

Modul 426

Software mit agilen Methoden entwickeln

Programm 23.8.2019

Unterrichtsziel:

- Was ist eine Vision?
- Klassische Softwareentwicklung
- Was heisst agil?
- Vorteile von agiler Softwareentwicklung
- Das agile Manifest
- Agile Softwareentwicklung
- Was ist eine Userstory?
- Andere agile Methoden
- Der Scrum Prozess

Modul 426:

Software mit agilen Methoden entwickeln



Modul 426:

Software mit agilen Methoden entwickeln

Ziel für Semester:

- ***Scrum kennen***
- Andere agile Methoden
- Viele praktische Arbeiten
- Einige Youtube-Videos
- Einige Artikel im Internet

Klassische Softwareentwicklung

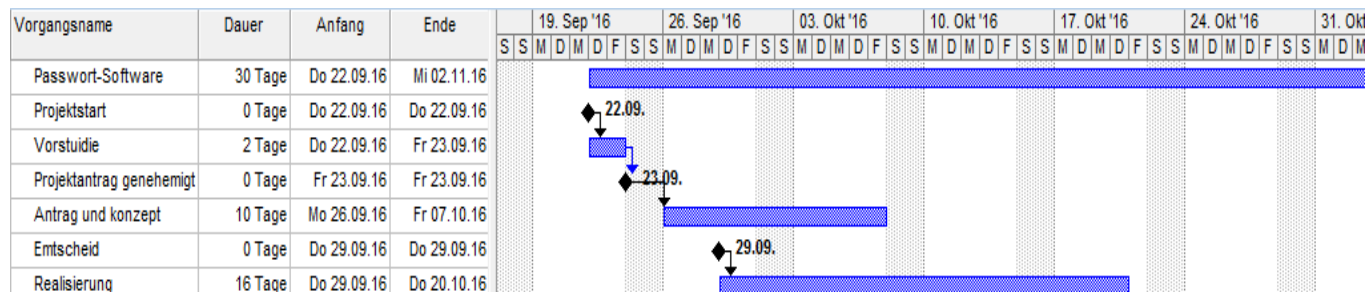


Fachliches Soll-Konzept

- **Bereinigung der Schwachstellen**
 - **Ziele**
 - Alternativvorschläge zu Aufgabenstellung der IST-Analyse
 - Bewertung Nutzen
 - Bewertung Wirtschaftlichkeit
 - Bewertung Vor- und Nachteile
 - Untersuchung der Realisierbarkeit
 - Entscheidung
 - Maßnahmen zur Bereinigung der Schwachstellen
 - Beeinflussung Aufbauorganisation und Ablauforganisation bei Änderung der Organisation mit
 - Entwurf neuer DV-Verfahren
 - Anpassung vorhandener DV-Verfahren
 - Unterteilung in DV und manuelle Maßnahmen
 - Prüfung auf Umsetzbarkeit im Unternehmen
 - Durchsetzbarkeit
 - Unterstützung Unternehmensleitung
 - Unternehmensdeckung der Maßnahmen

Wirtschaftsinformatik II
- Musterklausur
SS 2016

33



Agil, Agilität

Was bedeutet 'agil' oder 'Agilität'?



Gemäss Duden heisst 'agil':
beweglich, flink, gewandt,
Lebendigkeit

Softwareentwicklung

- Agile Softwareentwicklung steht im Widerspruch zu Konzept und Projektplan in herkömmlichen (klassischen) Softwareprojekten:

Klassische Softwareentwicklung	Agile Softwareentwicklung (Scrum)
Konzept	User Stories
Lastenheft	
Pflichtenheft	
Projektplan	

Agile Softwareentwicklung

Der Nutzen von agiler Softwareentwicklung.

Vorteile:

- Kürzere Entwicklungszeit (man redet von einer Entwicklungszeit die bis Faktor 5-10 schneller ist).
- Der Kunden bekommt exakt das, was er sich wünscht.

Nachteil:

- Die Kostenschätzung ist etwas schwieriger (sind in der Regel keine Festpreisprojekte)

Agiles Manifest (bed. Grundsatzerklärung)

die vier Leitsätze

1. Individuen und Interaktionen

sind wichtiger als Prozesse und Werkzeuge

2. Funktionierende Software

ist wichtiger als umfassende Dokumentation

3. Zusammenarbeit mit dem Auftraggeber

ist wichtiger als Vertragsverhandlungen

4. Reagieren auf Veränderungen

ist wichtiger als das befolgen eines Plans

Die vier Leitsätze

Auftrag:

