

Inhaltsverzeichnis:

Inhaltsverzeichnis

DevOps:	2
Bug-/Issuetracking.....	3
Bugtracker:	3
Issue tracking Systems:.....	3
Projektmanagement Theorie	3
Projektmanager:.....	3
Vorgehensmodelle:	3
Arten:.....	3
Grundlegende Prozessmodelle:	3
Iterative Prozessmodelle:	4
Agile Prozessmodelle:.....	4
V-Modell:	4
Scrum.....	5
User Stories:	5
Epic:	5
Task:.....	6
Sprint:	6
Quellen:	10

Projektmanagement-Tools

DevOps:

Der Begriff stellt sich zusammen aus Development (Dev), welcher für die Softwareentwickler steht, und Operations (Ops), stellvertretend für den IT-Betrieb (Prozesse und Dienste bezeichnet, und die Bereitstellung und Wartung von Hardware und Software für interne wie auch für externe Kunden.).

Um flexibel und schnell auf neue Geschäftsanforderungen reagieren zu können, müssen Entwicklung und Betrieb besser zusammenarbeiten und ihre Prozesse mehr aufeinander abstimmen.

Der DevOps-Ansatz hilft dabei, agile Methoden („Task Boards“, „Daily-Standup-Meetings“ und „User Stories“) auf den IT-Betrieb zu übertragen und Standardmodelle für Software-Entwicklung (V-Modell, Wasserfallmodell, Scrum) und IT-Betrieb miteinander zu verbinden.

Dadurch werden kürzere Release-Zyklen möglich und das Risiko von ungetesteten Elementen wird minimiert, da dieselben Verfahren im gesamten Software-Prozess identisch und ohne Brüche eingesetzt werden.

DevOps in der Praxis:

Die Verwendung des DevOps-Ansatzes bedeutet für die Entwickler eine vermehrte Beschäftigung mit dem Installieren von virtuellen Maschinen, mit Aspekten der IT-Sicherheit („DevSec“) oder mit der Planung und Durchführung von Auslieferungen.

Continuous Integration: Auch Continuous Integration ist ein wichtiger Punkt bei DevOps, Continuous Integration sorgt dafür dass neue Programmteile sofort getestet werden und zusammengeführt werden können, statt dies beispielsweise nur einmal Nachts zu tun.

Früher war es so dass vorgenommene Änderungen und Erweiterungen nur einmal täglich, in einer Kompilierung auf dem „Build Server“ zusammengeführt wurden. Das hat sehr viele Risiken gehabt, denn wenn neu geschaffener Code nicht stetig integriert wird, wird fehlerhafter Code zu spät erkannt.

Continuous Delivery: Weiters möchte ich noch einen Begriff erwähnen der auch immer bei DevOps vorkommt. Der Begriff der Continuous Delivery beschreibt eine Sammlung von Techniken, Prozessen und Werkzeugen, mit deren Hilfe kurze Entwicklungszyklen und die schnelle Auslieferung von Software-Updates oder produktiven Endsystemen ermöglicht werden.

Früher war es so zuerst wurde programmiert, dann getestet und nach Prüfungen kam sie beim Kunden an.

Mit Continuous Delivery wird das System verändert, und sorgt durch die kontinuierliche Überprüfung aller Änderungen und durch die automatisierten Tests für eine jederzeit lauffähige Software. Das bedeutet der Kunde bekommt zu jedem Entwicklungszeitpunkt eine lauffähige Version der Software.

Bug-/Issuetracking

Bugtracker:

Ein Bugtracker ist ein System in der Softwareentwicklung zur Dokumentation des Lebenszykluses von Fehlern. Ein „Bug“ ist ein Fehler und der Begriff „tracking“ bzw. „to track“ heißt ins Deutsche übersetzt „verfolgen“. Daraus ergibt sich, dass Bugtracking den strukturierten Ablauf von der Erfassung bis zur Beseitigung von Fehlern beschreibt.

Issue tracking Systems:

Ist eine Art von Software, um Empfang, Bestätigung, Klassifizierung und Bearbeitung von Kundenanfragen (Tickets bzw. Fälle) zu handhaben. Als Anfragen werden eingehende Kundenanrufe, E-Mails, Faxe und Ähnliches betrachtet.

