# Druchführung Scrum

In diesem Kapitel werden die acht Artefakte von Scrum: Release Planning Meeting, Product Backlog, Sprint Planning Meeting, Sprint Backlog,Sprint, Daily, Sprint Review und Retrospektiv, welche in der Durchführung umgesetzt wurden erläutert. Zusätzlich werden auch Probleme, so wie Besonderheiten welche auftraten und umgesetzt wurden geschildert.

## Release Planning Meeting

Am Anfang des Release Planning Meetings waren die globalen Anforderungen, welche für alle Projekte gleich sind, bereits gegeben, so mussten lediglich die speziellen Anforderungen für Scrum definiert werden. Dazu zählen:

* Die Countdown Funktion
* Auffüllen mit neuen Blöcken
* Erreichen einer minimal Punktzahl

Weiterhin wird die Sprintlänge auf die Dauer von einer Woche festgelet. Aufgrund der Anforderungen wurde anschließend ein Architekturmodell erstellt um Abhängigkeiten zu klären und sicherzustellen, dass keine grundlegenden Funktionen zu dieser Zeit vergessen wurden. Die Architektur im Planning beschreibt grob einen Kreislauf welcher jedoch auch Nebenläufigkeiten beinhaltet.

Architektur

So kommt man im Kreislauf von der Startseite zum ersten Level, vom Beendeten Level zur Ergebnisseite und von dieser entweder ins nächste Level oder zum speichern. Nach dem Speichern gelangt man zur Rangliste und von dort aus wieder zur Introseite. Zudem ist es ebenso möglich von der Introseite zur Rangliste zu springen. Weiterhin zählen hierbei zu den Nebenläufigkeiten das Abbrechen des Levels und das Auslassen des Speicherns welches beide eine Weiterleitung zur Introseite nach sich ziehen.

Aufgrund der vorgestellten Architektur und den Anforderungen wurde zusammen mit dem Product Owner eine Priorisierung der User Sories vorgenommen. Eine Priorisierung auf Basis einer bereits bestehenden Architektur hat den Vorteil, dem Product Owner bereits Abhänigkeiten verständlich aufzeigen zu können und diese in der Priorisierung auch zu berückksichtigen.

Im Release Planning Meeting soll zudem eine erste Definition of Done festgehalten werden. In dieser wurde beschlossen das eine User Story als erledigt „Done“ gilt, sobald alle Tasks innerhalb der jeweiligen User Story abgearbeitet sind und jeder Task von dem zweiten Teammitglied gereview wurde

## Product Backlog

Aufgrund der Priorisierung im Release Planning Meeting wurden die User Stories absteigend ihrer Priorität in den Produkt Backlog übernommen und in die beiden eTracking Programme Jira und Yodiz eingetragen. So wurde in diesem Projekt, die Einrichtung der Entwicklungsumgebung, die Konfiguration der Bords, die Grundlagenrecherche sowie das Design, besonders hoch Priorisiert. Dabei am niedrigsten priorisiert wurde das Anzeigen der Anleitung, die Anzeige der Bestenliste sowie das Speichern der Ergebnisse.

Sprint Planning Meetingstory

Die Abschätzung der User Story in Scrum erfolgt in Story points. Story Points sind keine feste Zeiteinheit sondern sie beziehen sich auf die kleinste User Story. Ein Story Point ist in Scrum also so groß wie die kleinste User Story. In diesem Projekt ist die kleinste User Story das Anlegen der Startseite.

Zum Abschätzen wurden für jede User Story welche in den Sprint gezogen werden sollte eine unabhänige Schätzung der Story Points gemacht werden. In dem jedes Teammitglied verdeckt eine Anzahl von Stoy Points welche den Fibonacci Zahlen entsprechen, auf eine Karte schrieb und diese gleichzeitig mit dem anderen Teammitglied vorzeigte.