- Lesen der Erklärungen zu den vier Leitsätzen

<http://doktor-scrum.de/agiles-manifest-4leitsaetze/>

Agile Softwareentwicklung

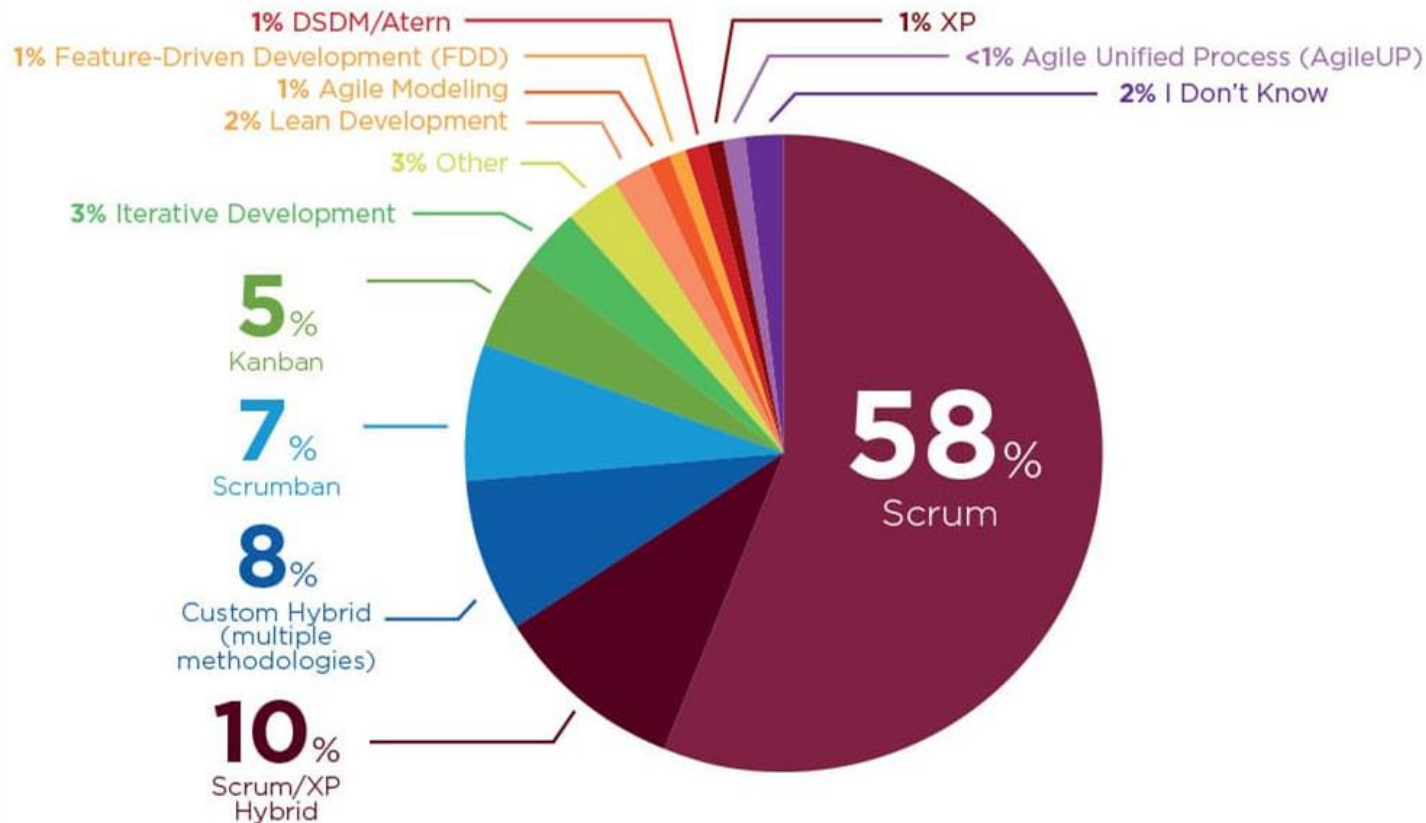
extreme Programming

Kanban

SCRUM

Feature Driven Development

Verbreitung verschiedener agilen Methoden



SOURCE: VERSIONONE 10TH ANNUAL STATE OF AGILE™ REPORT
© 2016 VERSIONONE INC. ALL RIGHTS RESERVED.

Wir haben eine Vision



Um die Vision umzusetzen, erstellen wir User Stories!

User Story

- Eine **User-Story** ("Anwendererzählung") ist eine in Alltagssprache formulierte Software-Anforderung. Sie ist bewusst kurz gehalten und umfasst in der Regel nicht mehr als zwei oder drei Sätze.

Ein Beispiel:

Als Vertragshändler möchte ich alle Gebrauchtwagen auflisten können, um diese meinen Kunden anzubieten:

- *Die Liste zeigt alle noch nicht verkauften Fahrzeuge*
- *Die Liste ist nach Kaufpreis sortiert*
- *Wenn keine Fahrzeuge vorhanden sind, erscheint ein Hinweistext.*

User Story

- Eine **User-Story** ("Anwendererzählung") ist eine in Alltagssprache formulierte Software-Anforderung. Sie ist bewusst kurz gehalten und umfasst in der Regel nicht mehr als zwei oder drei Sätze.

Ein Beispiel:

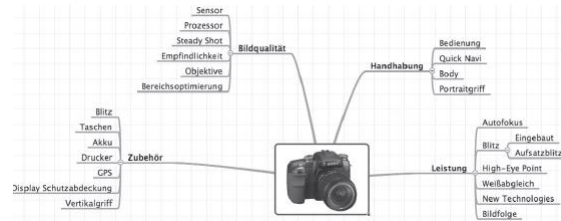
Als Vertragshändler möchte ich eine Liste der noch nicht verkauften Fahrzeuge auflisten können, um diese anzubieten.

Keine technischen Details

- Die Liste ist nach Kaufpreis sortiert
- Wenn keine Fahrzeuge vorhanden sind, erscheint ein Hinweistext.

Hilfsmittel für User Stories

- Mind Maps:



- Spread Sheets:

File Edit View Format Insert Tools Form Help					
A		B		C	
1	Theme	Story	UAT	Busine Value	Size
2	Ausbruch	Als Gefanger möchte ich eine Tool, so das ich ein Loch in die Wand machen kann, dam ich ausbrechen kann	Wenn ich das Tool benutze, entsteht ein Loch ein der Wand	100	13
3	Kommunikation	Als Gefangener möchte ich ein Tool, mit dem ich meine Mitgefangenen über den Ausbruch informieren kann	Wenn ich das Tool benutze, könne mich die anderen Gefangenen hören	100	20

- Story Cards: C6 Format

SCRUM (bedeutet Gedränge)



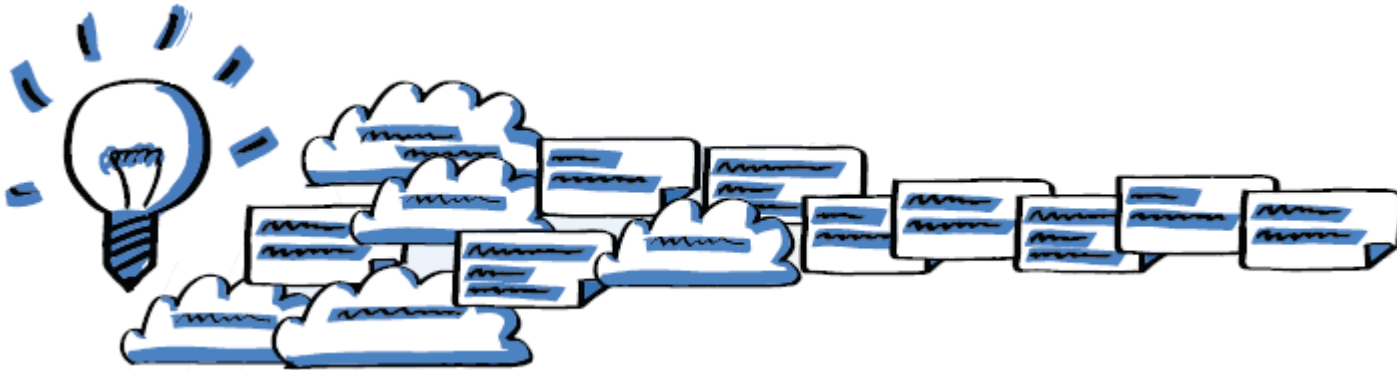
SCRUM



Auf Scrum übertragen, steht dieses Bild für den Zusammenhalt, den ein Scrum-Team entwickeln kann, und für das Einhalten weniger Regeln. Rugby ist ein sehr diszipliniertes Spiel, es geht zwar rauh zu, aber die Regeln werden rigide befolgt. Wir werden sehen, dass Scrum nur wenige Regeln hat, dass diese jedoch sehr genau umgesetzt werden.

