, patterns, and practices*, Prentice Hall, Upper Saddle River, NJ07458.
- MMA (2008), 'Leitfaden 2008 Qualifikationsprojekt 2 „Vertiefung im Fachbereich – Vorbereitung Berufspraktikum“'. Stand: 12.08.2009, Aufruf am: 03.05.2011, Publikation des Studiengangs MMA, Leitfaden.  
**URL:** [https://elearning.fh-salzburg.ac.at/lms/file.php/5/FAQ/QPT2/qpt2\\_leitfaden\\_2008-v2.pdf](https://elearning.fh-salzburg.ac.at/lms/file.php/5/FAQ/QPT2/qpt2_leitfaden_2008-v2.pdf)

- MMA (2011), 'Leitfaden 2011 Qualifikationsprojekt 1 „Fachbereichswahl“'. Stand: 03.05.2011, Aufruf am: 03.05.2011, Publikation des Studiengangs MMA, Leitfaden.  
**URL:** [https://elearning.fh-salzburg.ac.at/lms/file.php/5/FAQ/QPT1/qpt1\\_leitfaden\\_2011.pdf](https://elearning.fh-salzburg.ac.at/lms/file.php/5/FAQ/QPT1/qpt1_leitfaden_2011.pdf)
- MMA & MMT (2011a), 'Infrastruktur für Software Projekte bei MMA + MMT'. Stand: 28.06.2011, Aufruf am: 28.06.2011.  
**URL:** [https://mediacube.at/wiki/index.php/Infrastruktur\\_f%C3%BCr\\_Software\\_Projekte\\_bei\\_MMA\\_%2B\\_MMT](https://mediacube.at/wiki/index.php/Infrastruktur_f%C3%BCr_Software_Projekte_bei_MMA_%2B_MMT)
- MMA & MMT (2011b), 'Kleiner Kellerraum'. Stand: 29.06.2011, Aufruf am: 29.06.2011.  
**URL:** [https://mediacube.at/wiki/index.php/Kleiner\\_Kellerraum](https://mediacube.at/wiki/index.php/Kleiner_Kellerraum)
- MMA & MMT (2011c), 'Kojen'. Stand: 29.06.2011, Aufruf am: 29.06.2011.  
**URL:** <https://mediacube.at/wiki/index.php/Kojen>
- MMA & MMT (2011d), 'Leitfaden QPT3 Bachelor MultiMediaArt Bachelor MultiMediaTechnology Jahrgang B-2008 2010/11'. Stand: 03.05.2011, Aufruf am: 03.05.2011, Publikation der Studiengänge MMA und MMT, Leitfaden.  
**URL:** [https://mediacube.at/wiki/images/e/ee/Qpt3\\_leitfaden\\_2010\\_2011.pdf](https://mediacube.at/wiki/images/e/ee/Qpt3_leitfaden_2010_2011.pdf)
- MMA & MMT (2011e), 'MediaCube Projektserver'. Stand: 28.06.2011, Aufruf am: 28.06.2011.  
**URL:** [https://mediacube.at/wiki/index.php/MediaCube\\_Projektserver](https://mediacube.at/wiki/index.php/MediaCube_Projektserver)
- MMT (2008), 'ILV Software Engineering'. Stand: 02.11.2008, Aufruf am: 28.04.2011, Curriculum ILV Software Engineering.  
**URL:** [https://mediacube.at/wiki/index.php/Curriculum:MMT-2008\\_Modul\\_Softwareentwicklung#ILV\\_Software\\_Engineering](https://mediacube.at/wiki/index.php/Curriculum:MMT-2008_Modul_Softwareentwicklung#ILV_Software_Engineering)
- MMT (2010), 'Fachspezifische Qualifikation 2010'. Stand: 2010, Aufruf am: 03.05.2011, Publikation des Studiengangs MMT, Leitfaden.  
**URL:** [https://mediacube.at/wiki/images/7/7c/Fachspezqualifikation\\_leitfaden\\_QPT2A\\_2010.pdf](https://mediacube.at/wiki/images/7/7c/Fachspezqualifikation_leitfaden_QPT2A_2010.pdf)
- MMT (2011a), 'Leitfaden 2011 Basisqualifikation'. Stand: 03.05.2011, Aufruf am: 03.05.2011, Publikation des Studiengangs MMT, Leitfaden.  
**URL:** [https://mediacube.at/wiki/images/2/24/Basisqualifikation\\_leitfaden\\_2011.pdf](https://mediacube.at/wiki/images/2/24/Basisqualifikation_leitfaden_2011.pdf)
- MMT (2011b), 'Leitfaden 2011 Multimediale Qualifikation „QPT 2 b“'. Stand: 03.05.2011, Aufruf am: 03.05.2011, Publikation des Studiengangs MMT, Leitfaden.  
**URL:** [https://mediacube.at/wiki/images/2/25/Qpt2b\\_multimediaqualifikation\\_leitfaden\\_2011.pdf](https://mediacube.at/wiki/images/2/25/Qpt2b_multimediaqualifikation_leitfaden_2011.pdf)
- Pichler, R. (2008), *Scrum - Agiles Projektmanagement erfolgreich einsetzen*, dpunkt.verlag GmbH, Ringstraße 19, 69115 Heidelberg, Deutschland.
- Rödig, M. (2010), 'Erhöhung der Softwarequalität in kleinen Unternehmen durch einen Continuous Integration-Prozess'. Diplomarbeit, Hochschule RheinMain, Fachbereich Design Informatik Medien, Studiengang Allgemeine Informatik.  
**URL:** [http://www.eworks.de/research/2010/02/CI-Hudson/2010-05-25\\_Diplomarbeit\\_Martin\\_Roedig.pdf](http://www.eworks.de/research/2010/02/CI-Hudson/2010-05-25_Diplomarbeit_Martin_Roedig.pdf)
- Schwaber, K. (1997), 'SCRUM Development Process'. Stand: 1997, Aufruf am: 30.06.2011.  
**URL:** <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.86.4164&rep=rep1&type=pdf>
- Schwaber, K. & Sutherland, J. (2010), 'Scrum'. Stand: 2010, Aufruf am: 20.04.2011.  
**URL:** <http://www.scrum.org/storage/scrumguides/Scrum%20Guide.pdf>

Sepulveda, C. (2011), 'Is Tracker really free for public projects, individual use, non-profits, and educators?'. Stand: 04.05.2011, Aufruf am: 04.05.2011.

**URL:** <https://www.pivotaltracker.com/help#istrackerreallyfreeforpublicprojectsindividualuseno>

Sixtensson, A. (2011), 'Development using Scrum'. Stand: 29.06.2011, Aufruf am: 29.06.2011.

**URL:** <http://expo-c.se/pdf/SCRUM.pdf>

Sommerville, I. (2001), *Software engineering (5th ed.)*, Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA.

Takeuchi, H. & Nonaka, I. (1986), 'The New New Prdouct Development Game'. Stand: 1986, Aufruf am: 07.04.2011.

**URL:** <http://www.sao.corvallis.or.us/drupal/files/The%20New%20New%20Product%20Development%20Game.pdf>

West, D. & Grant, T. (2010), 'Agile Development: Mainstream Adoption Has Changed Agility'. Stand: 2010, Aufruf am: 30.06.2011.

**URL:** [http://www.osp.ru/netcat\\_files/18/10/h\\_d8eddd303b6cf0c38c23601c4363bee4](http://www.osp.ru/netcat_files/18/10/h_d8eddd303b6cf0c38c23601c4363bee4)

Williams, L., Kessler, R. R., Cunningham, W. & Jeffries, R. (2000), 'Strengthening the Case for Pair Programming'. Stand: 2000, Aufruf am: 28.04.2011.

**URL:** <http://collaboration.csc.ncsu.edu/laurie/Papers/ieeeSoftware.PDF>

Wirdemann, R. (2009), *Scrum mit User-Stories*, Carl Hanser Verlag, München.