Wenn beide Teammitglieder die gleiche Anzahl an Story Points geschätzt haben wird diese Anzahl in die User Story übernommen. Wenn jedoch eine unterschiedliche Anzahl geschätzt wurde, was deutlich häufiger auftrat erklärten beide die Gründe weshalb sie die Story Points so abgeschätzt hatten. Zu Angang des Projktes schrieb dann jedes Teammitglied erneut eine Anzahl an Story Points, welche dann erneut verglichen wurde. Wenn immer noch keine Einigung erzielt werden konnte wurde die größere der beiden Zahlen in die User Story übernommen. Ab circa der hälfte des Projektes wurde bei einer fehlenden Übereinstimmung der Story Points, wieder eine Erklärung jedes Mitglieds abgegeben, jedoch hat das Mitglied welche die höhere Zahl geschätzt hat angegeben ob sie die geschätzten Stroy Points nach unten korrigiert oder ob sie bei der Anzahl bleibt. Wenn das andere Teammitglied die Anzahl über die Story Points jetzt teilt wird diese Anzahl in die User Story übernommen, andernfalls die höchste der beiden Zahlen. Für die Plattform Yodiz war es zudem nötig gegen die Konventionen von Scrum auch die Tasks innerhalb der User Storys abzuschätzen. Dies musste hierbei in Minuten erfolgen. Innerhalb des Planing Meetings ist es auch möglich das der Product Owner, welcher in diesem Projekt aus Personalmange auch der Scrum Master war, eine Umpriorisierung vornimmt. Innerhalb des Projektes kam es zweimal zu einer Umpriorisierung. So wurde einmal die User Story Punkte anzeigen und Spiel abbrechen vertauscht. Zudem wurde die Anleitung, welche am Anfang eine hohe Priorisierung hatte vom Product Owner deutlich herabpriorisiert und wurde so in die letzte Iteration verschoben.

## Sprint Backlog

Im Sprint Backlog werden die User Storys, welche im Sprint Planing abgeschätzt wurden und im Sprint abgearbeitet werden festgehalten. Der Sprint Backlog stellt sicher das jedes Teammitglied weiß was in diesem Sprint für Aufgaben erledigt werden müssen und welche Aufgaben bereits erledigt sind. Zudem sieht man wer an welchen Aufgaben gerade arbeitet. Dies war in der Studienarbeit besonder wichtig, da an verschiedenen Standorten entwickelt wurde, um eine Doppelarbeit zu vermeiden. Zudem sind die User Stories die im Sprint Backlog festgehalten sind auch tatsächlich fest und können während einem Sprint nicht getauscht werden. Im Projekt gab es während eines Sprints keine Probleme mit den festgelegten User Stories.

Weiterhin muss erwähnt werden das die User Stories ihre Priorität wie auch im Product Backlog behalten und die Aufgaben so abgearbeitet werden müssen. Dies hat eher zu Problemen geführt denn wenn ein Teammitglied sich die wöchentliche Arbeitszeit eher am Ende der Woche eingeplant hat, aber verantwortlich war für einen der oberen User Stories hat dieses Mitglied die Arbeit des anderen Mitglieds behindert/ verhindert.

## Sprints

Die Länge der Sprints wurde wie in der Einleitung schon erwähnt auf eine Woche festgelegt. Dies Stellt einen Kontrast zu den anderen beiden Prozessen da, da mach jeder Woche ein Release zu machen war. Der Sprint beginnt jeweils Montags und endet jeweils Montags. Die tägliche Arbeitszeit wurde auf eine Stunde am Tag festgehalten, wobei auch zwei oder drei Arbeitstage in einem Arbeitstag zusammengefasst wurden um anderen Studentischen oder privaten Verpflichtungen gerecht werden zu können.

Die Zeitpunkte der Sprintlänge erwieß sich jedoch im Laufe der Entwicklung als ungünstig, denn an den Wochenenden fiel die Kommunikation der Teammitglieder schwerer als erwartet. So war es auch nur unter der Woche möglich Pairprogramming umzusetzten. Das Pairprogramming wurde jedoch trotz der „Terminprobleme“ so häufig wie möglich umgesetzt. Es wurde darauf geachtet mindestens einmal in der Woche Pairprogramming umgesetzt wurde. Zu Anfangs wurden die Tests etwas vernachlässigt, was auch daran lag das es Probleme bei der Aufsetzung der Testumgebung gab. Hieraus resultierte das die JSF Komponenten nicht getestet wurden.

## Dailies

Die Dailies wurden jeden Tag meist vor Unibeginn abgehalten. In den Dailies wurde besprochen was jeder am Vortag an Aufgaben des Sprints erledigt hat, was ihn dabei behindert hat und was er an diesem Tag erledigen will. Da am Anfang des Projekts nicht immer alle dieser drei Fragen beantwortet wurden und es auch für den Scrummaster schwierig war darauf zu achten wer welche Fragen nicht beantwortet hatte, da wir die Dailies zum teil an verteilten Orten über Skype durchgeführt hatten. Entschloss sich das Team eine Liste einzuführen in der die Fragen welche beantwortet wurden abgehakt wurden. Dies wurde so lange gemacht bis sich das Team an die Fragen gewöhnt hatte und die Liste nicht mehr benötigt wurde.