Wie geht Scrum

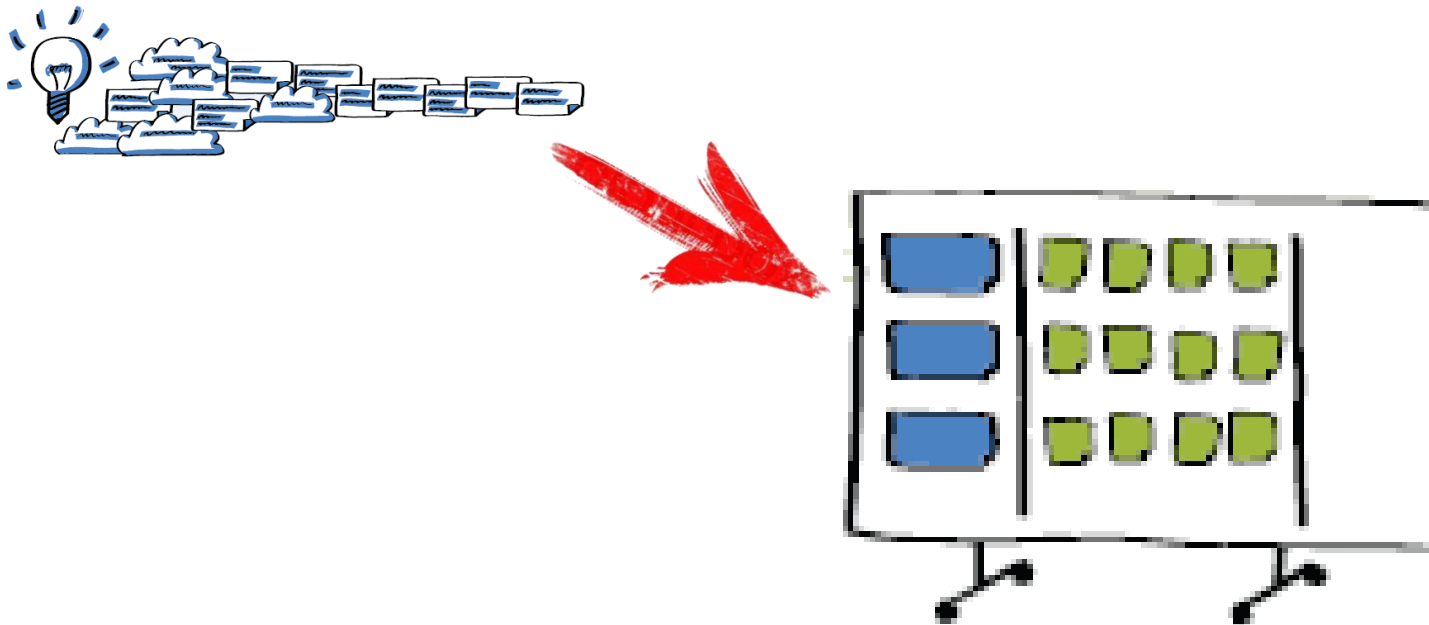
- Alle Funktionalitäten werden mittels User Stories aufgeführt!



- Dieses 'Artefakt' heisst **Product Backlog**

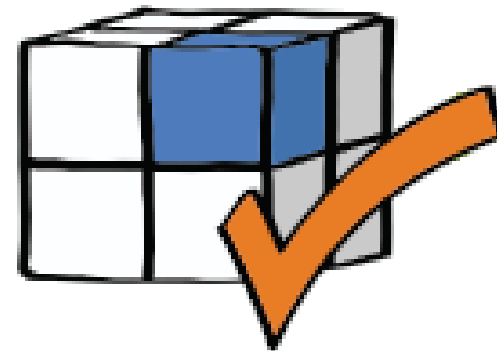
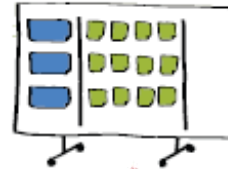
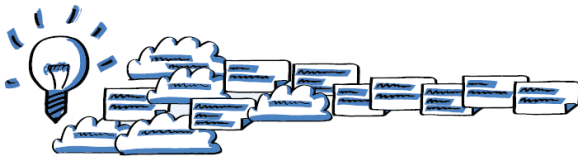
Artefakt Sprint Backlog

