Scrum im Praktikumseinsatz der Abteilung BIS

Simon Johanning, Hans-Gert Gräbe

Version vom 13. Juni 2015

Zusammenfassung

Dieses Dokument **Scrum.pdf** gibt einen Überblick über die Prozesse und Artefakte der Scrum-Methodik. Es ist für Teilnehmer/innen und Tutor/inn/en in Praktika der Abteilung BIS gedacht, die nach der Scrum-Methodik organisiert sind und berücksichtigt die terminologischen Änderungen, die in Auswertung der Erfahrungen aus dem SWT-Praktikum 2015 vorgenommen wurden.

Inhaltsverzeichnis

1 Einleitung

Scrum ist eine agile Methode der Projektarbeit, in der die selbstständige Arbeit des *Entwicklerteams* (kurz *Team*) im Vordergrund steht, welches begleitet vom *Scrum Master* ein Projekt im Auftrag des *Project Owners* (Projektträger) umsetzt.

Ziel von Scrum ist die ressourceneffiziente und qualitativ hochwertige Umsetzung eines Projekts, das einer zu Beginn formulierten *Vision* entspricht und diese verfeinert.

Die Umsetzung der *Vision* in das fertige Release erfolgt durch das Ausformulieren von klaren Funktionalitäten aus Anwendersicht, die in den *Sprints* iterativ und inkrementell umgesetzt werden.

In einem solchen *Sprint* – einer etwa vierwöchigen Organisationseinheiten – wird eine während des *Sprint Planning* besprochene und im *Sprint Backlog* zusammengefasste Menge von *User Stories* (Implementierungsaufträge) vom *Team* umgesetzt.

Gängigen Software-Entwicklungmodelle gehen heute von einer Strukturierung nach Aktivitäten und Meilensteinen aus. Da in unserem Kontext ein Sprint im Wesentlichen einer solchen Aktivitätsphase entspricht, werden zu einem Sprint weiterhin ein *Sprint Goal* (Zusammenfassung des zu erreichenden Ziels in zwei Sätzen) sowie *Sprint Artefakte* (herzustellende Arbeitsprodukte) spezifiziert.

Am Ende eines jeden Sprints steht die Qualitätsprüfung der hergestellten Artefakte, die im *Sprint Review* in Form eines Meilensteins durchgeführt wird.

Anschließend werden die Artefakte in das *Project Increment* übernommen.

2 Rollen und Verantwortlichkeiten

Die *Scrum-Methodik* kennt im Wesentlichen drei Rollen: den *Product Owner*, den *Scrum Master* und das *Entwicklungsteam*, im Weiteren kurz *Team*.

2.1 Project Owner

Der *Project Owner* ist für die strategische Projektsteuerung zuständig. Er verantwortet die Konzeption und Mitteilung einer klaren Projekt-*Vision*, die Festlegung und Priorisierung der jeweils zu entwickelnden Artefakte (im *Project Backlog*) sowie die Entscheidung darüber, ob die vom *Team* am Ende jedes Sprints gelieferte Funktionalität akzeptabel ist.

Der *Project Owner* ist verantwortlich für die Formulierung der *Vision* und das Führen des *Project Backlog*s.

2.2 Scrum Master

Der Scrum Master ist für die Einhaltung der *Scrum*-Regeln sowie die Organisation und Durchführung der *Scrum Prozesse* zuständig. Weiterhin hat der *Scrum Master* die Aufgabe, externe Hindernisse (*Impediments*) aus dem Weg zu räumen, die sich dem *Team* in den Weg stellen. Dazu arbeitet er mit dem *Team* eng zusammen, gehört aber selbst nicht zum Team.

Der *Scrum Master* ist verantworlich für das Führen des *Impediment Backlog* sowie die Organisation und Moderation der *Scrum Prozesse*.

2.3 Team

Das *Team* ist für die Lieferung der Projektartefakte in der vom *Project Owner* im *Project Backlog* spezifizierten Priorisierung zuständig. Zudem trägt das *Team* die Verantwortung für die Einhaltung der zuvor vereinbarten Qualitätsstandards. Das *Team* entscheidet selbst, wie viele und welche Funktionalitäten es in einem *Sprint* liefern will. Hierzu geht es im *Sprint Planning* entsprechende *Commitments* ein.

In Scrum tritt das *Team* immer als Einheit auf – gute und schlechte Ergebnisse verantworten nie einzelne Mitglieder, sondern immer das *Team* als Ganzes. Das ideale Teammitglied ist sowohl Spezialist als auch Generalist, damit es Teamkollegen beim Erreichen des gemeinsamen Ziels helfen kann.

