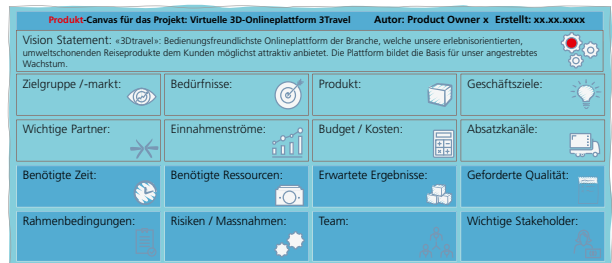


Ein Sprint „Entwicklungsphase“ enthält immer alle Entwicklungsschritte. Das heisst: Analyse & Entwurf, die Entwicklung (z.B. Programmierung) sowie den Modul- und Integrationstest eines Releases. In der Praxis hat sich gezeigt, dass insbesondere bei grossen Entwicklungen vor dem Ausliefern eines Releases in Form eines separaten Sprints auch noch ein Integrations- und Systemtest durchgeführt werden sollte.

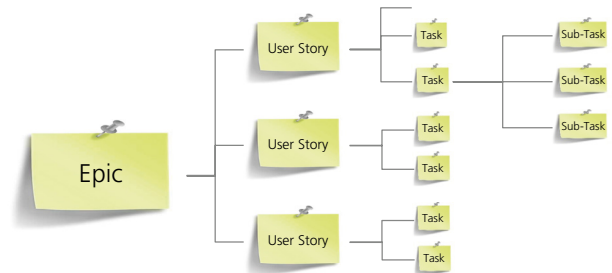
Spezielle Instrumente bzw. Artefakte

Bei der agilen Entwicklung werden spezielle Instrumente (Artefacts) eingesetzt. Diese haben sich in den letzten Jahren von der Urversion zum Teil stark modifiziert und erweitert. In Anlehnung an Schwaber [Sch 2001] und Wirdemann [2017] sind dies:

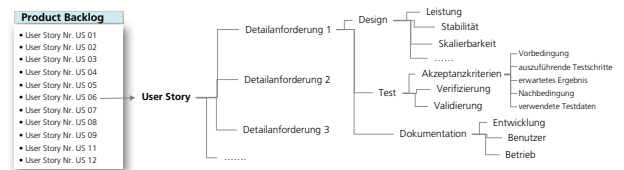
Produktvision: Eine Vision ist ein klares, deutliches, attraktives und emotionalisierendes Bild des angestrebten Produkts – zum Beispiel eine voll digitalisierte Mitgliederverwaltung. Die Produktvision kann in Form eines *Project Canvas* festgehalten werden. Ein Project Canvas ist ein vorstrukturiertes Plakat, mit dessen Hilfe Projektteams die Projektvision sowie die wichtigsten Inhalte im Sinne eines Projektsteckbriefes definieren und dokumentieren.



Epics: Epics sind eine aggregierte, grobgranulare Sicht auf neue Anforderungen eines Produkts (eine grosse oder mehrere, nicht detailliert beschriebene User Stories), z.B. der professionelle Umgang mit Mitgliederausweisen. Epics dienen zur Entwicklung eines Product Backlog. Sie sind zu gross, um geschätzt zu werden, oder auch zu gross, um innerhalb eines Sprints umgesetzt zu werden. Epics entsprechen „Grobanforderungen“ in der konventionellen Vorgehensweise. Eine Epic unterscheidet sich von einer US insbesondere dadurch, dass die Epic nicht geschätzt werden kann (da noch zu unklar) oder dass sie zu gross ist, um diese in einem Sprint umzusetzen. Epics werden gleich beschrieben wie User Stories.