- Es wird entschieden, welche Userstories zuerst realisiert werden



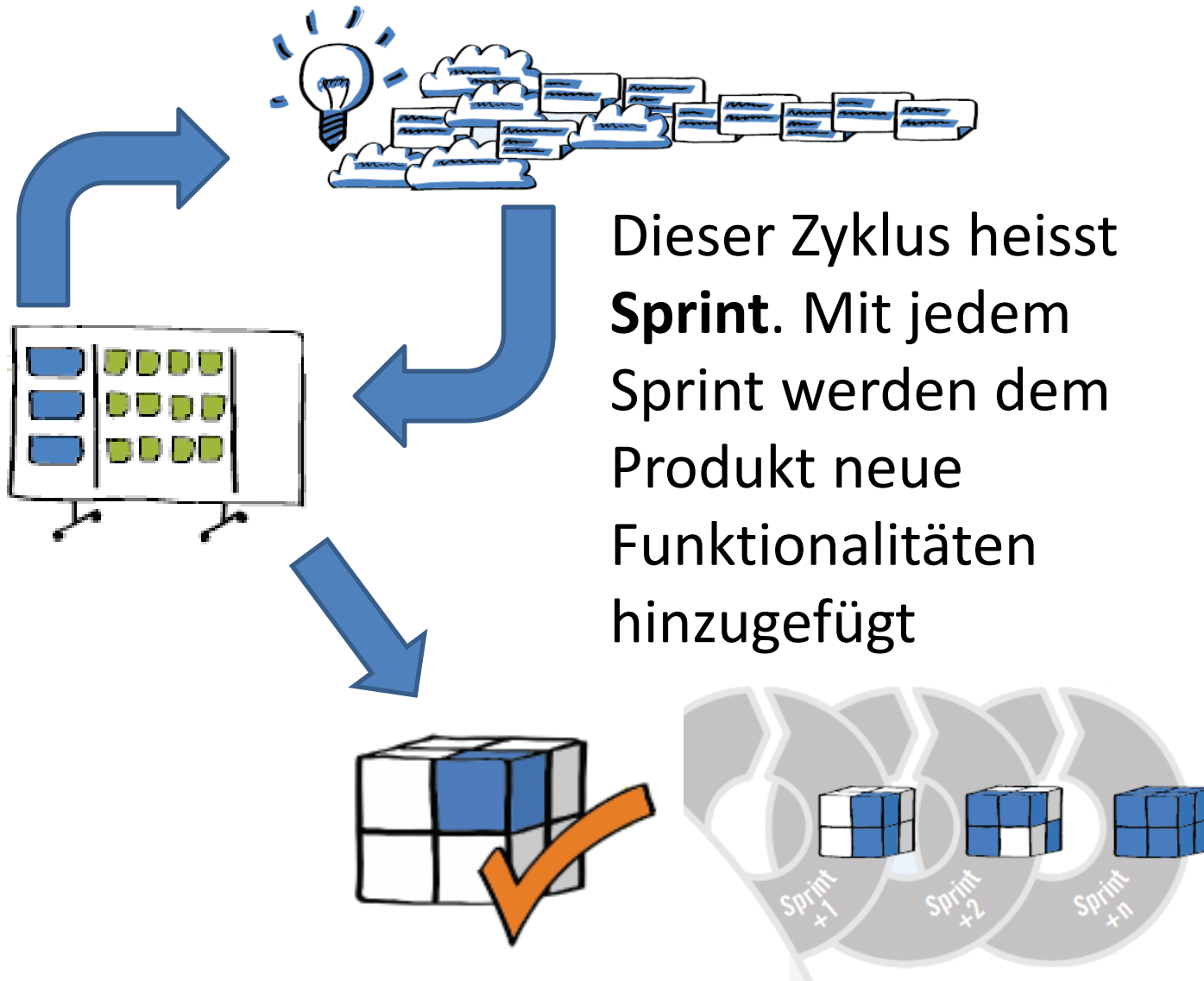
- Dieses 'Artefakt' heisst **Sprint Backlog**

Das Product Increment

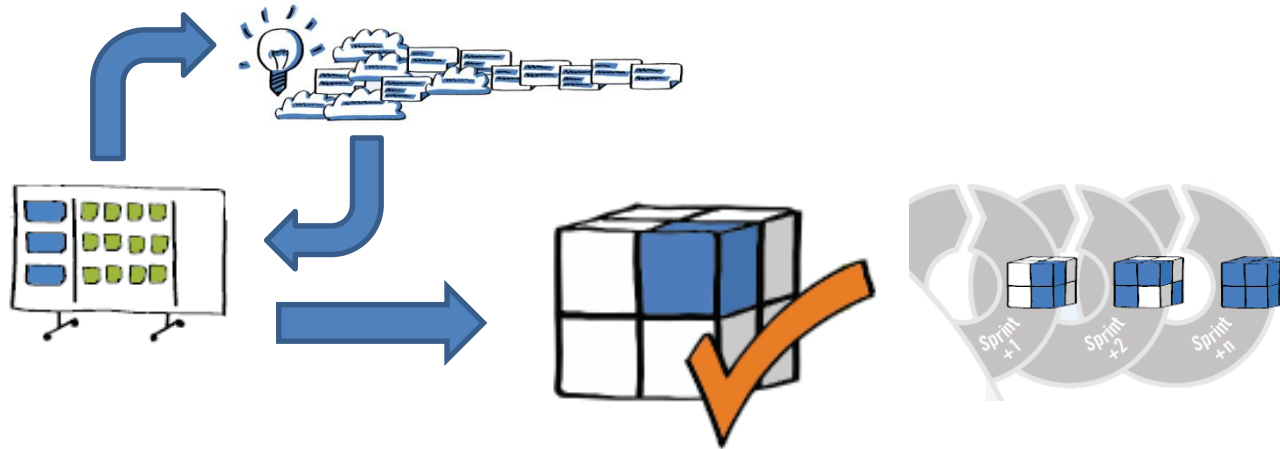


- Aus den gewählten Items des **Sprint Backlogs** wird ein Product increment entwickelt! Das ist das Artefakt **Product Increment**