## Abbildungsverzeichnis

1	Beispiel zusätzliche Signalisierung des Build-Verlauf bei Continuous Integration Server . . . . .	13
2	Beispiel Release-Burndown . . . . .	16
3	Beispiel Sprint-Burndown . . . . .	17
4	Beispiel Scrum Zyklus Quelle: <a href="http://de.wikipedia.org/wiki/Datei:Scrum_process-de.svg">http://de.wikipedia.org/wiki/Datei:Scrum_process-de.svg</a> . . . . .	23
5	Beispiel Release-Planning . . . . .	48

## Tabellenverzeichnis

1	Potenzielle Scrum Projekte FHS . . . . .	29
---	--	----

## A Interviews

### A.1 Interview mit Frau Jeanny Gucher

Gucher = Gucher Jeanny (Expertin) Frick = Frick Matthias (Autor)

1 Frick:

2 Hallo! Es geht um meine Bachelorarbeit. Ich möchte das Thema „Scrum in FH-Projekten“ an-  
3 gehen bzw. bearbeiten und analysieren wie man das am besten umsetzen kann. Was muss hier  
4 seitens der FH noch getan werden und wie muss generell vorgegangen werden? Und hierfür benö-  
5 tige ich noch eine professionelle Meinung dazu.

6 Gucher:

7 Ok, danke für die Ehre.

8 Frick:

9 Aus diesem Grund wollte ich Sie fragen was sie bisher generell für Erfahrungen mit Scrum ge-  
10 macht haben. Was haben Sie bisher für Erfahrungen gemacht?

11 Gucher:

12 Naja ich habe nicht nur hier an der FH Scrum gemacht, sondern auch im Rahmen einer Un-  
13 ternehmensberaterischen Tätigkeit. Hier habe ich bereits einige Kunden bei der Implementierung  
14 von Scrum begleiten dürfen, zum Beispiel. Und ich würde einmal sagen, dass es auf jeden Fall für  
15 das Thema „Softwareentwicklung“, egal ob jetzt für „Game“ oder für andere Sparten ist, inzwi-  
16 schen die „State-of-the-art“-Methode geworden ist. Und das es sicher die beste Variante ist um  
17 einen sehr schwer planbaren Entwicklungsprozess dennoch steuerbar zu machen und dennoch mit  
18 der entsprechenden Involvierung aller notwendigen Stakeholder hinzubekommen. Deshalb bin ich  
19 sehr begeistert von dieser Methode und ich finde, dass sie gut ist. Aber natürlich gelingt dennoch  
20 nicht allen Unternehmern das auch wirklich in vollem Umfang umzusetzen. Es braucht von der Or-  
21 ganisation natürlich auch einen spezifischen Zugang und ein spezifisches Verständnis zum Thema  
22 Projekt und Projektarbeit das Scrum auch wirklich implementiert werden kann.

23 Frick:

24 Sehen Sie den Einsatz von Scrum in FH-Projekten als sinnvoll an?

25 Gucher:

26 Also ich denke mir, wir haben sehr viele Projekte, die einen hohen entwicklerischen bzw. software-  
27 entwicklerischen Anteil haben, so wie alle Projekte die zum Beispiel mit „Game“ zu tun haben,  
28 oder viel Animation dabei ist, oder ähnliches. Hier wäre Scrum für solche Teilbereiche zumindest,  
29 eine sehr sehr sinnvolle Methode. Das wird von mir auch unterstützt. Also wir haben jetzt im  
30 Master zum Beispiel zwei große „Game“-Projekte die auf jeden Fall mit Scrum arbeiten werden.

31 Frick:

32 Okay. Und generell finden Sie das Scrum leicht zu erlernen ist? Oder wie kommt man am bes-  
33 ten hinein in diese ganze Schiene zu arbeiten?

34 Gucher:

35 Also ich glaube Scrum ist jetzt von dem wie es funktioniert nicht wirklich kompliziert. Es ist  
36 ein bisschen wie eine eigene Sprache zu lernen ist, da es sehr viel eigene Begrifflichkeiten verwen-  
37 det werden wie zum Beispiel „Scrum Master“ oder „Product Backlog“, oder „Burn-Down-Chart“  
38 usw. Aber in Wirklichkeit bedeutet Scrum einfach einem Team, einer Organisation ein sehr ho-  
39 hes Maß an Disziplin auch abzuverlangen, nämlich vom Rahmen her. Also wirklich jeden Montag  
40 „Scrum-Planning“ und jeden Tag „Daily-Scrum“, d.h. gewisse Routinen werden einfach diszipliniert  
41 eingefordert. Und das paradoxe daran ist, dass erst durch diese Diszipliniertheit vom Rahmen der

42 Freiraum für das inhaltliche Programmieren entsteht. Das ist gerade das Schöne daran, dass quasi  
43 eine Seite davon sehr diszipliniert sein muss, damit auf der anderen Seite der maximale Freiraum  
44 erst wirklich ausgeschöpft werden kann.

45 Frick:

46 Denken Sie, dass in der FH Scrum direkt aus der Literatur eins-zu-eins übernommen werden  
47 kann?

48 Gucher:

49 Nein, ich denke aufgrund mehrerer Dinge nicht, weil natürlich studentische Projekte immer anders  
50 sind, als wie wenn man Projekte hauptberuflich innerhalb einer Agentur oder einer Spiele-Firma  
51 macht. Sie haben ja niemals diese disziplinären Möglichkeiten mit dem Team. Es ist viel schwie-  
52 riger an der FH bei nicht erledigten Aufgaben irgendwelche Sanktionen zu erteilen. Also diese  
53 Beschränkung werden Sie immer haben. Ich denke aber das Scrum dennoch viele Elemente enthal-  
54 tet, die man gut umsetzen kann. Hierzu würde ich zum Beispiel die wöchentlichen Routinen oder  
55 die tägliche kurze Abstimmung zählen. Das sind Themen die auch Studierende für sich organisie-  
56 ren können und die einen Sinn machen. Aus diesen Gründen würde ich sagen, ja Scrum ist aus der  
57 Literatur übernehmbar, aber sicher nicht zu 100%. Ich würde sagen gute 80% sind übernehmbar,  
58 dann bin ich schon ganz happy.

59 Frick:

60 Damit diese 80% erreichbar sind, was würden Sie hier von der FH-Seite, zum Beispiel am Curri-  
61 culum ändern?

62 Gucher:

63 Naja vom Curriculum her, was wir bereits getan haben, was beim nächsten Jahrgang, der mit  
64 dem Bachelor beginnt, wirksam werden wird, ist, dass wir das Thema „Projektmanagement“ im  
65 Curriculum völlig umstrukturiert haben. Das war bis jetzt ja leider eine sehr große Vorlesung mit  
66 60 Personen, daher war es Didaktisch schwierig in Gruppen dort zu arbeiten. Das hat sich jetzt  
67 verändert, jetzt werden wir in kleinen Gruppen arbeiten können und damit werden wir auch auf  
68 speziellere Vorgehensmethoden wie Scrum wahrscheinlich schon im Bachelor, zumindest ein biss-  
69 chen besser, eingehen können, als wie es bis jetzt der Fall war, nämlich gar nicht. Und im Master  
70 ist es so, dass es definitiv Teil vom Lehrplan ist. Also für alle Leute die den MMT-Master machen  
71 auf jeden Fall, weil die müssen agile Methoden der Entwicklung lernen und für die MMA-Master-  
72 Studierenden, wenn sie ein solches Projekt leiten, ebenfalls.

73 Frick:

74 Hier geht es dann in die Richtung „Leadership“?

75 Gucher:

76 Genau, bei MMT wäre es der Bereich „LEAD“, und hier gibt es eben auch einen Teil „agile  
77 Methoden“ und Scrum gehört zu den so genannten agilen Methoden des Projektmanagements.  
78 Und bei den MMA Master Studierenden die Management und Führung als Vertiefung machen,  
79 und „Game“-Projekte übernehmen, müssen sie auch mit Scrum arbeiten.

80 Frick:

81 Liegt dann hier der Schwerpunkt nur bei den Game-Projekten?

82 Gucher:

83 Momentan schon, je nachdem wenn ein anderes Projekt natürlich ebenfalls einen hohen Pro-  
84 grammieraufwand beinhalten würde, dann würde es dort genauso empfohlen werden. Momentan  
85 sind das in erster Linie Game-Projekte.

86 Frick:

87 Okay. Und würden Sie bereits im Rahmen des Bachelors irgendwelche Workshops beispielswei-

88 se für „Product Owner“ oder „Scrum Master“ sinnvoll finden, so dass man hier zum Beispiel für  
89 einen Tag lang reinschnuppern kann? Oder das das sogar Teil vom Lehrplan wird?