User Story: Eine User Story („Anwendererzählung“) ist eine in Alltagssprache formulierte Produkthanforderung (Kundensicht). User



Stories werden jeweils aus Sicht einer Persona formuliert; d.h. vorgängig müssen die betroffenen/beteiligten Rollen geklärt werden. Sie sind bewusst kurz gehalten und umfassen in der Regel nicht mehr als zwei Sätze (in Anlehnung an Wikipedia). Beispiel: Als Administrator will ich neue, digitalisierte Mitgliederausweise erstellen können. In den User Stories wird die wesentliche Konversation zwischen dem Team und dem Product Owner festgehalten [Tim 2017].

Die Ausgangslage („Hauptsatz“) ist die Basis für die Arbeit in den Iterationen zwischen den Entwicklern und den Fachbenutzern. Die User Stories werden in diesem Rahmen ergänzt um:

- wesentliche Konversation zwischen Entwicklungsteam und Fachspezialisten (Fragen/Antworten)
- Akzeptanzkriterien

User Stories sind per Definition unvollständig und veränderlich! Eine Sammlung verwandter User Stories oder/und Epics wird als Thema bezeichnet. Ein Thema oder auch „Feature“ bezeichnet eine bequeme Möglichkeit anzuzeigen, dass eine Reihe von User Stories etwas gemeinsam haben, z. B. den gleichen Funktionsbereich. Themen dienen somit der Gliederung von User Stories, z. B. einzelne Workflowschritte einer Produktfunktionalität etc. Damit können grosse Schwerpunktbereiche, die allenfalls das ganze Unternehmen betreffen, definiert werden.

User Stories sind unvollständig und veränderlich!

Persona: Um User Stories wirkungsvoll definieren zu können, werden Personas definiert. Die Persona stellt einen Prototyp für eine Gruppe von Nutzern dar, mit konkret ausgeprägten Eigenschaften und einem konkreten Nutzerverhalten. Eine Persona besitzt Merkmale wie fiktiver Vor-/Nachname und Foto (um sie noch besser darzustellen). Hinzu kommen weitere Attribute je nach Relevanz für das System, zum Beispiel Tätigkeit/Arbeitsrolle, Familienstand, Ziele, Wünsche etc.



Eine Persona respektive ihr Verhalten kann sowohl auf funktionale Anforderungen (z.B. „möchte mit Kreditkarte bezahlen“) wie auch auf nicht funktionale Anforderungen (z.B. „Systemfeedback innerhalb bzw. < 1 Sekunde“) Einfluss haben. Die Kunst besteht darin, berechnete von „unberechtigten“ Ansprüchen zu trennen bzw. die richtigen Prioritäten zu setzen und trotzdem alle relevanten Gruppen (zufriedenstellend) abzuholen. Eine Persona ist eine spezialisierte Rolle.

Akzeptanzkriterien:

Akzeptanzkriterien geben vor, wann eine User Story fertig ist und sie für den Kunden einen Mehrwert liefert. Das heisst, mit den Akzeptanzkriterien wird geprüft, ob die Erwartungen der Kunden erfüllt sind. Daher sind sie eine Art von Geschäftsregeln und können als Vorbedingung, Aktion oder Ergebnis formuliert sein. Sie unterstützen im Wesentlichen die Entwickler beim Verständnis, „was zu entwickeln ist“. Akzeptanzkriterien sollten so früh wie möglich (-> Aufwandsschätzung) vorliegen, werden aber auch erst während der Umsetzung finalisiert/vervollständigt.

User Story 015

„Als Administrator will ich neue, digitalisierte Mitgliederausweise erstellen können.“

- [AK01]** Die Suche mit „exakten“ Werten wird korrekt ausgeführt und ein Mitglied angezeigt.
- [AK02]** Jedes Mitglied wird mit Foto, Name, Vorname, Alter, Mitglieder-Nr. und Wohnadresse angezeigt.
- [AK03]** Die ausgewählten Mitglieder werden in die Mitgliederausweisbestellung übernommen.
- [AK04]** Der Administrator kann wählen, weitere Mitglieder zu suchen oder zum Ausdrucken zu gehen.