Der Zyklus



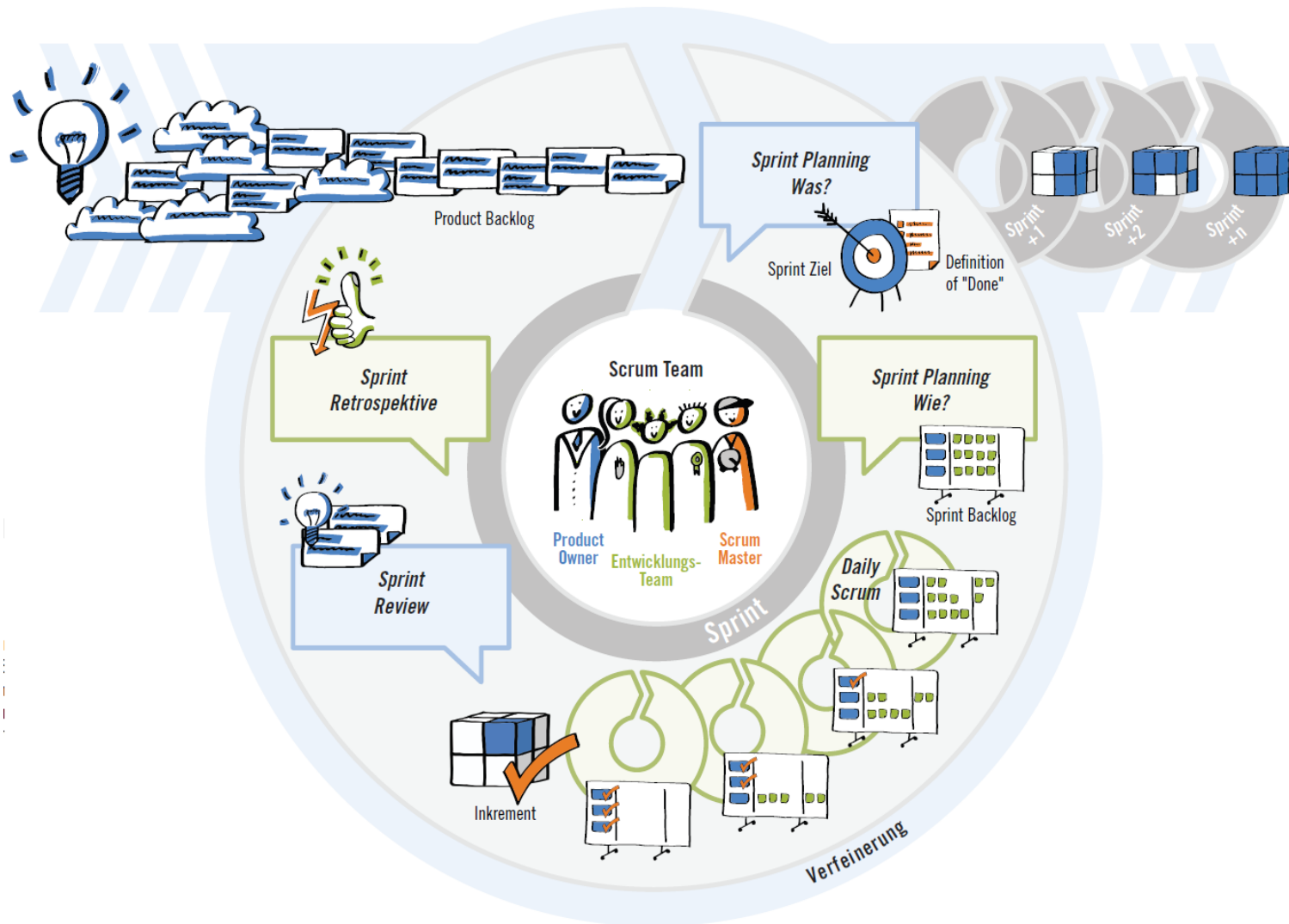
Der Zyklus



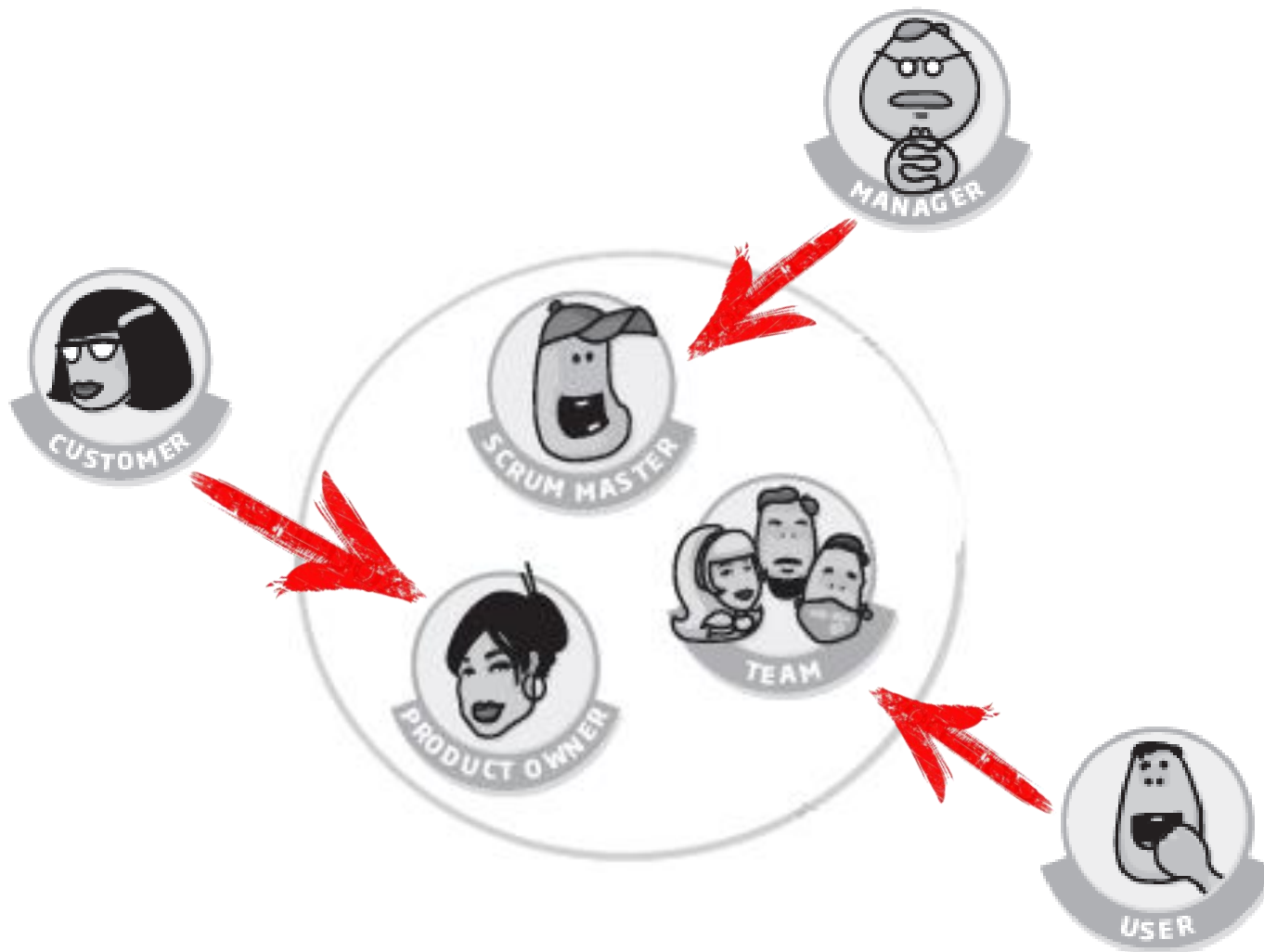
Das ist natürlich etwas vereinfacht dargestellt, aber grundsätzlich wird mit jedem neuen Sprint das Produkt 'fertiger'.