90 Gucher:

91 Naja, das Problem ist ein bisschen das wir leider im Bachelor sehr wenig Unterrichtsstunden  
92 für Managementfächer haben. Ich wüsste ehrlich gesagt nicht, woher wir die Unterrichtsstunden  
93 nehmen sollten. Wir könnten uns überlegen ob es irgendwas in Richtung Freifach geben könnte.  
94 Aber ja das müsste man sich anschauen ob wir noch irgendwo einen Puffer an Stunden hätten.  
95 Hier ist mehr das Problem dabei, dass diese Stunden zum Thema Management die wir haben be-  
96 reits für Dinge verwendet werden, die wir sowieso unterrichten müssen, sowieso nicht ausreichen.  
97 Deshalb ist noch ein zusätzliches neues Thema aufzunehmen sehr sehr schwierig. Ja, also sinnvoll  
98 wäre es für mich schon, aber aus den dargelegten Gründen ist es schwierig. Sinnvoll wären aus  
99 meiner Perspektive mehr Management-Stunden auf jeden Fall, aber das wird im Bachelor nicht  
100 der Fall sein. Im Master ist es besser, hier haben wir das gut abfangen können. Im Bachelor ist  
101 das sehr schwierig.

102 Frick:

103 Im MMT-Master ist jetzt sowas wie eine Scrum Master Ausbildung in der LEAD-Vertiefung ge-  
104 plant?

105 Gucher:

106 Das ist sowieso dabei. Also es ist so, dass wir jetzt keine vollständige Ausbildung zum Scrum  
107 Master anbieten werden, dafür gibt es andere Institutionen. Aber es ist so, dass wenn Sie LEAD  
108 machen, die Vertiefung, auf jeden Fall eine gute, versierte Basis davon haben, wie Scrum funktio-  
109 niert.

110 Frick:

111 Sollte dann praktisch derjenige der LEAD macht die anderen überzeugen und ihnen dies bei-  
112 bringen? Wenn einer nicht LEAD macht, wird es dann für denjenigen schwieriger?

113 Gucher:

114 Naja, der kann sich dann trotzdem in Vorlesungen des Bereichs LEAD setzen und diese Besu-  
115 chen. Man hat ja immer die freie Wahl noch etwas zusätzlich zu machen. Ich denke mir, inwieweit  
116 die Sinnhaftigkeit und die Funktionsweise von Managementmethoden für alle aus den multimedial-  
117 en Studiengängen zugänglich werden, ist eh eine Riesfrage. Weil es ja leider Gottes sehr viele  
118 Leute gibt die nach wie vor glauben, das ist etwas was man nicht so wirklich braucht, oder nur  
119 bei großen Projekten. Das ist natürlich nicht meine Meinung, aber das ist die Ambivalenz die wir  
120 hierzu einfach haben. Also das gerade der Kreativbereich nicht so wahnsinnig auf die Manage-  
121 mentsachen normalerweise steht und erst durch hartes Erfahrungslernen mitbekommt für was das  
122 eigentlich schon einen Sinn macht, wenn man jemanden hat der steuert, managet oder führt. Aber  
123 ja ich bin guter Dinge, da wir das Curriculum sehr sehr breit und gut aufgebaut haben, das das  
124 eh erlebbar wird, das das einen Unterschied macht, wenn man jemanden hat, der das gut im Griff  
125 hat, oder nicht.

126 Frick:

127 Wie könnte man die für Scrum benötigte Infrastruktur an der FH verändern?

128 Gucher:

129 Naja die Geschichte ist das, selbst wenn der Zubau der FH fertig ist, können wir nicht jeder  
130 Projektgruppe einen Raum zur Verfügung stellen und garantieren. Das wird einfach nicht so sein,  
131 sondern wir werden wahrscheinlich eher ein Modell wählen müssen, wo sich die Projekte um Räu-  
132 me bewerben. Also so wie man sich um Stipendien bewirbt, wird man sich um Räume bewerben  
133 können. Das natürlich die vom Konzept her, das Zusammenarbeiten von Leuten in einem Raum  
134 sinnvoll erscheinen lassen und da sind wieder diese Projekte dabei. Diese haben wahrscheinlich ein-  
135 fach gute Argumente warum sie den Raum brauchen und haben dadurch bei Bewerbungen auch

136 dementsprechende Chancen die auch zu bekommen. Eins ist klar, Scrum funktioniert im eigentli-  
137 chen wenn man die Leute in einem Raum zusammensetzen kann. Und wenn man dort dann auch  
138 die Infrastruktur für Scrum hat, sprich die Charts und so. Das ist sicher eine Voraussetzung das  
139 Scrum gut funktionieren kann. Wobei hier muss ich nochmals festhalten, dass in Zukunft nicht alle  
140 mit Scrum geführten Projekten Räume bekommen, da die Ressourcen einfach so knapp sind. Das  
141 ist auch eine Managementaufgabe zu sagen, wo können wir das Projekt realisieren. Kriegen wir  
142 einen Raum der FH dafür, oder organisieren wir einen anderen Raum für die Zusammenreffen,  
143 das bleibt einfach trotzdem die Aufgabe des Managers oder der Managerin. Wir werden nie für  
144 alle Projekte genügend Räume haben, das wird einfach nie so sein.

145 Frick:

146 Ja es werden ja auch immer mehr Studenten.

147 Gucher:

148 Genau wir haben immer mehr Projekte und immer mehr Studierende Gott sei Dank, aber das  
149 bedeutet gleichzeitig, dass der Platz nie reichen wird, auch wenn dann der Zubau fertig ist.

150 Frick:

151 Und wie sehen sie generell das Problem, dass an der FH nicht Vollzeitanwesenheit herrscht  
152 und Studierende auch anderes zu tun haben wie diverse andere Projekte, Prüfungen, etc. und  
153 nicht nur ein Scrum geleitetes Projekt ansteht?

154 Gucher:

155 Das finde ich eigentlich sehr gut. Ich denke mir, dass das eigentlich eine sehr gute spätere Realität  
156 widerspiegelt, da man dann auch nicht nur an einem Projekt mitarbeitet. Die Realität bedeutet ja  
157 meistens, dass ich drei, vier oder fünf Projekte sogar gleichzeitig habe und ich mich auch in meinen  
158 Job später auf unterschiedliche Dinge gleichzeitig konzentrieren werde müssen. Und ob das jetzt  
159 gleichzeitig ein anderes Projekt oder gleichzeitig eine Prüfung oder eine Arbeit ist, ist ziemlich  
160 egal. Ja, aber Tatsache ist, dass ich keine einzige Firma kenne wo Leute 100% nur an einem Pro-  
161 jekt gleichzeitig mitarbeiten. Eigentlich sind die Mitarbeiter überall immer in mehrere Projekte  
162 gleichzeitig involviert. Und damit sind sie genau in der gleichen Situation wie die Studierenden.  
163 Ich glaube, dass das spätere Realität sogar sehr gut abbildet.

164 Frick:

165 So wäre das eigentlich eine gute Vorbereitung?

166 Gucher:

167 Ja genau eine gute Vorbereitung, weil man wird immer die Wahl haben und für sich selber Prio-  
168 ritäten setzen, wo arbeite ich jetzt mit welcher Intensivität und mit welchen Prioritäten, das wird  
169 auch später so sein. Das ist die schlechte Nachricht.

170 Frick:

171 Könnten Sie noch Tipps, Anregungen, etc. zu meiner Arbeit geben? Habe ich eventuelle Fragen  
172 vergessen? Wie sehen Sie das?

173 Gucher:

174 Die Fragen haben mir gut gefallen. Die Frage ist, was ist denn Ihre Forschungsfrage dabei?

175 Frick:

176 Meine Forschungsfrage ist, wie kann man Scrum sinnvoll in FH-Projekten einsetzen?

177 Gucher:

178 Okay

179 Frick:

180 Und ich möchte damit im Endeffekt als Ergebnis einen Fahrtenplan für zukünftige Studieren-  
181 de und deren Projekte erstellen. Dadurch soll es leichter sein für diese Gruppen zu arbeiten und  
182 Scrum sinnvoll einzusetzen.