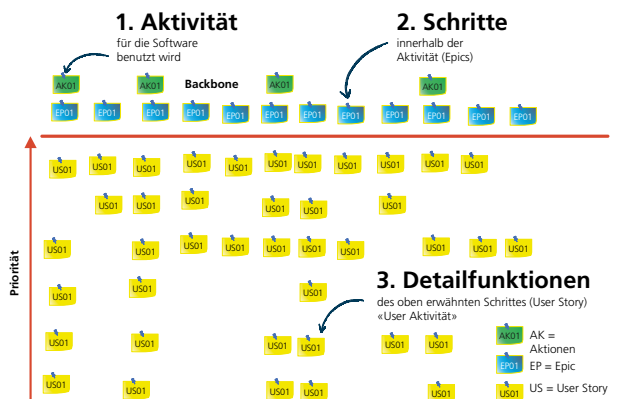
Product Backlog:

Priorisierte Liste sämtlicher Produktanforderungen (z.B. in Form von User Stories) mit zeitlichen Schätzwerten (Estimates) für deren Umsetzung. Die Priorisierung erfolgt primär auf dem Mehrwert, den die User Stories für das Unternehmen bringen. Das Product Backlog ist ein dynamisches Dokument. Im Laufe des Projekts verändern sich Anforderungen: Es kommen neue hinzu, Anforderungen werden verfeinert, Prioritäten ändern sich oder Anforderungen fallen ganz weg.

Nr.	ID	User Story	Story Points	PB-Priorität	Absolute Priorität	Bemerkungen
01	K4	Als Kunde will ich.....	21	• Sehr hoch	1	
02	E2	Als Entscheider will ich....	15	• Hoch	6	
03	E3	Als Entscheider will ich.....	13	• Sehr hoch	2	
04	L4	Als Lieferant will ich.....	34	• Mittel	10	
05	A4	Als Anwender will ich.....	23	• Sehr hoch	4	
06	L1	Als Lieferant will ich.....	8	• Niedrig	11	
07	E8	Als Entscheider will ich....	3	• Sehr niedrig	12	
08	K3	Als Kunde will ich.....	25	• Hoch	7	
09	L2	Als Lieferant will ich.....	56	• Sehr hoch	3	
10	E5	Als Entscheider will ich....	69	• Hoch	8	
11	A7	Als Anwender will ich.....	34	• Sehr hoch	5	
12	L3	Als Lieferant will ich.....	20	• mittel	9	

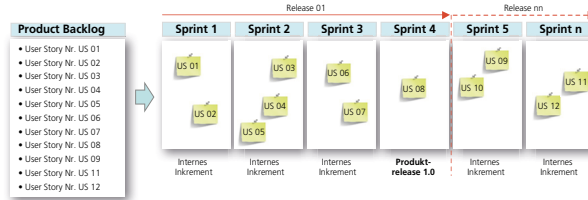
Story Map:

Eine Story Map ist eine Visualisierungsmethode, mit der Benutzeranforderungen berücksichtigt werden können und gleichzeitig ein Überblick über das Backlog ermöglicht wird. Die Verwendung der Methode verbessert das gegenseitige Verständnis des Gesamtprozesses innerhalb des Projektteams [Lat 2017]. Um ein solches Verständnis zu erreichen, hat Jeff Patton das Story Mapping als Best-Practice-Methode vorgeschlagen [Pat 2014]. Für ihn gibt es einen elementaren Grund, diese Technik in der agilen Softwareentwicklung



einzusetzen: „Mit Story Mapping konzentrieren wir uns auf Benutzer und ihre Erfahrungen. Das Ergebnis ist eine bessere Konversation und letztendlich ein besseres Produkt.“

Releaseplan: Er zeigt auf, welche Ziele, welcher Nutzen (Mehrwert) erreicht werden kann und allenfalls mit welchen Epics diese umgesetzt werden sollen.