Jedes Product Increment ist mit den neuen Funktioniltäten voll funktionsfähig und weist keine Fehler auf!

Der Scrum-Prozess



Rollen in Scrum



Die 3 Rollen in Scrum

- **Der Product Owner**

- Trägt die Budget-Verantwortung und ist die Schnittstelle zum Auftraggeber. Er erstellt das Product Backlog (User Stories) und priorisiert die Backlog Items.


- **Das Entwickler Team**

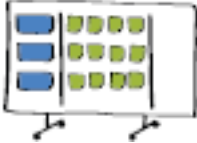
- Das Entwickler Team entscheidet wie viele Backlog Items in das Sprint Backlog kommen. Es entwickelt die Software nach den Regeln von Scrum. Das Entwickler-Team organisiert sich selbst.


- **Der Scrum Master**

- Räumt Hindernisse aus dem Weg und kontrolliert, ob die Spielregeln eingehalten werden.

Die 3 Artefakte

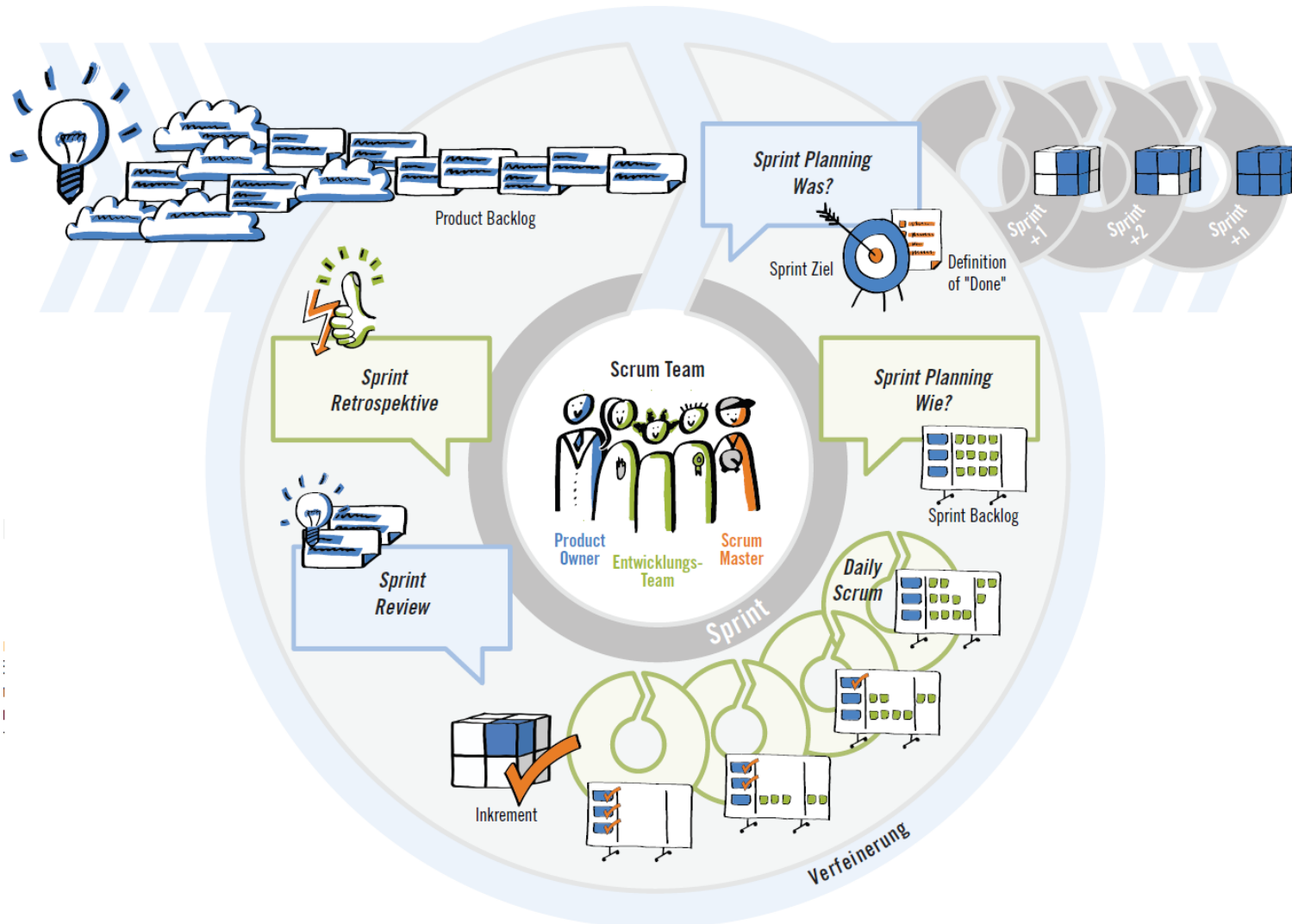
- **Product Backlog** 

Das Product Backlog (User Stories) ist eine geordnete Liste, von allem, was im Produkt enthalten sein soll.
- **Sprint Backlog** 

Das Sprint Backlog ist die Menge der für den Sprint vorgesehenen Backlog Einträgen.
- **Inkrement** 

Das Inkrement ist das Ergebnis aus allen in einem Sprint fertiggestellten Backlog-Einträgen und dem Resultat der Inkremente aller früheren Sprints.