183 Gucher:

184 Ich denke mir das erste ist immer, dass man sich überlegen muss für welche Arten von Projekten  
185 kommt Scrum am besten in Frage. Also ich würde zum Beispiel kein Filmprojekt in Scrum ma-  
186 chen, weil das eignet sich nicht dafür. Das wäre für mich die erste Fragestellung die zu beantworten  
187 ist. Und dann wenn ich sage, okay das ist eine Projektart für die Scrum geeignet ist, dann würd  
188 ich sagen muss man sicher dem ganzen Team die Methode zuerst mal erklären und transparent  
189 machen, was bedeutet das? Welchen Rahmen wird das für uns darstellen? Welche Rollen gibt's  
190 dabei? Inwieweit unterscheidet sich zum Beispiel ein Scrum Master vom klassischen Projektleiter  
191 oder Producer? Was bedeutet Selbstorganisation zum Beispiel bei Scrum? Das wären für mich so  
192 Fragestellungen die auf jeden Fall mit dem Team abzuklären sind und dann geht es natürlich auch  
193 darum, wie Sie gesagt haben, habe ich einen Raum dafür? Kann ich dort die Infrastruktur schaffen  
194 die Scrum braucht? Und ja dazu wäre sicher gut, wenn ich mir so früh wie möglich überlege, und  
195 mich aus der Managementrolle so früh wie möglich darum kümmere, dass das Team einen Platz  
196 hat wo es Scrum miteinander auch leben kann.

197 Frick:

198 Also finden Sie nach der Reihenfolge am sinnvollsten, dass man zuerst mit dem Team selber  
199 alles ausmacht und dann die Rahmenbedingungen schafft?

200 Gucher:

201 Ja, wobei das mit dem Raum und den Rahmenbedingungen würd ich schon im Vorfeld klären.  
202 Ansonsten wecke ich vielleicht Erwartungshaltungen im Team, die ich dann später nicht erfüllen  
203 kann. Das ist dann auch schwierig, dann freuen sich alle darauf, dass jetzt so gearbeitet wird und  
204 dann ist es schwierig, dass man zum Beispiel nicht über die gesamte Projektzeit einen Raum zur  
205 Verfügung hat. Hier könnte man dann einfach einschränken und sagen, dass man nur in der Phase  
206 Scrum einsetzt, wo ein Raum zur Verfügung steht. Also von dem her denke ich, dass Scrum gut  
207 geeignet ist für studentische Projekte, die von ihrem Inhalt her auch die Anwendung von Scrum  
208 sinnvoll machen. 100% wird nicht gehen, aber 80% ist, denke ich, realistisch möglich, das man  
209 umsetzen kann und ich glaube, dass genauso wie bei Scrum in der realen Wirtschaft, auch im stu-  
210 dentischen Bereich dieser Rahmen, der relativ viel Disziplin verlangt, inhaltlich sehr viel Freiraum  
211 eröffnet.

212 Frick:

213 MMT hat ja bisher die Lehrveranstaltung Software Engineering und dort den theoretischen Back-  
214 ground erlernt. Bei MMA ist mir bisher von solch übermitteltem Wissen nichts bekannt. Denken  
215 Sie, dass dieser Unterschied Probleme machen könnte?

216 Gucher:

217 Meiner Meinung nach nicht, weil es ist so, dass in diesen multimedialen Projekten ja immer Fach-  
218 bereiche zusammen kommen, die nicht voll inhaltlich wissen, was der andere tut. Das ist ja gerade  
219 der Scherz daran, dass man eben unterschiedliche Diskurse hat. Und eine dieser Logiken ist dann  
220 die Management-Logik dazu und wenn man sich hier für Scrum entscheidet, dann gilt das für alle  
221 und nachdem ja die Leute von MMA genauso Teil dieser Entwicklung sein müssen, ist es glaub ich  
222 ziemlich egal. Wenn es beispielsweise um Spiele-Entwicklung geht, dann entwickeln sich die Parts,  
223 die dann die Creative-Parts sind, genauso in iterativen Schritten wie sich die Software iterativ  
224 entwickelt. Also vom Prozedere her, wie ich dann zu Ergebnissen komme, ist sich das so ähnlich oder  
225 ähnlich genug, dass das aus meiner Sicht überhaupt kein Problem darstellt und insgesamt diesen  
226 Prozess wirklich unterstützen wird, auch wenn es sich nicht nur auf reine Softwareprogrammierung  
227 bezieht.

228 Frick:

229 Okay, das war somit eh schon meine letzte Frage.

230 Gucher:

231 Ich finde das ganze wirklich spannend! Und wenn Sie noch Fragen haben oder irgendetwas brau-  
232 chen, können Sie gerne nochmals vorbeikommen. Ich finde wie gesagt Scrum sehr sinnvoll und  
233 wünsche Ihnen viel Erfolg bei Ihrer Forschung.

234 Frick:

235 Danke für das Gespräch und die vielen Informationen.

Transkript des Interviews zwischen der Expertin Gucher Jeanny und dem Autoren Frick Matthias,  
2011, Puch bei Hallein

## A.2 Interview mit Herrn Stefan Randelshofer

Randelshofer = Randelshofer Stefan (Experte) Frick = Frick Matthias (Autor)

236 Frick:

237 Hallo! Ich bearbeite während meiner Bakk2 das Thema Scrum in FH-Projekten mit der For-  
238 schungsfrage „Wie kann der Einsatz von Scrum in Projekten der Studiengänge MMA und MMT,  
239 an der Fachhochschule Salzburg, sinnvoll funktionieren?“ Ja hier möchte ich erforschen was alles  
240 getan werden muss, um den Einsatz von Scrum in FH-Projekten sinnvoll gestalten zu können.

241 Randelshofer:

242 Okay. Was mich vorweg interessiert. Verwendest du in einem Projekt oder einer Arbeit gerade  
243 Scrum? Oder möchtet ihr Scrum dann hernehmen? Oder ist diese Bakk2 von dir eine reine Theo-  
244 rie wie es einmal sein könnte?

245 Frick:

246 Also wir verwenden Scrum im Rahmen unseres QPT3s für die Programmierer intern und wen-  
247 den Scrum an. Aber im gesamten Team, sprich inklusive den MMA-Studenten, ist es sich diesmal  
248 leider nicht ausgegangen.

249 Randelshofer:

250 Bei was für einem Game-Projekt bist du dabei?

251 Frick:

252 Ich bin bei keinem Game-Projekt dabei. Wir machen ein gerade ein Webprojekt. Wir gestal-  
253 ten das Portfolio der Studiengänge MMT und MMA neu, zusätzlich erstellen wir einen Nachfolger  
254 des mfg-Magazins. Unser Projekt trägt den Namen „quer-hoch“.

255 Randelshofer:

256 Und warum machen die MMAler beim Scrum-Prozess nicht mit? Haben sie gesagt sie wollen  
257 nicht mitmachen, oder weil ihr sie nicht gefragt habt?

258 Frick:

259 Ja sie wollten nicht wirklich mitmachen. Und wir Programmierer haben uns dazu entschieden,  
260 das wir Scrum einsetzen wollen.

261 Randelshofer:

262 Und wer macht bei euch den Scrum Master? Wie habt ihr das System aufgesetzt?

263 Frick:

264 Im Moment macht der Dominik Goltermann eine Art Scrum Master, aber es läuft eher schwim-

265 mend. Wir probieren es erst einmal aus.