Das *Team* ist verantwortlich für das Führen der *Sprint Backlog*s.

3 Konzepte

3.1 Sprint

Im *Sprint* werden die vereinbarten Projektartefakte entwickelt. Am Ende jedes Sprints steht die Lieferung fertiger Artefakte mit Funktionalitäten, wie sie im *Sprint Planning Meeting* zusammen mit dem *Product Owner* als *User Stories*, *Epics* oder *Themen* festgelegt und im *Sprint Backlog* festgehalten wurden.

Im weiteren *Sprint Planning* verfeinert das Team diese *User Stories* zu *User Tasks*, die in einer Planungseinheit (Woche) umgesetzt werden können, ordnet diese einzelnen Verantwortlichen zu und vermerkt dies im *Sprint Backlog*. *User Stories* mit der höchsten Priorisierung werden zuerst bearbeitet. Werden die *User Stories* in *Themen* zusammengefasst, so erfolgt eine solche Priorisierung innerhalb jedes Themas.

Bei einem aktuellen und überlegt geführten *Sprint Backlog* weiß jeder im Team, welche Funktionalität gerade entwickelt wird, und man ist in der Lage, sich gegenseitig auszuhelfen. Das Entwicklungsteam macht sich erst dann an die Bearbeitung der nächsten Funktionalität, wenn alle zuvor vereinbarten Akzeptanz- und Testkriterien erfüllt und die Funktionalität damit tatsächlich fertig ist. Dadurch wird verhindert, dass am Ende des *Sprints* die Bearbeitung vieler Funktionalitäten begonnen wurde, aber keine davon fertiggestellt ist.

Während des Sprints trägt das *Team* allein die Verantwortung, das vereinbarte Ziel zu erreichen. Um diese Verantwortung auch wahrnehmen zu können, muss das *Team* vor Störungen geschützt werden. Von außen herangetragene, zusätzliche Aufgaben gehören in den Verantwortungsbereich des *Scrum Masters*.

3.2 User Stories

User Stories beschreiben eine Softwareanforderung aus der Sicht eines Anwenders „*Als ... möchte ich ..., um ...* “ sowie die zugehörigen Akzeptanzkriterien.

User Stories sind bewusst kurz gehalten und sollten (neben den Akzeptanzkriterien) nicht mehr als 2–3 Sätze umfassen. Sie sind prosaisch aus Sicht zukünftiger Anwender geschrieben.

User Stories werden vorrangig verwendet, um eine Basis für Aufwandsschätzungen zu schaffen.

3.3 User Tasks

User Tasks beschreiben einen konkreten Auftrag an ein Mitglied des *Teams* mit einer konkreten Schätzung des Arbeitsaufwandes in Stunden. Dieser Auftrag wird nicht von außen zugewiesen; stattdessen übernehmen die Mitglieder des *Teams* selbstständig Verantwortlichkeiten für die *User Tasks*.

In Abgrenzung zu den *User Tasks* enthalten *User Stories* keine Spezifizierung bezüglich der Implementierung und sind eher in der Sprache der Anwender gehalten.

3.4 Epics

Epics sind Anforderungsbeschreibungen, ähnlich zu *User Stories*, unterscheiden sich jedoch in Umfang und Granularität von diesen. *Epics* sind deutlich umfangreicher und weniger strukturiert als *User Stories* und sind im weiteren Projektverlauf in mehrere *User Stories* zu verfeinern.

Epics entstehen entweder bei der Anforderungsanalyse, wenn eine Anforderung (noch) nicht in der Form einer *User Story* formuliert werden kann (bspw., weil die Anforderung noch nicht ganz klar ist), oder können zur Strukturierung während des *Backlog Grooming* als Aggregation verschiedener, zusammengehöriger *User Stories* entstehen.

3.5 Themen

Ein *Thema* (Themenbereich) ist eine thematische Strukturierungsebene, der *User Stories* oder *Epics* zugeordnet werden. *User Stories* oder *Epics* gleicher Domäne haben in der Regel das gleiche *Thema*.

Themen können in einer *Story Map* als „Mind Map“ erfasst werden, um den inneren Zusammenhang zwischen verschiedenen Themen darzustellen.

Beispiele für Themenbereiche sind:

> Backend, UI, Frontend, Technische Grundlagen usw.

oder auch

> Drehort X, Drehort Y, CGI, Post-Production, Music & Sound usw.