Scrum-Prozess



Die 5 Zeremonien

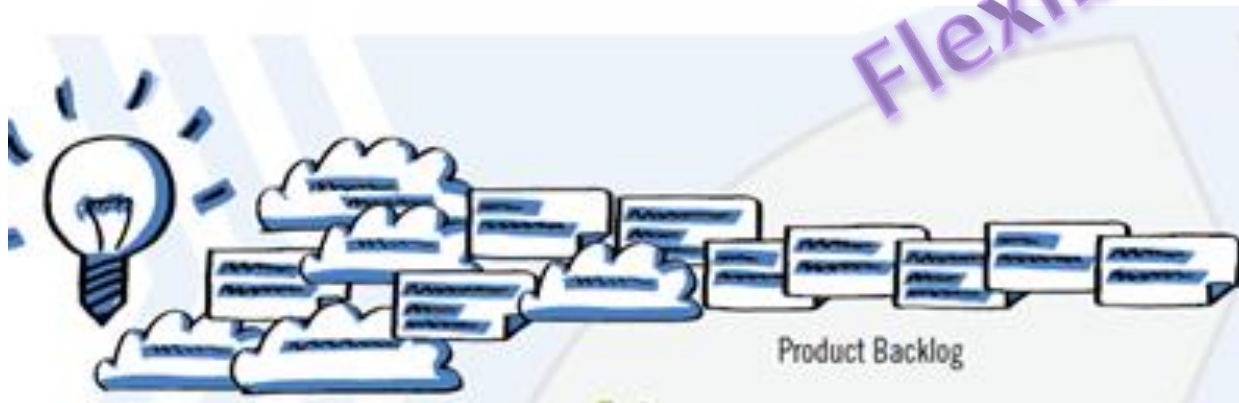
- **Sprint:** Fixe Zeiteinheit (*Time-Box*), in der Regel 2 – 4 Wochen.
- **Sprint Planning:** definiert das Sprint Ziel. Ist auf jeden Fall *ein lauffähiges Produkt!*
- **Daily Scrum:** In der Regel 15 Minuten pro Tag. Aktualisierung des Sprint Backlogs.
- **Sprint Review:** Das Team zeigt dem Auftraggeber das Produkt, bekommt eventuell weitere Anforderungen welche ins Product Backlog einfließen.
- **Sprint Retrospektive:** Die Leistung eines Teams kontinuierlich verbessern, und wie sie abläuft.

Sprint

- Nach jedem Sprint (*!Time-Box!*) präsentiert der Scrum Product Owner dem Auftraggeber das fertig gestellte Resultat. Bezeichnend ist, dass nach jedem Sprint ein lauffähiges Produkt präsentiert wird.
- Dem Auftraggeber bietet sich die Möglichkeit, neue Anforderungen an das Produkt zu stellen:
 - > Dies ergibt neue Product Backlog Einträge (User Stories)

Product Backlog

- Kann jederzeit ergänzt werden:



Google Docs boris.gloger@gmail.com | [New features](#) | [Docs Home](#) | [Help](#) | [Sign Out](#)

Product Backlog - Jail Scrum Share Autosaved on 5/29 AM


File Edit View Format Insert Tools Form Help

123 10pt B A

	A	B	C	D	E
1	Theme	Story	UAT	Busine Value	Size
2	Ausbruch	Als Gefanger möchte ich eine Tool, so das ich ein Loch in die Wand machen kann, dam ich ausbrechen kann	Wenn ich das Tool benutze, entsteht ein Loch ein der Wand	100	13
3	Kommunikation	Als Gefangener möchte ich ein Tool, mit dem ich meine Mitgefangenen über den Ausbruch informieren kann	Wenn ich das Tool benutze, könne mich die anderen Gefangenen hören	100	20

Scrum-Prozess

- Letzte Gelegenheit:
 - Scrum Poster ist auf Laufwerk U:

Dieser PC > U (\\192.168.2.5) (U:) > KTN nur lesen > Dürr > Modul 426			
	Name	Änderungsdatum	Typ
	 Scrum_Guide_Poster_1.1	11.12.2015 16:02	Ado

Time-Boxing

- Was ist eine Time-Box?



- Lesen dieser Seite:
- <http://scrum-master.de/Scrum-Meetings/Time-Boxing>

Repetition

- Was sind die Grundsätze des agilen Manifests?
- Welche agilen Methoden kennen Sie?
- Was ist eine User Story?
- Welchen Umfang hat eine User Story?
- Mit welchen agilen Methoden entwickeln wir Software?
- Welchen Rollen Kenn Sie in Scrum?
- Und welche Artefakte?
- Was ist ein Product Backlog?
- Was ist ein Sprint Backlog?
- Wieviel Daily Scrums hat ein Sprint?
- Wieviel Sprints hat ein fertiges Produkt?
- Wer kann Scrum erklären?

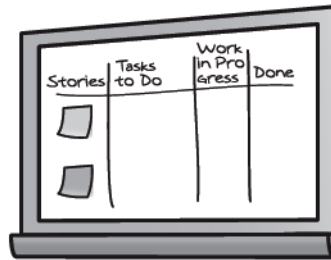
Trockenübung

User Stories:

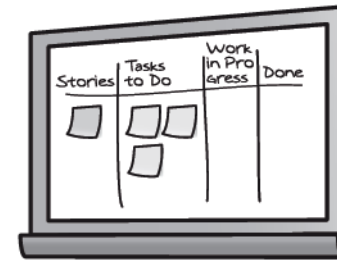
- *Ich will Intuitiv bedienbare Kundenkartei mit der gesamten Adresse und den Kontaktdaten.*
- *Jedem Kunden müssen beliebige Klassifizierungen zugeordnet werden können.*
- *Es müssen Mailings erzeugt werden können (alle oder nach Klassifizierung)*

Product Backlog und Sprint Backlog

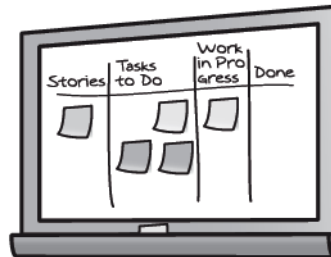
- Wir erstellen gemeinsam das Product Backlog mit den User Stories.
- Im Daily Scrum wird das Sprint Backlog aktualisiert:



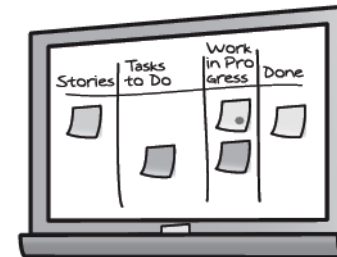
1. Selected Product Backlog am Taskboard unter „Stories“