266 Randelshofer:

267 Ok, das wollte ich nur zu deinem Hintergrund wissen. Dann leg einmal los.

268 Frick:

269 Ja, ich wollte Sie fragen, was Sie generell bisher für Erfahrungen mit Scrum in der FH gemacht  
270 haben?

271 Randelshofer:

272 Du kannst bitte „Du“ zu mir sagen. Was ich für Erfahrungen mit Scrum gemacht habe?

273 Frick:

274 Ja.

275 Randelshofer:

276 Das es schwierig ist, aber möglich um es einmal so kurz wie möglich zusammen zu fassen. Es  
277 ist schwierig, aber möglich. Man hat natürlich nicht die selben Rahmenbedingungen wie in einer  
278 Firma. In einer Firma ist das anders. In einer Firma weiß man, dass es einen Auftraggeber gibt,  
279 dass es das Bezahlungs-Verhältnis gibt, es gibt die Leistung die zu erbringen ist. Egal ob das  
280 jetzt ein Freelancer ist, der ein Werk oder eine Leistung erbringen muss, oder ob es ein Angestell-  
281 ter ist, der nur die Arbeitszeit erbringen muss, aber man kann zu diesen Personen sagen „Mach  
282 was“ Zusätzlich gibt noch eine Weisungsgebundenheit, usw. Dies alles fehlt einem komplett in einem  
283 Scrum-Projekt an der FH. An der FH ist man vielleicht Weisungsgebunden, dass man eine Abgabe  
284 macht, aber der Weg dahin kann keiner einem vorschreiben, so zu sagen. D.h. als Scrum Master  
285 wird man es immer mit einem Team zu tun haben, das schlicht und einfach keinen Bock hat,  
286 oder sich komplett heraus hält. Als Scrum Master hat man hier kaum eine Möglichkeit irgendwie  
287 vorzugehen um die Team-Mitglieder zu motivieren. Man hat nicht die Möglichkeit Sanktionen zu  
288 erteilen. Für mich persönlich ist Sanktionen erteilen sowieso keine Lösung, aber manche brauchen  
289 das einfach um etwas zu arbeiten. Das einzige was man hier machen kann, und so mache ich es  
290 bei den gerade laufenden Projekten, ist es die anderen zu begeistern. Nur über die Begeisterung  
291 der Leute kann man arbeiten. Das ist meiner Meinung nach die viel edlere Methode. Ich finde die  
292 Methode, dass der Chef Sanktionen erteilen kann, und deshalb wird etwas gemacht, unpassend.  
293 Die edle Methode ist sicherlich die Methode, bei der die Team-Mitglieder mehr Einsatz zeigen  
294 und mehr hergeben, als wenn sie geknechtet werden. Ich finde, dass Scrum dafür steht, dass eben  
295 nicht dieses geknechtete System eingesetzt wird, sondern das Begeisterungs-System. Man ist hier  
296 an der FH dazu gezwungen, alleine über die Begeisterung der Team-Mitglieder, diese zum Einsatz  
297 motivieren zu können. Man muss eine Vision des Produktes bilden und haben und ist gezwungen  
298 die Leute diese Vision gut finden zu lassen. Deshalb arbeiten sie für einen. Die andere Methode ist  
299 für alle Team-Mitglieder mehr oder weniger irrelevant. Jeder kann zu jeder Zeit im Rahmen der  
300 FH sagen, dass er/sie aus dem Projekt aussteigen will, oder einfach eine Woche nichts arbeiten  
301 will. Hier hat man keinen Angriffspunkt an die Team-Mitglieder. Wenn der Scrum-Master jetzt  
302 irgendein Lehrer wäre, wäre es natürlich etwas anderes, aber das ist ja nicht der Fall, sondern  
303 ihr wollt euch ja selber organisieren. Das würde ich einmal als meine wichtigste Erfahrung beim  
304 Einsatz von Scrum in einer Schule festhalten. Es fehlt einfach die Hierarchie. Alle Studenten un-  
305 tereinander sind gleichwertige Kollegen. Auch wenn einer jetzt Product Owner innerhalb eines  
306 Projektes ist, ist er auch nur Student und dadurch nicht höher gestellt.

307 Frick: Ist deiner Meinung nach deshalb Scrum aus der Literatur nicht eins-zu-eins anwendbar? Mit  
308 den ganzen Voraussetzungen die gegeben sein müssen um Scrum einzusetzen?

309 Randelshofer: Doch ist schon möglich. Ich finde das es möglich ist. Natürlich wird man nicht  
310 darum herumkommen es zu tweaken und Modifikationen zu machen. Aber ich kann jetzt hier  
311 behaupten, dass alle Meetings abgehalten werden können. Ich kann ein Estimations machen, ich  
312 kann Sprint Planning 1 und 2 machen, ich kann Review und Retrospektive machen. Ich krieg

313 mein Daily-Scrum zusammen, aber ich muss überall gewisse Sachen einhalten. Bei Scrum hat  
314 man seine Meetings, seine Rollen und deine Artefakte. Das sind die drei Überkategorien. Wenn  
315 man alle diese Überkategorien einmal durchgeht: Meetings: sind alle möglich zu gestalten und  
316 einzutragen. Beim einzigen wo es schwierig wird, ist das Daily-Scrum Meeting. Man hat einfach  
317 nicht die Firmenstruktur, man hat unterschiedliche Stundenpläne. Ich versuche das bei laufenden  
318 Projekten folgendermaßen zu machen. Wir haben hier ein wenig modifiziert. Wir sind vom Daily-  
319 Scrum in einen MMFly-Scrum übergegangen. D.h. das normale Daily-Scrum Meeting findet nur  
320 am Montag, Mittwoch und Freitag statt. Wir treffen uns jeden Montag, Mittwoch und Freitag  
321 um 10:30 Uhr und halten hier unser 10-Minuten Daily-Scrum als Stand-Up Meeting ab. Das  
322 ist manchmal trotzdem ein Problem wenn jemand nicht da ist. Für mich ist das Daily-Scrum  
323 aber ein sehr wichtiger Bestandteil, wenn nicht sogar der wichtigste von Scrum. Aber wir haben  
324 eingeführt, sollte jemand einmal nicht können, nimmt er/sie einfach über Skype am Meeting teil.  
325 D.h. im Klartext kann man die Daily-Scrum Meetings abhalten, zwar etwas herunter reduziert,  
326 aber trotzdem ist es möglich, ohne das die Team-Mitglieder durchdrehen. Es ist einfach nicht  
327 möglich, dass jeder immer um 10:30 Uhr anwesend sein kann, da einfach diese Firmenstruktur  
328 an der FH fehlt. Durch das MMTfly-Scrum habe ich einen relativ gutes Update wie gerade jeder  
329 am arbeiten ist. Rollen: Der Scrum-Master ist klar, bzw. meistens klar. Einer macht meistens den  
330 Steuerungsschwerpunkt oder ist bereit den Scrum Master zu machen. Das Team ist mehr oder  
331 minder auch klar. Manager ist soweit auch klar, das ist die FH Salzburg, weil es die Facilities der  
332 FH sind. Wenn der Raum nicht funktioniert, oder das Game-Lap, oder der 315er Raum, dann  
333 muss man sich einfach an den zuständigen Raummanager oder Fachbereichsleiter wenden.

334 Frick:

335 Was meinst du mit der Rolle des Managers genau?

336 Randelshofer:

337 Ja es ist so, ich habe eine erweiterte Methode von Scrum von Boris Gloger gelernt. Der Boris  
338 Gloger geht über die drei Grundrollen hinaus und sagt es gibt neben Product Owner, Scrum Mas-  
339 ter und Entwicklerteam noch den Manager, den User und den Kunden. Als den Kunden würde ich  
340 immer den Lehrer einteilen. Dieser kommt ja auch und sagt, dass er das haben will. Der einzige  
341 Unterschied zu einer echten Kundenrolle ist der, das der Kunde Anforderungen stellt und mit dem  
342 Product Owner zusammenarbeitet. Er sagt dem Product Owner immer was er haben will. In der  
343 realen Wirtschaft ist der Kunde derjenige der bezahlt. An der FH ist es aber eine Mischrolle, weil  
344 man eigentlich den Kunden nicht hat. Du hast jemand der dir einen Auftrag gegeben hat, in eurem  
345 Fall ist das beispielsweise Frau Jellinek. Mit ihr sitzt ihr in irgendwelchen Meetings zusammen und  
346 sie sagt, dass sie von euch eine Website abgeliefert bekommen will. Man hat dadurch ganz klar  
347 einen Auftraggeber. Man hat aber niemanden der bezahlt und dir als Student werden die Formen  
348 der Umsetzung offen gelassen. Also ist die Kundenrolle eine ein wenig veränderte Rolle. Die Rolle  
349 des Users ist absolut klar. Der User ist der Endbenutzer. Im vorigen Beispiel wären das die Leute,  
350 die die Website im Internet anschauen. Der User ist derjenige, den ihr hin und wieder mal beim  
351 Sprint Planning 1 einladet um ihn zu fragen wie diese User-Story am besten ist. Ihr wollt ja Feed-  
352 back, da ist die Userrolle sehr wertvoll. Und beim Review zeigt ihr die neuen Ergebnisse her, und  
353 der User ist auch dabei und gibt wieder Feedback. Der Scrum Master ist eh klar, das Entwick-  
354 lerteam ist auch klar. In eurem Fall müsst ihr halt die MMA-Abteilung als Outsourcing-Bereich  
355 ansehen, da sie ja nicht bei Scrum wirklich mitmachen. Dann muss es allerdings einen geben, der  
356 die Schnittstelle zwischen Outsourcing und Entwicklerteam macht.