## Sprint Review

Nach einem Sprint steht der Review, wobei dem Product Owner die vollständig erledigten User Stories vorgestellt wurden, welche er dann testen konnte. Da in diesem Projekt wie schon erwähnt kein unabhäniger Product Owner zur Verfügung stand übernahm dies auch wieder der Scrummaster. Da wurden die User Stories auf dem Laptop des Scrummasters getestet. Innerhalb der ersten zwei Sprints hatte der Product Owner keine Beanstandungen und nahm die fertigen User Stories ab. Ab dem dritten Review kam immer wieder Kritik an einzelnen Use Cases auf, da entweder einzelne Sachen geändert oder hinzugefügt werden sollten. Wenn die genannten Änderungen nicht in der Beschreibung der Use Cases oder als Tasks aufgeführt waren beschloss das Team mit dem Product Owner dies als Change Request in den Product Backlog aufzunehmen. Fall jedoch ein Task nicht nicht oder vergessen wurde umzusetzen galt diese User Story als nicht vollständig und musste somit im nächsten Sprint wieder mit aufgenommen werden.

## Retrospektive

Bei der Retrospektive welche nach jedem Sprint stattfindet werden gute oder schlechte Ereignisse während des letzen Sprint tematisiert. Innerhalb des Projektes wurde dies bis auf eine Ausnahme so umgesetzt. Zu dieser einen Ausnahme kam es da ein Teammitglied zu diesem Zeitpunkt erkrankt war. Während jeder Retrospektive wurde ein Zeitstrahl an das White Board gemalt welcher den Vergangenen Sprint abbildete. Jedes Teammitglied bekam dann Post Its worauf es dann jeweils genau eine positive oder negative Empfindung des Sprints geschrieben wurde. Positive Empfindungen waren nicht rein auf den Sprint bezogen sondern konnten auch ein besonderes Lob des anderen Teammitglieds darstellen. Die Methode das „versteckten“ Schreibens, wie bei der Schätzung der Story Points, hatte zum Vorteil das jeder unabhängig vom anderen seine Meinungen und Empfindungen äußern konnte ohne sofort Kritik zu befürchten. Im Anschluss hefteten die Teammitglieder nacheinander die jeweiligen Post Its an die entsprechende Stelle des Zeitstrahls.

Die negativen Punkte welche am Zeitstrahl eingetragen wurden, wurden analysiert und deren Verbesserungsvorschläge in der Retrospektiv Liste Festgehalten

## Retrospektiv Liste

|  |  |  |
| --- | --- | --- |
| Retro-Nr. | Zeitraum | Änderungen |
| 1 | 17.02-24.02 | - Velocity maximal 10 Stroy Points  - Beantwortung der 3 W-Fragen im Daily  - Größe pro Task maximal 3 Stunden |
| 2 | 24.02-03.03 | - Abschätzung User Stories vor Sprint nochmal  - Taskgröße von 3 beachten  - Daily Überwachung mit Liste  - Tests als Tasks im Board  - Einführung Unit Tests  - Neue Dafinition of Done |
| 3 | 03.03-10.03 | - Taskgröße für Pairprogramming maximal 5 Stunden  - Yodiz: Task mit Comment und Zuweisung bei Review  - Tasks innerhalb User Story schätzen 🡪 besseren Schätzen User Story |
| 4 | 10.03-17.03 | - 1 Daily am Wochenende  - Rückfragen Review spätestens am nächsten Tag  - Velocity erhöhen maximal 15 |
| 5 | 17.03-31.03 | - Dailies ohne Liste  - Velocity in 1. Woche zu niedrig |
| 6 | 31.03-07.04 | - Reviews von Bugs besser Koordinieren |

## Review des gesamten Projektes

Im Anschluss an die letzte Rerospektive wurde eine Beurteilung des gesamten Projektes in gleicher weise wie die Reviews durchgeführt. Dabei wurde festgehalten was bei zukünftigen Projekten mit Scrum beibehalten oder verbessert werden sollte

|  |  |
| --- | --- |
| Positiv | - die Zeiteinteilung  - das Pairprogramming  - die Dailies |
| Verbesserungen | - früher Tests schreiben  - Reviews mehr Beachtung schenken  - schnellere Reaktion auf Kundenänderungen  -aktuell halten des Boards  - mehr Wissenstransfair  - mehr Pairprogramming |