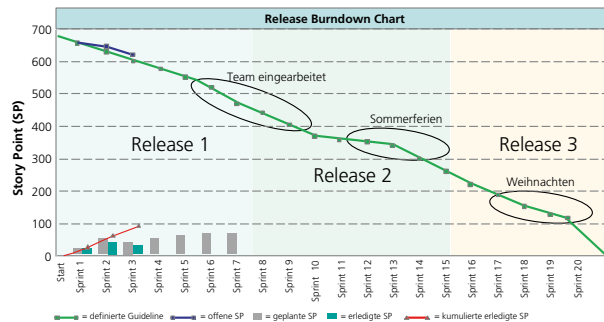


Drei bis fünf Sprints pro Release

Der Product Owner benötigt eine Vorstellung davon, wie viele Sprints es dauern wird, um relevante User Stories umzusetzen. Wichtig dabei ist, dass die geschätzte Grösse, die Priorität und der Mehrwert pro User Story aufgeführt werden. Ein Release kann sich aus drei bis fünf Sprints zusammensetzen. Die Praxis hat gezeigt, dass jeweils der letzte Sprint idealerweise ein Stabilisierungssprint ist [Tim 2017]. Der Releaseplan sagt aus, wie viele Sprints für welchen Release notwendig sind, um die Sprintziele zu erreichen. Daraus abgeleitet aber auch, welche (allenfalls zusätzlichen Ressourcen) für einen Release benötigt werden.

Release Backlog: ist das gleiche Instrument wie das Product Backlog und hat allenfalls bereits weitere Sprints für ein geplantes Releasedatum definiert. Das Release Backlog ist die grobe Einteilung im Sinn der Priorisierung der Epics/User Stories und damit eine Verfeinerung des Product Backlogs. Es ist das Basiskonstrukt für den Releaseplan.

Release Burndown Chart: Darstellung, die sprintaktuell den Arbeitsfortschritt eines gesamten Releases aufzeigt. Zeigt auch den Aufwand für den Restwert auf. Der Chart kann mit verschiedenen Kurven wie einzelne Sprints PLAN/IST etc. ergänzt werden.



Sprint Backlog: Enthält diejenigen Anforderungen (Product Backlog Items „PBI“), welche vom Team aufgrund der Priorisierung des Product Owners als im aktuellen Sprint umsetzbar definiert (committed) worden sind. Das Sprint Backlog ist das Ergebnis der ersten Planungssitzung der Sprintplanung. Es de-

Nr.	ID	User Story	Geplante Tage Aufw.	Story Points	PB-Priorität	Bemerkungen	Committed	Status
01	K4	Als Kunde will ich.....	5	21	• Sehr hoch		Team A, Sprint 1	Geplant
03	E3	Als Entscheider will ich.....	6	13	• Sehr hoch		Team A, Sprint 1	Geplant
09	L2	Als Lieferant will ich.....	12	56	• Sehr hoch		Team A, Sprint 1	Geplant
05	A4	Als Anwender will ich.....	6	23	• Sehr hoch		Team B, Sprint 4	Erledigt
11	A7	Als Anwender will ich.....	8	34	• Sehr hoch		Team B, Sprint 4	Geplant
02	E2	Als Entscheider will ich.....	3	15	• Hoch		Team B, Sprint 4	In Arbeit
08	K3	Als Kunde will ich.....	25	25	• Hoch			Offen
10	E5	Als Entscheider will ich.....	69	69	• Hoch			Offen
12	L3	Als Lieferant will ich.....	6	20	• Mittel			Offen
04	L4	Als Lieferant will ich.....	8	34	• Mittel		Team A, Sprint 2	Geplant
06	L1	Als Lieferant will ich.....	8	8	• Niedrig			Offen
07	E8	Als Entscheider will ich.....	3	3	• Sehr niedrig			Offen