357 Frick:

358 Ja bis jetzt in unserem Fall macht die Schnittstelle unser Scrum Master mit dem Leiter der MMA-  
359 Webabteilung. Die MMAler sind nämlich nochmals in Web und Print unterteilt. Und ganz oben  
360 drüber steht der eigentliche Projektleiter. Wir haben uns da selber etwas zusammen gebastelt.

361 Randelshofer:

362 Diese Situation ist eher eine komplexe Situation. Diese sollte in einem 3-4 Stunden langen Mee-

363 ting einmal ordentlich durch besprochen werden und alle Verantwortlichkeiten festgelegt werden.  
364 Wichtig ist das die Entscheidung solcher Verantwortungsbereichsfestlegungen kommuniziert wer-  
365 den. Ja und der Manager ist soweit auch klar, dass ist wie gesagt die Facility, in diesem Fall die  
366 FH. In einer normalen Firma wäre das der Typ, der schaut, dass beispielsweise das Büro da ist,  
367 das der Computer da steht, das der Internetanschluss funktioniert oder ähnliches. Das jeder gut  
368 arbeiten kann und das die Stromrechnung bezahlt wird. Und das das Ganze am Schluss rentabel  
369 ist und wir nicht zu viel Geld ausgegeben haben. Und jetzt kommen wir wieder zurück zu den Rol-  
370 len. Die schwierigste Rolle innerhalb des FH-Settings ist für mich die Rolle des Product Owners.  
371 Der Product Owner ist derjenige, der mit dem Kunden redet und die Sachen klar macht. Diese  
372 zwei erstellen zusammen den Backlog. Die zwei priorisieren zusammen. Die Frau Jellinek wird  
373 sich nicht mit euch hinsetzen und priorisieren, um nochmals auf das Beispiel zurückzukommen.  
374 Sie wird vielleicht ein paar Deadlines setzen, aber der Rest muss von euch kommen. Der Product  
375 Owner muss immer schauen, dass der Backlog schön hergerichtet und priorisiert ist. An der FH  
376 ist das schwierig. Der Product Owner seit ihr im Prinzip selber und hier ergibt sich dann Rollen  
377 technisch ein Problem. Ihr im Team seit die die entscheiden was am Ende alles vorhanden sein soll.  
378 Gleichzeitig seit ihr diejenigen die entscheiden wie alles umgesetzt werden soll. Hier kann dann  
379 ein Konflikt entstehen. Es muss klar kommuniziert werden, dass es die Product Owner Rolle gibt,  
380 und am Ende muss einer die Entscheidung treffen und diese Rolle einnehmen. Einer muss für die  
381 Vision die Verantwortung übernehmen, und der bekommt dann die Rolle des Product Owners.  
382 Wenn man das an das gesamte Team übergibt, wird es unglaublich komplex und schwierig, da  
383 jeder eine eigene Meinung hat und diese am liebsten durchsetzen möchte. Jeder hat ein eigenes  
384 Bild vom Endergebnis. Den entscheidenden Punkt warum man den Product Owner braucht ist,  
385 dass derjenige die Vision zusammenhalten muss. Aus allen verschiedenen Inputs muss ein Output  
386 entstehen. Der Product Owner muss die Verantwortung vom Team zugeschanzt bekommen. Die  
387 Entscheidung muss im Endeffekt von einer Person getroffen werden. Es braucht eine letzte In-  
388 stanz. Alles was ich dir hier sage, ist schon das Tweaking, die kleineren Modifikationen. Hier sind  
389 wir nicht mehr bei den echten Rollendefinitionen, aber hier an der FH gehts leider nicht anders.  
390 Beim Product Owner ist immer schwierig herauszufinden wer das jetzt genau ist, wer diese Rolle  
391 genau übernehmen soll. Ich habe den Versuch gestartet, dass es zwei Vision Keeper gibt, die dann  
392 vom Team die Entscheidungen zusammenfassen und einer entscheidet dann am Ende. Wichtig ist  
393 aber, dass diese Entscheidungsperson vom Team selber ernannt wird. Man braucht deshalb einen  
394 im Team der seine technischen oder art Ansprüche gut formulieren kann und dem die restlichen  
395 vertrauen. Das ist aber schwierig zum finden. Man wird im Rahmen der FH leider keinen echten  
396 Product Owner bekommen, man muss eine getweakte Variante dieser Rolle verwenden. Es muss  
397 einer praktisch gewählt werden, und dieser muss von allen akzeptiert werden, das ist besonders  
398 wichtig.

399 Frick:

400 Wären hier eine Product Owner Schulung oder eine Scrum Master Schule im Rahmen der FH  
401 sinnvoll? Beispielsweise als Workshop neben den normalen Lehrveranstaltungen?

402 Randelshofer:

403 Natürlich wäre es sinnvoll, aber außer mir und der Frau Gucher gibt es keinen der Scrum ver-  
404 steht oder gelernt hat oder beispielsweise eine Scrum Master Ausbildung hat.

405 Frick:

406 Also wird es wieder schwierig von den Rahmenbedingungen her?

407 Randelshofer:

408 Ja, da wird es wieder schwierig von den Rahmenbedingungen her. Mit der Frau Gucher ist es  
409 bereits so besprochen, dass es in Zukunft im Rahmen des Unterrichts mehr hergenommen wird.  
410 Aber hier bin ich mir nicht ganz sicher ob stundenplanmäßig die Gegebenheiten schon da sind. Im  
411 Moment mache ich das ja auch eher als eine Art Extrawurst. Ich würde für die größeren Game-  
412 und Webprojekte aber auf jeden Fall Scrum vorschlagen, da beide Bereiche sehr softwarelastig

413 unterwegs sind.

414 Frick:

415 Also würdest du es befürworten, wenn Scrum in Game- und Webprojekten mehr eingesetzt werden  
416 würde?

417 Randelshofer:

418 Ja. Aber ich weiß es gibt genügend Stimmen dafür und dagegen. Die Stimmen dagegen sind  
419 meisten die, die es zu einem technischen Prozess, Ablauf verkommen lassen und die Grundzüge  
420 vernachlässigen. Die Grundzüge, die ich bei Scrum als wichtigstes finde sind „Transparenz“,  
421 „Ehrlichkeit“ und „Vision Keeping“. Darunter verstehe ich den Austausch und die Diskussion unter-  
422 einander. Es wird im Scrum immer miteinander geredet und diskutiert, sei dies jetzt im Sprint  
423 Planning 1 oder im Sprint Planning 2 oder andere. Bei Scrum steht immer das Team im Mittel-  
424 punkt. Es geht darum, dass Experten ihre Arbeit so erledigen können, wie sie es für angebracht  
425 halten. Die Daily-Scrum Meetings schaffen unglaubliche Transparenz, jeder weiß immer, dass die  
426 anderen auch arbeiten. Das hat ein bisschen etwas mit Kontrolle zu tun, aber nicht mit der Art  
427 Kontrolle, dass man nach drei Monaten einmal nachschaut was der andere gearbeitet hat. Durch  
428 das Daily-Scrum entsteht eine Art Mini-Kontrolle, die jedoch nicht als negativ, sondern positiv von  
429 den Team-Mitglieder aufgefasst wird. Scrum ist die Transparenz, offene Ehrlichkeit und dadurch  
430 wird alles überschaubar.

431 Frick:

432 Wir MMTler haben zum Beispiel Fächer wie Software-Engineering, bei denen wir den theoretischen  
433 Hintergrund zu Scrum lernen. Die MMAler haben, so weit ich jetzt weiß, keine vergleichbaren  
434 Lehrveranstaltungen oder Unterrichtsgegenstände. Macht es hier Sinn, dass vom Curriculum her  
435 Änderungen vorgenommen werden, dass hier ein Ausgleich stattfindet?