4 Scrum Dokumente

Einer der großen Vorteile von Scrum sind die sehr genauen Vorgaben zur Dokumentation der eigenen Prozesse, was sich als außerordentlich hilfreich für Teams erwiesen hat, in denen nicht immer alle bei jeder Teambesprechung dabei sind.

Die Scrum Dokumente dienen dem Team zur Orientierung

- über die Ziele und Prioritäten des Projektes (*Project Vision* und *Project Backlog*);
- über Ziele, Prioritäten und Ergebnisse der einzelnen *Sprints* (*Sprint Backlog und Burndown Chart*;
- über *User Stories*, deren Beziehungen und Abhängigkeiten ggf. in einer *Story Map* festgehalten werden;
- über den erreichten Projektstand (*Project Increment*) sowie
- über die Hindernisse der Projektarbeit (*Impediment Backlog*).

Die Scrum Dokumente werden in der Regel vom Team angelegt und verwaltet; dem *Project Owner* und dem *Scrum Master* kommen hierbei beratende Funktionen zu.

4.1 Project Vision

Die *Project Vision* stellt das Herz des Projektes dar. Sie beschreibt das überspannende Ziel, welches alle Beteiligten teilen, und soll die Frage des *warum* statt des *was* und *wie* beantworten. Die *Project Vision* umfasst die Essenz des Projekts und dient dazu, den Zweck und das Ziel des Projektes zu verstehen, um dem *Team* Orientierung zu geben. Die *Project Vision* ist die Basis für die Formulierung der einzelnen *Sprint Goals* durch den *Projekt Owner*.

Die *Projekt Vision* sollte möglichst frühzeitig spezifiziert und im weiteren Projektverlauf nicht modifiziert werden, um Anker des Projektes zu sein. Weiterhin liefert diese Vision Sortierkriterien für die Priorisierung im *Project Backlog*, dient als Basis für die Ableitung von *Epics* und macht projektkritische Entscheidungen vorhersehbarer.

4.2 Project Backlog

Das *Project Backlog* enthält eine priorisierte Liste der Anforderungen, die im Laufe des Projektes zu bearbeiten sind. Diese Items werden nach Bedeutung für den Erfolg des Projektes priorisiert und Zug um Zug als *User Stories* (und evtl. *Epics*) in den einzelnen *Sprints* bearbeitet.

Ein (priorisierter) Teil der Items des *Project Backlogs* wird während des *Sprint Planning Meetings* in das *Sprint Backlog* übernommen und im weiteren Verlauf des Sprint bis zu *User Tasks* heruntergebrochen.

Die Abhängigkeiten und Umsetzungsplanungen der einzelnen Items können in einem *Release Plan* erfasst werden als Orientierung, in welchem *Sprint* welches *Backlog Item* vom *Team* entwickelt wird. Es kann damit als die Basis für eine detaillierte und effiziente Planung der nächsten *Sprints* dienen und von großer Bedeutung im *Sprint Planning Meeting* sein.

Üblicherweise orientiert sich ein solcher *Release Plan* an den Erfahrungen im Entwicklungsprozess und ggf. den Ergebnissen eines *Backlog Groomings* und wird in einem *Release Planning Meeting* verfasst bzw. bearbeitet.

4.3 Sprint Backlog und Burndown Chart

Das *Sprint Backlog* umfasst die *Epics*, *User Stories* und *User Tasks*, die während des aktuellen *Sprints* vom *Team* abzuarbeiten sind.

Die Items des *Sprint Backlog* stammen von den Items aus dem *Project Backlog*, deren Bearbeitung sich das Team im *Sprint Planning Meeting* für den aktuellen *Sprint* verschrieben hat. Diese Items werden vom Team im Laufe des *Sprints* in *User Tasks* heruntergebrochen, für deren Bearbeitung einzelne Mitglieder des Teams Verantwortung übernehmen. Diese *User Tasks* müssen so kleingranular sein, dass sie einen Arbeitsaufwand von einigen Stunden nicht überschreiten. Damit das *Sprint Planning Meeting* zeitlich nicht ausartet, ist es möglich, die Taskplanung auch auf ein weiteres (zeitnahes) Treffen zu verschieben.

Abgearbeitete Tasks werden in die *Burndown Chart* übernommen zusammen mit einer Information über den Bearbeiter und die dafür aufgewendete Zeit. Wegen der großen inhaltlichen Nähe werden bei uns beide Teile in einem Dokument *Sprint Backlog und Burndown Chart* geführt.