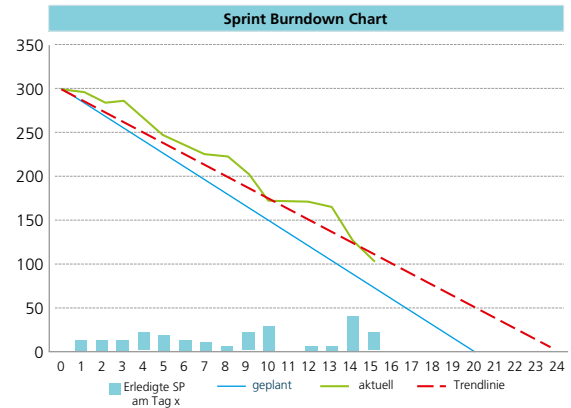
finiert die Menge der Arbeit des nächsten Sprints, die das Team akzeptiert hat und bleibt unverändert während des ganzen Sprints.

Backlog Tasks: vom aktuellen Sprint aus dem Sprint Backlog abgeleitete Tätigkeiten zur Umsetzung der entsprechenden Anforderungen (sie werden bei Scrum als Task bezeichnet). Die Liste präzisiert sich während des Sprints und wird täglich (normalerweise im Daily Scrum Meeting) von allen Team Members gepflegt.

Team: A		Sprint: 1				15. Jan bis 15. Feb.		optional						
Nr.	ID	User Story/Task	Story Points	SOLL-Tage	IST-Tage	Status	Wer	Tag 1	Tag 2	Tag 3	Tag 4	Tag 5	Tag 6	Tag ...
01	K4	Als Kunde will ich ...	21	5	5									
		Task 1	3	1	1		Bruno	0.5	0.5					
		Task 2	7	2	2		Kathrin	1	0.5	0.5				
03	E3	Als Entscheider will ich ...	13	6	6									
		Task 1	6	3	3		Urs	1	1	1				
		Task 2	7	3	3		Peter				1	1	1	
09	LZ	Als Lieferant will ich ...	56	12	12									
		Task 1	6	1	1		Peter				1			
		Task 2	13	2			Kathrin					1	0.5	0.5
		Task 3	9	2			Klaus	1	1					

Sprint Burndown Chart:

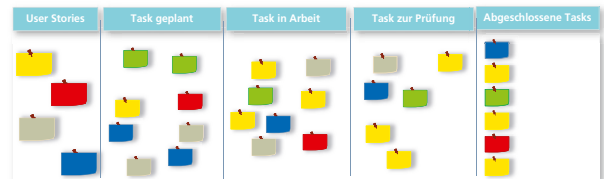
Darstellung, die tagesaktuell den Arbeitsfortschritt des Sprints aufzeigt. Zeigt auch den geschätzte Aufwand für den Restwert auf. Neben den entsprechenden Kurven macht es Sinn, auch den Erledigungsgrad pro Tag grafisch festzuhalten.



Tagesaktuelle Arbeitsfortschritte

Kanban Board/Taskboard:

Einfaches und übersichtliches Planungs-, Status- und Open-Task-Tool. Ideal ist, wenn das Entwicklungsteam das Board stets aktuell hält und im Daily Scrum Meeting bespricht. Dabei werden die User Stories in priorisierter Reihenfolge aufgeführt. Auf einem einzelnen Post-it-Zettel wird ein einzelner Task aufgeführt (Task kann innerhalb eines Tagesaufwandes bearbeitet werden). Die Teammitarbeiter wählen selbstständig aufgrund ihrer Skills einen Zettel respektive eine Task (Status: Task geplant) und notieren darauf ihren Namen und nehmen diesen somit „in Arbeit“.



Impediment Backlog:

eine Auflistung aller Hindernisse, denen das Sprint Team begegnet. Es wird in der Retrospektive und ggf. im

Eigner	Problem	Priorität	Zieldatum	Status	Verantwortlich
Huber	Teamraum zu klein	Prio. 1	14.08.20nn	Erledigt	ALS
Meier	Klimaanlage reparieren	Prio. 2	13.10.20nn	In Arbeit	ZUB
Schmidt	Testing-Umgebung fehlt	Prio. 3	10.09.20nn	Offen	ADM