436 Randelshofer:

437 Wer unterrichtet bei euch Scrum, wenn ich fragen darf?

438 Frick:

439 In der Lehrveranstaltung Software Engineering hatten wir den Herrn Angleitner.

440 Randelshofer:

441 Das ist schwierig und vor allem ein sehr komplexes Thema. Hier geht es natürlich um Re-  
442 Akkreditierungen und Austausch von curricularen Sachen, das ist nicht so leicht zu sagen. Es gibt  
443 Selbstmanagement- und Teamprozessgeschichten beim MMA Bachelor, vielleicht wäre es sinnvoll  
444 das hier einmal unterzubringen. Ich weiß jetzt aber auch nicht an welcher Stelle. Mir sind die  
445 curricularen Sachen und der Ablauf grundsätzlich schon klar, aber ich würde einmal sagen, dass  
446 das schwierig ist. Das QPT 1 ist bei euch auch ein Einzelprojekt oder?

447 Frick:

448 Ja.

449 Randelshofer:

450 Ok. Vielleicht wäre es gut beim QPT 2a die Leute einfach mal rennen zu lassen. Beim QPT  
451 2b Scrum einmal als Prozess vorzustellen, und schauen ob das Ergebnis dadurch besser wird. Und  
452 beim QPT 3 komplett im Scrum-Prozess zu arbeiten. Ich könnte mir vorstellen, dass im 3. oder 4.  
453 Semester Scrum irgendwie eingebunden werden kann. Diese Frage ist jetzt aber sehr speziell für  
454 MMA und MMT und schwierig zu beantworten. Wenn man das einmal auf andere Unis ausbreiten  
455 würde, sollte es sicher unterrichtet werden. In was genau für einem Rahmen ist aber sehr schwierig.  
456 Also um den Scrum-Prozess zu verstehen braucht man nicht allzu lange. Es ist eigentlich ein sehr  
457 einfacher Prozess, aber den Grund zu verstehen warum man ihn anwenden sollte, das ist das was  
458 länger dauert.

459 Frick:

460 Und von der Infrastruktur her, beispielsweise einen eigenen Projektraum und dergleichen. Wie  
461 könnte man hier die Rahmenbedingungen an der FH verbessern? Es kommt jetzt ja zwar ein  
462 Zubau, aber trotzdem ist es ein Problem wo man jetzt seine Charts aufhängt, oder das Product  
463 Backlog organisiert, sodass jeder vom Team unabhängig diese Sachen in Betrachtung nehmen  
464 kann.

465 Randelshofer:

466 Das ist natürlich schwierig. Gerade an der FH ist das von der Infrastruktur her schwierig zu  
467 sagen. Einmal anders angefangen: Früher wo ich hier noch studiert habe, war das noch anders. Da  
468 hieß es, wenn Projekte in Zusammenarbeit gemacht werden wollen, können sie das getan werden,  
469 wenn nicht, dann nicht. Sprich es wurden nie wirklich Räume benötigt. Die Sache ist erst jetzt ge-  
470 fördert worden. Man merkt jetzt, dass die Leute gerne Räume hätten. Allerdings gibt es schon seit  
471 langem andere Unis die auch mit Projekten arbeiten und die hatten nie Räume. Studenten können  
472 sich auch am Campus in einem Zimmer treffen und dort eine Pinnwand an die Wand hängen. Wir  
473 haben auch mit großen Projekten gearbeitet, und wir haben es auch irgendwie geschafft. D.h. ein  
474 Raum ist ein gutes Zuckerl, und wenn Räume vorhanden wären, dann wäre es perfekt. Aber auf  
475 der anderen Seite muss ich sagen, wenn man am Campus ist, dann klebt man dort eine Wand mit  
476 Post-Its voll und trifft sich dort. Ich glaube nicht, dass es an der Räumlichkeit prinzipiell scheitert.  
477 Aber es wäre definitiv besser, wenn schöne kleine Projekträume vorhanden wären und man darin  
478 arbeiten kann. Der Zubau wird eh welche haben, und der Zugang wird sowieso relativ flexibel wer-  
479 den. Man kann aber auch elektronische Tools hernehmen wie den Pivotaltracker. Das wäre auch  
480 eine Möglichkeit sich zu organisieren. Ja, schöner ist es, wenn man einen Raum hat, das sagt auch  
481 Scrum. Mach die Meetings zur selben Uhrzeit, am selben Ort und unter den selben Bedingungen  
482 um den Prozess aufrecht zu erhalten. Vorher habe ich ja gesagt die Meetings, die Rollen und die  
483 Artefakte machen Probleme. Hier wären wir jetzt gerade bei den Artefakten. Für die Pinnwand  
484 muss Platz vorhanden sein, und es könnte schwierig werden an einer Hochschule, weil einfach nicht  
485 davon ausgegangen wird, dass Studenten Räumlichkeiten zur Verfügung gestellt werden. Auf der  
486 anderen Seite muss ich dazusagen, dass wir eh hartnäckig daran arbeiten, dass die Räume etwas  
487 werden, und das es Räume gibt. Es sind mittlerweile auch schon viel mehr vorhanden als früher.  
488 Bei mir gab es früher einfach keine Projekträume. Es wäre aber auch keiner auf die Idee gekommen  
489 einen Raum anzufordern. Ich muss aber auch dazusagen, dass mittlerweile jedes kleine Projekt,  
490 dass zum Beispiel nur zwei Wochen zusammenarbeitet, auch einen eigenen Raum haben will. Das  
491 kann nicht funktionieren. Das Projekt muss für einen Raum auch eine gewisse Wichtigkeit haben.  
492 Hier würde ich Projekten welche einen Prozess einsetzen wie Scrum zuerst einen Raum geben, weil  
493 es bei diesen einfach mehr Sinn macht. Das Problem ist auch, dass die Studiengänge MMA und  
494 MMT auch innerhalb einer Struktur sind. Es gibt noch andere Studiengänge an der FH und diese  
495 können nicht hinaus geschmissen werden. Ich würde auch vorschlagen Projekträume zu teilen. Drei  
496 Teams haben einen Projektraum und man muss sich einfach organisieren.

497 Frick:

498 Dann bin ich eh schon fast am Ende meiner Fragen. Aber eine Frage hätte ich noch. Wie könn-  
499 te man jetzt weitere Hindernisse, wie eben keine Vollzeitanzwesenheit, Ablenkungen durch andere  
500 Projekte, Prüfungen, etc. beseitigen bzw. umgehen?

501 Randelshofer:

502 Das ist ganz einfach. Scrum verwenden.

503 Frick:

504 Ok.

505 Randelshofer:

506 D.h. die grundsätzliche Idee von Scrum sollte man verwenden, das was Scrum grundsätzlich vom  
507 Prozess beinhaltet. Ich habe ja meinen Backlog und dafür gibt es das Sprint Planning 1 und hier

508 wird ja darüber gesprochen was für Sachen im nächsten Sprint gemacht werden sollen. Hierbei  
509 wählt ja das Team selbständig aus, welche Backlog-Items diesen Sprint bearbeiten werden. Dann  
510 ist es ganz einfach, wenn ich weiß, dass nächste Woche Urlaub ist, oder wenn ich weiß, dass nächste  
511 Woche Prüfung ist, dann nehme ich mir einfach nicht soviel von den Backlog-Items. Dazu ist der  
512 Prozess genau vorhanden. Das gleiche gilt ja auch im Falle des Auftreten von Bugs. Das ist reine  
513 Einteilungssache und genau dafür ist Scrum da. Und natürlich gibt es irgendwann eine Deadline  
514 und die Sachen müssen fertig sein, hierfür ist es aber einfach wichtig richtig zu priorisieren. Im  
515 Notfall müssen die unwichtigen Features weggelassen werden. Man hat innerhalb von Scrum dafür  
516 genau Funktionen um das zu steuern.

517 Frick:

518 Mein Ziel ist es ja am Ende dieser Bachelorarbeit einen Fahrtenplan für zukünftige Projekte aufzu-  
519 stellen, wie diese vorgehen müssen um Scrum sinnvoll einzusetzen. Hast du hier noch irgendwelche  
520 Tipps, Tricks, Anmerkungen, habe ich noch Dinge bei meinen Überlegungen vergessen?