Gemeinsam haben all diese Systeme die Möglichkeiten der Zuweisung eines Tickets an eine Funktionsstelle oder an eine Person innerhalb einer Funktionsstelle zur weiteren Bearbeitung bis zur Lösung. Mit dem Ticketsystem soll sichergestellt werden, dass keine Nachricht verloren geht und jederzeit ein Gesamtüberblick über die zu bearbeitenden Vorgänge möglich ist.

Projektmanagement Theorie

Projektmanager:

Ist die zentrale Person im Projekt und verantwortlich für

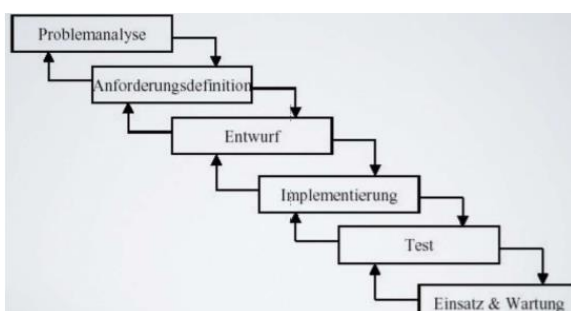
- für die Vorgabe von Zielen
- für die Planung
- für die Reibungslose Abwicklung
- durch Kontrolle und Steuerung

Vorgehensmodelle:

Arten:

Grundlegende Prozessmodelle:

Beispiel: Wasserfallmodell: Beim Wasserfallmodell wird die Softwareentwicklung in aufeinander folgende Projektphasen unterteilt. Dabei gehen die Phasenergebnisse wie bei einem Wasserfall immer als bindende Vorgaben für die nächsttiefere Phase ein. Die Phasen haben dabei vordefinierte Start- und Endpunkte mit eindeutig definierten Ergebnissen. Meistens beinhalten sie auch eine Beschreibung der einzelnen Aktivitäten die zu erledigen sind, um die Ergebnisse zu erreichen.



Iterative Prozessmodelle:

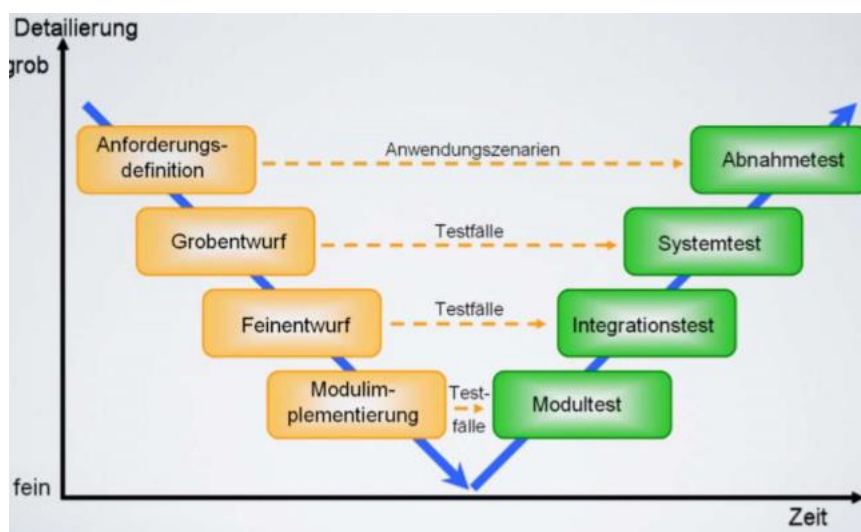
Bei iterativen Prozessmodellen wird die Softwareentwicklung in kleine Schritte unterteilt, die das Gesamtergebnis immer wieder um einen kleinen Schritt verbessert. Der Endzustand steht in der Regel nicht fest. Beispiel: RUP, OEP

Agile Prozessmodelle:

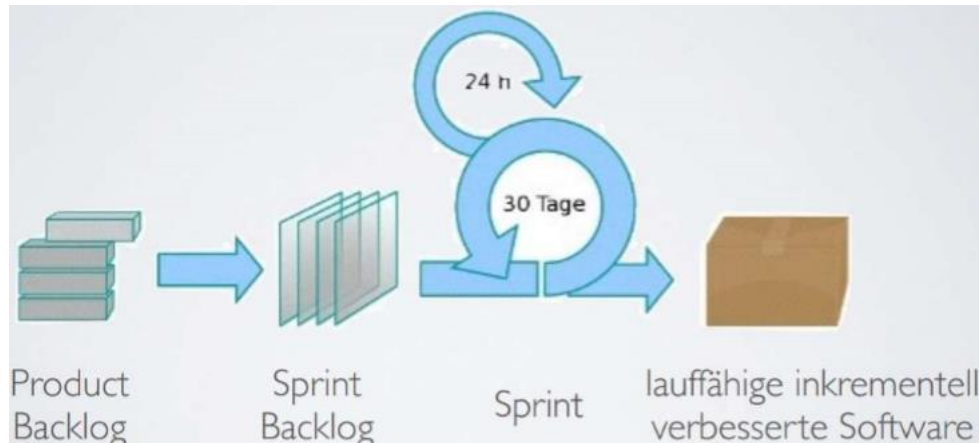
Die Agile Softwareentwicklung erhöht die Transparenz und Flexibilität, was somit zu einem schnelleren Einsatz der entwickelten Systeme führt, was somit das Risiko des Entwicklungsprozesses minimiert. Das Ziel agiler Softwareentwicklung ist es, den Entwicklungsprozess flexibler und schlanker zu machen, als das bei den klassischen Vorgehensmodellen der Fall ist. Die agile Softwareentwicklung ist eine Gegenbewegung zu den oft als schwergewichtig und bürokratisch angesehenen traditionellen Softwareentwicklungsprozessen. Es wird versucht auf technische und soziale Probleme einzugehen und den Fokus auf das Ziel zu richten. Dabei stehen Individuen und Interaktionen vor Prozessen und Werkzeugen, funktionierende Software vor umfangreicher Dokumentation, die Zusammenarbeit mit dem Kunden vor der Verhandlung von Verträgen und das Reagieren auf Veränderungen vor dem Befolgen eines Planes. Beispiel: XP, Scrum

V-Modell:

Das V-Modell ist ein Vorgehensmodell in der Softwareentwicklung, bei dem der Softwareentwicklungsprozess in Phasen organisiert wird. Neben den Entwicklungsphasen definiert das V-Modell auch das Vorgehen zur Qualitätssicherung (also die Tests) phasenweise. Auf der linken Seite wird mit einer funktionalen/fachlichen Spezifikation (Gesamtheit von Vorgaben, nach denen etwas produziert werden soll) begonnen, die immer tiefer detailliert zu einer technischen Spezifikation und Implementierungsgrundlage ausgebaut wird. In der Spitze erfolgt die Implementierung, die anschließend auf der rechten Seite gegen die entsprechenden Spezifikationen der linken Seite getestet wird.



Scrum

*User Stories:*

Eine User Story ist eine in Alltagssprache formulierte Software-Anforderung. Sie ist bewusst kurz gehalten und umfasst nicht mehr als zwei Sätze.

Eigenschaften einer guten User-Story:

- User Stories sollen unabhängig voneinander sein
- User Stories sollen verhandelbar sein
- User Stories sollen einen Wert für dein Kunden haben
- User Stories sollen schätzbar sein
- User Stories sollen klein sein
- User Stories sollen testbar sein

Erstellung einer User Story:

- Zuerst wird die User Story formuliert.
- Dann wird ein Epic (siehe unten) erstellt/ausgewählt.
- Nun werden zusätzliche Notizen gemacht, die die User Story in einzelne Aktivitäten unterteilt.
- Anschließend wird die Dauer der User-Story mit Punkten bewertet.
- Nun werden Akzeptanzkriterien gebildet. Akzeptanzkriterien stecken den Rahmen der User Story ab und sind vor dem Sprint zu erstellen

"Als <Rolle> möchte ich <Ziel/Wunsch>, um <Nutzen>" z.B.: Entwicklung einer Software, um freie Arbeitsstellen informiert zu werden: „Als Benutzer möchte ich einen guten Überblick über Jobs, die für mich relevant sind, haben, um zu wissen, an welche Firmen ich eine Bewerbung schicken werde und an welche nicht.“

Epic:

Der Begriff "Epic" steht einfach für eine große User Story. Es gibt keine exakte Grenze, wann User Stories zu Epics werden. Sie sind einfach nur "große User Stories" und das Prinzip kann gut mit Filmen verglichen werden. Wenn ich sage, dass ein bestimmter Film ein Actionfilm ist, sagt das etwas über den Film aus. Mit großer Wahrscheinlichkeit gibt es Verfolgungsjagden, Schießereien usw. Und das weiß man, obwohl es keine feste Definition davon gibt, dass in einem Actionfilm beispielsweise mindestens drei Verfolgungsjagden, 45 Schüsse etc. vorkommen müssen.