Die *Burndown Chart* sollte wöchentlich (für den *Weekly Sprint*) aktualisiert und kurz während des *Weekly Sprints* besprochen werden.

4.4 Story Map

Die *Story Map* ist ein Scrum Dokument, welches die Stories auf verschiedenen Abstraktionsebenen anzeigt. Die Stories sind dabei meist nach *Themen* geordnet. Innerhalb eines *Themas* werden die Abhängigkeiten zwischen den *Epics* und *User Stories*, die zu diesem *Thema* gehören, aufgelistet und ggf. grafisch dargestellt.

4.5 Impediment Backlog

Im *Impediment Backlog* werden die Hindernisse aufgelistet, die dem *Team* im Weg stehen, aber vom Team selbst nicht beeinflusst werden können.

Das *Impediment Backlog* wird vom *Scrum Master* priorisiert und dient ihm als Basis dafür, dem Team „Steine aus dem Weg zu räumen“.

4.6 Stärken und Schwächen Profil

Das *Stärken und Schwächen Profil* ist kein Dokument im klassischem Sinne von *Scrum*, sondern wird vom Team als Sprint Artefakt im Zuge des „Team Building“ erstellt.

Eine gute Kenntnis der Stärken und Schwächen der einzelnen Teammitglieder ist für eine erfolgreiche Teamarbeit wesentlich, um die im Team vorhandenen Stärken optimal zur Geltung zu bringen, die vorhandenen Schwächen bestmöglich zu kompensieren und damit den Gesamtaufwand für das Team zu minimieren.

5 Scrum Prozesse

Vorbemerkung

In der Scrum-Literatur (insbesonders online) herrscht eine große Diversität an Vorstellungen über die verschiedenen Elemente, sodass es *die* Scrum-Methodik nicht gibt. Dies gilt insbesonders für die Abgrenzung von *User Stories* und *User Tasks*, weswegen auch die Beschreibung in diesem Dokument (insbesondere im *Project Backlog* und *Sprint Backlog*) teilweise nicht klar zwischen *User Stories* und *User Tasks* unterscheidet, sondern teilweise einfach von *Items* die Rede ist.

5.1 Sprint Planning Meeting

Das *Sprint Planning Meeting* ist eines der zentralen Treffen für den Erfolg eines *Sprints*. Es dient dazu, dass sich das *Team* einem *Sprint Goal* (Ziel) für den aktuellen *Sprint* verschreibt, ist zentral für die Erstellung des *Sprint Backlogs* und auf dessen Basis für eine Liste der in diesem *Sprint* zu erledigenden *User Tasks*.

Im *Sprint Planning Meeting* versucht das *Team* eine realistische Einschätzung zu finden, wie viel Arbeit im Rahmen dieses *Sprints* investiert werden kann, und übernimmt (ggf. auf der Basis des *Release Plans*) eine Menge von *User Stories* aus dem *Project Backlog* in das *Sprint Backlog*. Zentral ist für jedes zu übernehmende Item zu entscheiden, ob es im aktuellen *Sprint* umgesetzt werden kann.

Das *Sprint Planning Meeting* gliedert sich in zwei Teile:

- Im ersten Teil werden die (relevanten) Anforderungen aus dem *Project Backlog* vom *Project Owner* erläutert, und das *Team* diskutiert, ob die jeweilige Anforderung im aktuellen *Sprint* umgesetzt werden kann.

  Entscheidet das Team positiv, wird das Item in das aktuelle *Sprint Backlog* aufgenommen. Das Ziel dieses Teils ist es, aus dem *Project Backlog* ein *Sprint Goal* (Ziel) für den aktuellen *Sprint* abzuleiten, zu dessen Erfüllung sich das Team verpflichtet.
- Der zweite Teil des *Sprint Planning Meeting* dient – sofern das noch nicht mittels *Backlog Grooming* geschehen ist – dazu, das *Sprint Backlog* in *User Tasks* herunterzubrechen. Hierzu werden aus den *User Stories* einzelne *User Tasks* hergeleitet, mit einer Aufwandsschätzung versehen und einzelnen Teammitgliedern zugeordnet.

5.2 Sprint Review Meeting

Das *Sprint Review Meeting* ist ein *inhaltliches* Evaluationstreffen für den Abschluss des aktuellen *Sprints*. Im *Sprint Review Meeting* präsentiert das *Entwicklungsteam* die Artefakte, die im *Sprint* entstanden sind oder verändert wurden.