521 Randelshofer:

522 Hier kann ich nur ein paar Sachen sagen, die mir gerade spontan einfallen: Lass die Teams selber  
523 entscheiden, lass es selber entscheiden wer welche Rollen übernimmt. Als Tipp an das Team, lass  
524 sie überlegen wer welche Rollen übernehmen soll. Das muss vom Team abgesegnet sein. Man kann  
525 niemanden zu etwas befördern, dass das Team eigentlich nicht haben will. Am Anfang am besten  
526 ein 2-3-4 Stundenmeeting abhalten, bei dem man sich hin setzt und alles in Ruhe durch bespricht.  
527 Hier sollte zuerst Scrum vorgestellt werden und anschließend alle gefragt werden ob der Einsatz bei  
528 genau diesem Projekt Sinn macht und wer welche Rolle übernehmen kann. In Diskussionen wird  
529 dann wahrscheinlich herauskommen wer was am besten machen kann. Scrum ist niemals in Stein  
530 gemeißelt. Scrum kann immer verändert werden, es ist ja ein agiler Prozess. Wenn man merkt,  
531 dass ein zwei Wochen-Rhythmus zu stressig ist, dann macht man einen drei Wochen-Rhythmus  
532 daraus. Wenn man merkt, dass die Rolle des Product Owner doch nicht die richtige Person war,  
533 dann muss darüber geredet werden und eventuell Veränderungen vorgenommen werden. Das ist  
534 das schöne bei Scrum, durch den agilen Prozess können Sachen diskutiert, verändert und dadurch  
535 verbessert werden. Das waren eh schon wichtige Punkte. Ken Schwaber hat ja sowas ähnliches  
536 gesagt wie, dass Scrum nur eine Hilfestellung ist komplizierte Projekte irgendwie zu regeln. Man  
537 sollte daher nicht den Scrum-Prozess über das Projekt stellen. Der Scrum-Prozess soll das Projekt  
538 führen und leiten und eine organisatorische Ebene bilden, aber wenn am Schluss der Scrum-Prozess  
539 ein größerer Aufwand wird, oder ein größeres Hindernis, dann machst natürlich auch keinen Sinn.

540 Frick:

541 Ok, dann wäre ich von meiner Seite aus fertig und möchte mich noch für das Gespräch bedanken.

542 Randelshofer:

543 Jaja, das passt schon. Bitte.

Transkript des Interview zwischen dem Experten Randelshofer Stefan und dem Autoren Frick  
Matthias, 2011, Puch bei Hallein

## B E-Mails

Dieser Abschnitt enthält den E-Mail Verkehr zwischen dem Autor und der Betreuungsperson Frau DI Brigitte Jellinek. Diese E-Mails beinhalten FHS-spezifische Fragen und Informationen, welche für die vorliegende Bachelorarbeit von hoher Bedeutung sind.

E-Mail am **14. Juni 2011, 17:58 Uhr** von Matthias Frick (Autor) an Frau DI Brigitte Jellinek (Betreuerin):

544 Hallo Brigitte!

545 Hast du die Fragen in deinem E-Mail-Postfach wieder gefunden betreffend meiner Bachelorarbeit2?

546 Ich habe bei meinen Notizen folgende Stichpunkte mitgeschrieben:

547 1. Wie funktionieren die Projekte an der FH?

548 2. Was gibt es für Projekte? (Realprojekte, Forschungsprojekte, etc.)

549 3. Rahmenbedingungen von Projekten an der FH?

550 4. Was gibt es für Projektkunden?

551 5. Wieviele Projekte laufen an der FH? QPT2, QPT2a, QPT2b, QPT3, Masterprojekte? (für  
552 die Berechnung der benötigten Räume an der FH)

553 Könntest du mir diese Fragen noch beantworten? Und könntest du bitte nochmals nachschauen  
554 ob du deine Mitschrift noch findest?

555 Wo finde ich den Leitfaden für MMA Masterprojekte? Ich habe ihn immer noch nirgends entdeckt.  
556 Für MMT Masterprojekte existiert noch kein Leitfaden so weit ich weiß, stimmt das?

557 Schöne Grüße

558 Matthias

E-Mail am **16. Juni 2011, 14:54 Uhr** von Frau DI Brigitte Jellinek (Betreuerin) an Matthias Frick (Autor):

559 Liebe Matthias,

560 hier sind die Antworten auf Deine Fragen:

561 die häufigsten Projekte sind rein studentische Projekte: Die StudentInnen entwickeln die Idee  
562 selbst. Es gibt aber auch Auftragsarbeiten: entweder FH-intern, oder extern.

563 Ein FH-Internes Projekt ist z.B. das Portfolio der Studiengänge MMA + MMT: es wurde auf  
564 meine Initiative hin durchgeführt, hier bin ich (als Lehrende) also Auftraggeberin + Kundin.

565 Aus den Forschungs-Projekten des Studiengangs gibt es auch immer wieder Projektaufträge: z.B.  
566 war Robert Praxmarer der Auftraggeber + Kunde für das AR-Projekt s'bickt  
567 <http://portfolio.multimediaart.at/projects/2011-sbickt>

568 Wenn jemand von ausserhalb der Studiengänge MMA + MMT mit einer Projektidee an unser  
569 heran tritt, dann kann es auch Projekt-Aufträge geben. Allerdings nur dann, wenn das Projekt

570 zu dem passt, was die Studierenden gerade lernen sollen, und wenn die ökonomischen Rahmen-  
571 bedingungen passen: für gemeinnützige Vereine und ähnliches werden Projekte schon für geringe  
572 Beträge gemacht (z.B. ein paar hundert Euro Preisgeld). ein Beispiel dafür:  
573 <http://portfolio.multimediaart.at/projects/2010-salzburger-heimatvereine>

574 Wenn der Auftraggeber aber eine „ganz normale“ kommerzielle Firma ist, dann verlangt die FH  
575 auch ganz normale Markt-Preise. (das ist sehr wichtig: wir wollen ja für die AbsolventInnen von  
576 MMA + MMT nicht die Markt-Preise verderben).

577 Dann gibt es meist ein Modell, bei dem a) ein Konzept oder Prototyp im Unterricht entsteht und  
578 benotet wird, wofür die LektorIn eine Aufwandsentschädigung für die zusätzliche Kommunikation  
579 mit der externen KundIn erhält, und die FH für zur Verfügung gestellte Geräte / Software, etc.

580 b) anschließend die StudentInnen für die Umsetzung direkt von der KundIn angestellt oder beauf-  
581 tragt werden. Das kann z.B. ein Praktikum sein, oder einfach ein Sommer- oder Neben-Job, auf  
582 jeden Fall aber ganz normal bezahlt.

583 Beispiele dafür: <http://portfolio.multimediaart.at/projects/2009-ace-of-mace>

584 Die Anzahl der QPT- und Master Projekte kann man aus der Team-größe und der Zahl der  
585 Studierenden ausrechnen:

586 MMA-B: 60 - 65 Personen pro Jahrgang MMA-M: 45 Personen pro Jahrgang MMT-B: 30 - 35  
587 Personen pro Jahrgang MMT-M: 20 Personen pro Jahrgang

588 QPT1: Einzelprojekte QPT2a: nur bei MMT Pflicht, 2 Personen pro Projekt QPT2b: bei MMT  
589 typischerweise 2 - 3 Personen pro Projekt, bei MMA 2 - 6 QPT3: interdisziplinär, 3 bis 10 Personen  
590 pro Projekt Masterprojekte: interdisziplinär, 3 bis 10 Personen pro Projekt

591 Es gibt noch keinen Leitfaden für Masterprojekte, es wird nur einen gemeinsamen MMA & MMT  
592 geben.

593 Die Projekte sind sehr ähnlich dem QPT3, nur die zeitlichen Rahmenbedingungen sind anders.  
594 z.B. MMT-M2011:

595 September - Dezember: Ideenentwicklung Dezember: Ideen-Pitch, hier fällt die Entscheidung wel-  
596 che Ideen weiterentwickelt werden dürfen Jänner: Ressourcen-Pitch, hier muss zu jedem Projekt  
597 ein Projekt-, Zeit- und Finanzplan vorliegen, sowie Anträge auf z.B. Kameron, Räume, Förder-  
598 stipendien, etc. Februar - Mai: Konzeptentwicklung, Prüfung der technischen Machbarkeit, Pro-  
599 totypen, Schreiben des Drehbuchs, ... Juni: Konzept-Galerie zeigt fertige Konzepte, Drehbücher  
600 ==> Note für 2.Semester Juli - Dezember: Umsetzung Jänner: Projekt ist fertig, alle Features  
601 vorhanden, interne Präsentation ==> Note für 3.Semester April: "Release-Woche": hier finden die  
602 Film-Premieren, Ausstellungen, Aufführungen, Launch-parties, etc. statt ==> Note für 4.Semes-  
603 ter