“Epic” ist also einfach nur der Name für eine große User Story. Manchmal kann er aber auch zusätzliche Informationen vermitteln. Wenn Sie mich z. B. fragen, ob ich gestern die User Stories zu den monatlichen Verkaufsberichten geschrieben habe, sage ich “Ja, aber die meisten davon sind Epics.” Dann wissen Sie automatisch, dass ich sie zwar geschrieben habe, aber sie noch nicht in kleinere Stories aufgeteilt habe, die direkt realisiert werden können.

Task:

Ein Task wird normalerweise nur von einer Person bearbeitet.

Unterschied User Story und Task: User Stories beinhalten verschiedene Arten von Arbeit (z. B. Programmieren, Testen, Database Design, User Interface Design etc.), während sich Tasks nur auf eine bestimmte Art von Arbeiten beschränken.

Rollen:

In Scrum gibt es in der Regel einen Product owner der fachlich verantwortlich ist, einen Scrum Master der organisatorisch verantwortlich ist, Team members die die Entwicklung durchführen, sowie Users und Stakeholders.



Sprint:

Bei einem Sprint wird ein Abschnitt der Software in meistens 30 Tagen umgesetzt. Nach jedem Sprint gibt es eine lauffähige Version der Software. Für einen Sprint wird zuvor immer ein Sprint Backlog gebildet (siehe unten). Jeden Tag gibt es ein Treffen des Scrum Teams, um unter anderem den Status des Sprints zu klären.

Schritte:

1. Anforderungen
 - a. Zuerst werden Anforderungen in Form von User-Stories erstellt, die dann den Product Backlog bilden

2. Backlog priorisieren

- a. In diesem Schritt wird festgelegt welche User-Stories unbedingt umgesetzt werden müssen und welche nur den Status „Nice-To-Have“ tragen.

Must Have	
Als Verkäufer will ich Rechnungen schreiben	!!!
Als Verkäufer will ich neue Kunden erfassen	!!!
Als Lagerverantwortlicher will ich den Wareneingang verbuchen	!!!
Als Lagerverantwortlicher will ich den Warenausgang verbuchen	!!!
Should Have	
Als Verkäufer will ich den derzeitigen Lagerstand pro Produkt abrufen	!
Als Verkäufer will ich den maximalen Rabatt pro Produkt kennen	!
Als Systemadministrator will ich die Benutzer verwalten	!
Als Geschäftsführer will ich die Übersicht über sämtliche Verkäufe	!
Als GF will ich den monatlichen Umsatz pro Produktgruppe	!

3. Schätzen

- a. Nun wird festgelegt mit welchem Aufwand die Umsetzung der User-Stories verbunden ist.

User Story	Größe
Must Have	
Als Verkäufer will ich Rechnungen schreiben	5
Als Verkäufer will ich neue Kunden erfassen	3
Als Lagerverantwortlicher will ich den Wareneingang verbuchen	3
Als Lagerverantwortlicher will ich den Warenausgang verbuchen	3
Should Have	
Als Verkäufer will ich den derzeitigen Lagerstand pro Produkt abrufen	1
Als Verkäufer will ich den maximalen Rabatt pro Produkt kennen	1
Als Systemadministrator will ich die Benutzer verwalten	3
Als Geschäftsführer will ich die Übersicht über sämtliche Verkäufe	1
Als GF will ich den monatlichen Umsatz pro Produktgruppe	1

4. Design der User Stories

- a. Nun wird festgelegt mit welchen einzelnen Schritten eine User-Storie verbunden ist:



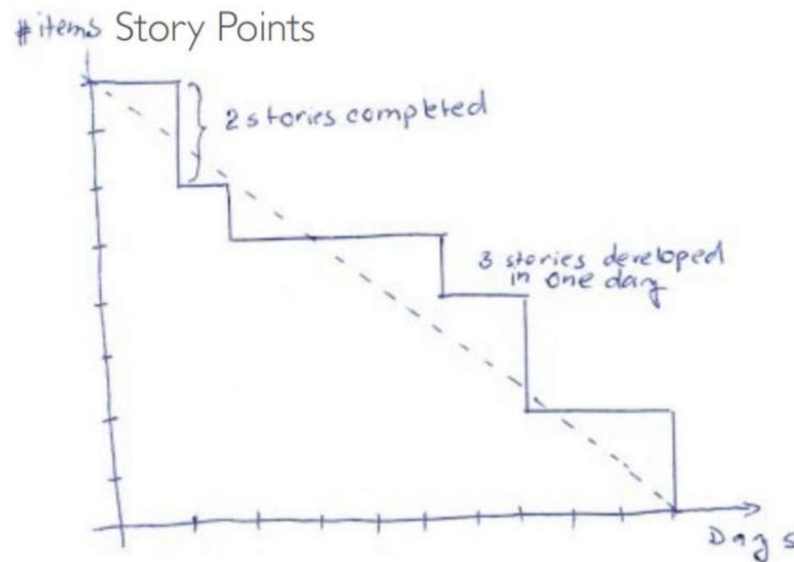
5. Sprint Backlog

- a. Anschließend wird der Sprint Backlog gebildet. Im Sprint-Backlog befinden sich alle User-Stories die im nächsten Sprint durchgeführt werden sollen. Die benötigten Aktivitäten für die User-Stories können nun unterteilt werden in „offen“, „in Arbeit“ und „fertig“.



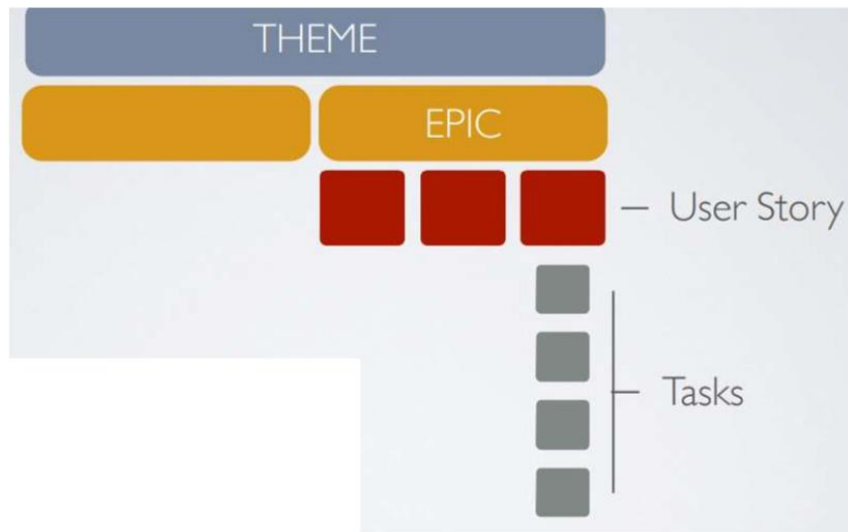
6. Sprint Burn-Down-Chart

- a. Ein Burn-Down-Chart stellt grafisch den verbleibenden Aufwand eines Sprints, in Relation zur verbleibenden Zeit dar.



7. Detaillierungsgrad

- a. Die Software kann grundsätzlich in Epics unterteilt werden. Epics sind Anforderungen an die Software auf hohem Abstraktniveau. Epics können in mehrere User-Stories unterteilt werden und diese in mehrere Tasks.



Quellen:

<https://t2informatik.de/wissen-kompakt/bugtracker/>
<https://t3n.de/news/was-bedeutet-eigentlich-devops-723440/>
<https://community.atlassian.com/t5/Jira-questions/How-to-add-task-to-a-story/qaq-p/292666>
<https://www.computerwoche.de/a/wie-devops-die-it-beschleunigen,3071433>
<https://www.dev-insider.de/was-ist-continuous-integration-a-690914/>
<https://de.wikipedia.org/wiki/DevOps>
http://openbook.rheinwerk-verlag.de/it_handbuch/kap_05_os_konzepte_001.html
https://www.projektmagazin.de/artikel/issue-tracking-im-projektmanagement-teil-1_6797
<https://trello.com/>
<https://www.atlassian.com/de/software/jira>
<https://stock.adobe.com/>
<https://medium.com/faun/most-popular-ci-cd-pipelines-and-tools-ccfdce429867>
<https://github.com/integrations/jira>
<https://jasonstcyr.com/2013/11/11/how-to-set-up-trello-board-for-scrum/>
Folien(Vorgehensmodell) von Prof. Stütz
<https://www.scrum-academy.de/product-owner/wissen/user-stories-epics-themes/>