In diesem Meeting werden nur vollständig umgesetzte *User Tasks* präsentiert. Unvollständig umgesetzte Aufgaben werden nicht präsentiert, jedoch besprochen. Diese werden ins *Project Backlog* zurückgestellt und können im *Sprint Planning Meeting* für den nächsten *Sprint* in das zugehörige *Sprint Backlog* übernommen werden.

Das *Sprint Review Meeting* spielt die Rolle eines *Meilensteins*, in dem die erstellten Artefakte einer Qualitätsprüfung unterzogen werden sowie Bilanz gezogen wird, was im *Sprint* geschafft wurde und was als „Altlast“ in einen späteren *Sprint* übernommen werden muss.

5.3 Sprint Retrospective

Die *Sprint Retrospective* ist ein *organisatorisches* Evaluationstreffen für den aktuellen *Sprint*. Es dient als Reflexion der Arbeitsvorgänge im Team während des aktuellen *Sprints* und soll die Prozesse analysieren und verbessern.

Zentral stehen hierbei zwei Fragen:

- Was lief gut?
- Was kann verbessert werden?

Die verschiedenen Projektbeteiligen stellen dar, welche Prozesse aus ihrer Sicht gut bzw. schlecht gelaufen sind bzw. welche Artefakte hilfreich (oder weniger hilfreich) waren, und es wird nach Lösungen gesucht kommende *Sprints* effektiver zu gestalten.

5.4 Weekly Scrum

Der *Weekly Scrum* (als Modifikation des „Daily Scrum“) dient dazu, sich gegenseitig über den Fortschritt der Arbeit an den *User Tasks* zu informieren. Falls Ergebnisse vorgestellt werden, sollte sich darauf konzentriert werden, Funktionalitäten zu präsentieren.

Jedes Mitglied des *Entwicklungsteams* beantwortet die Fragen:

- Woran habe ich seit dem letzten *Weekly Scrum* gearbeitet?
- Was plane ich bis zum nächsten *Weekly Scrum* zu tun?
- Welche Hindernisse haben sich ergeben?

Der *Weekly Scrum* sollte möglichst kurz gehalten werden und keine Diskussionen zum *Project Backlog*, *Sprint Backlog*, zur *Story Map* oder weiteren Scrum Dokumenten enthalten. Dazu dienen andere Treffen zu spezifischen Prozessen. Spezifische Probleme können (besonders, wenn sie nur einen Teil des Teams betreffen), anschließend im informellen Rahmen geklärt werden.

Der *Weekly Scrum* wird damit beendet, dass jedes Mitglied des *Teams* ankündigt, woran es in der nächsten Woche arbeiten wird.

5.5 Story Time

Story-Time-Treffen dienen dazu, im *Team* eine bessere Vorstellung zu den Items des *Project Backlogs* zu entwickeln. In diesem Treffen „erzählt“ der *Project Owner* eine „Story“, worum es in den Items des *Project Backlogs* geht, und das Team hat die Möglichkeit, Fragen zu stellen, um zu verstehen, was mit diesen Items genau gemeint ist.

Die *Story Time* soll Klarheit über Motivation und Hintergrund der *Project Backlog Items* geben und so Missverständnissen entgegenwirken. Weiterhin bietet die *Story Time* die Möglichkeit, Einschätzungen des *Teams* zu entwicklungsrelevanten Möglichkeiten und Restriktionen zu geben (bspw. Abhängigkeiten und Entwurfsbeschränkungen).

5.6 Backlog Grooming

Das *Backlog Grooming* ist eine Form konzentrierter Arbeit an der Strukturierung der Anforderungen und dient dazu, die *Story Map* zu verfeinern, *Epics* in *User Stories* herunterzubrechen, *User Stories* in *Epics* zu aggregieren oder *Themen* für *User Stories* in derselben Domäne zu finden.

Weiterhin liefert das Backlog Grooming die Möglichkeit, bestehende *User Stories* zu verfeinern, falsch klassifizierte *User Stories* oder *Epics* zu reklassifizieren oder verworrene *User Stories* zu entwirren und bei Bedarf in mehrere zu trennen.

5.7 Release Planning

Release Planning ist insbesondere für größere Projekte mit multiplen Abhängigkeiten für die Anforderungsstrukturierung auf der Ebene des *Project Backlogs* von Bedeutung.

Das *Release Planning Meeting* dient der Erstellung oder Überarbeitung des *Release Plans* und dem Priorisieren des *Project Backlogs* nach Nutzen und Aufwand. Zur Aufwandsschätzung werden oft *Story Points* verwendet, die z. B. durch Techniken wie Planning Poker zugewiesen werden können.