2. Stories aufgeteilt in bearbeitbare 1-Tages „Tasks to Do“

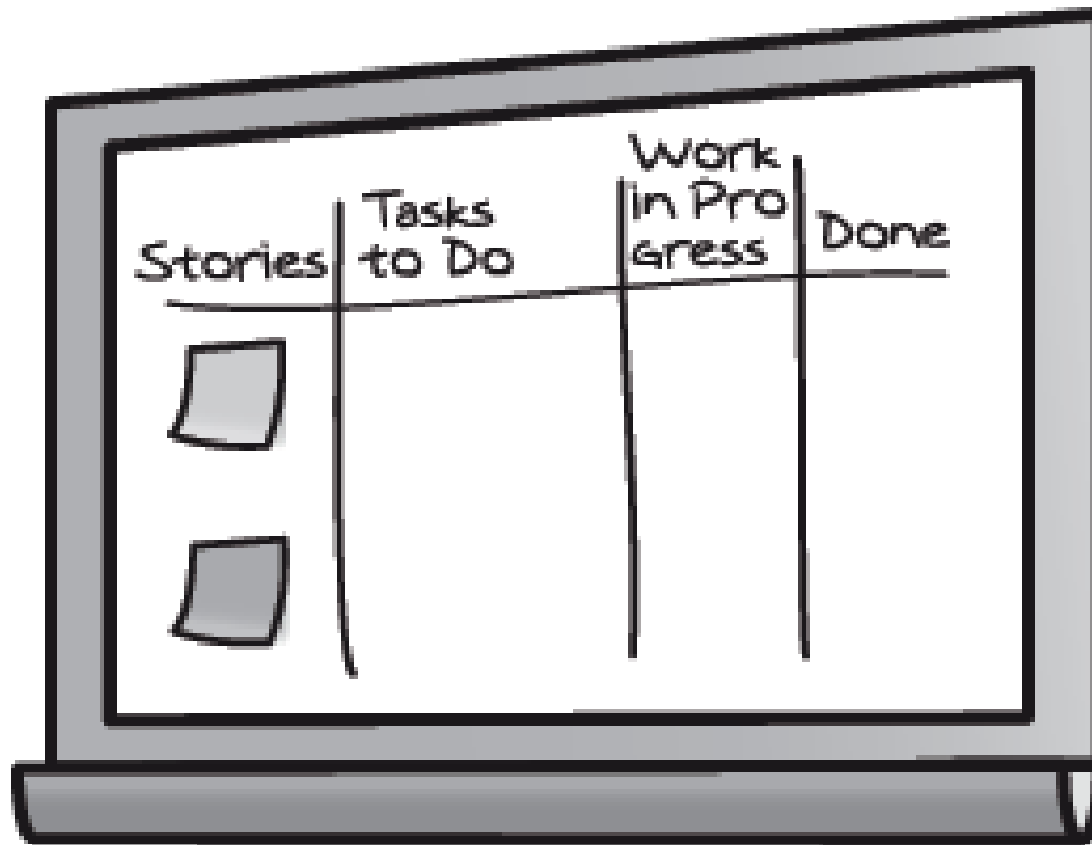


3. „Work in Progress“ zeigt Tasks, an denen jetzt gearbeitet wird, evtl. verändern sich Tasks, sind zerlegt



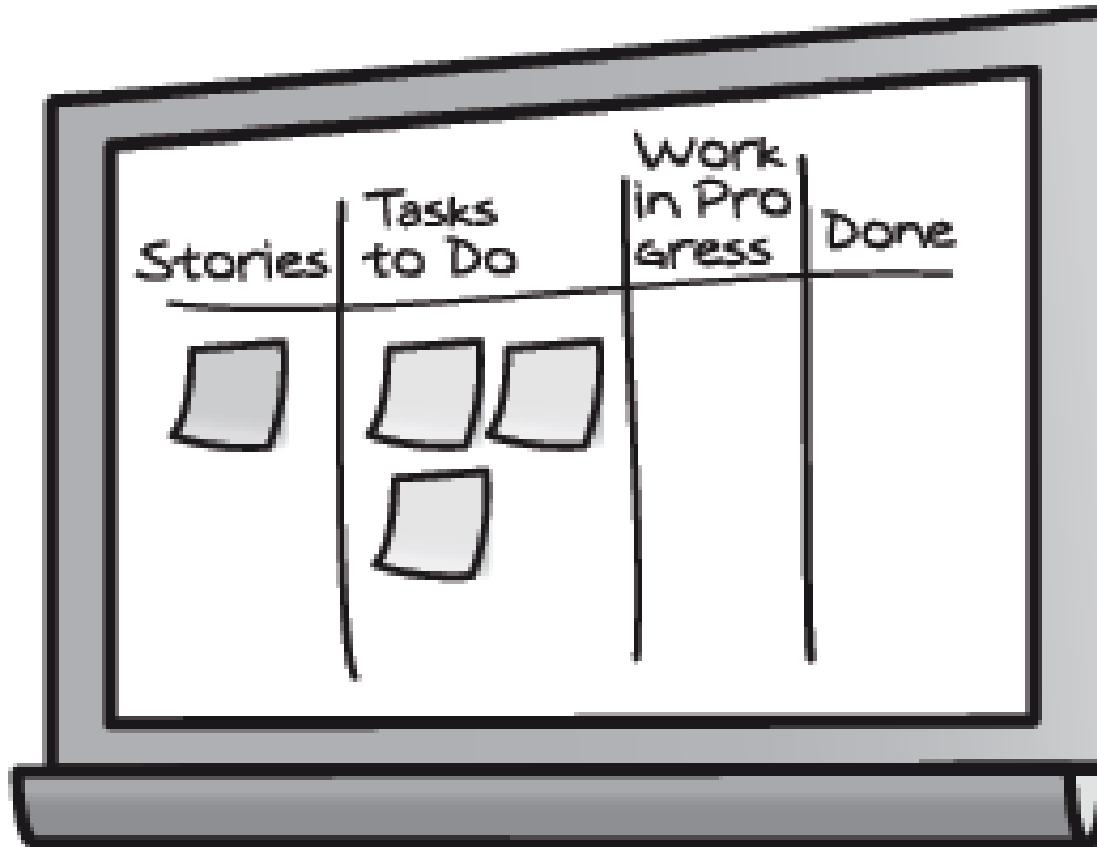
4. Tasks sind „done“ oder sind mit Punkten (Impediments) markiert, wenn sie nicht fertig wurden

Das Sprint Backlog im Detail



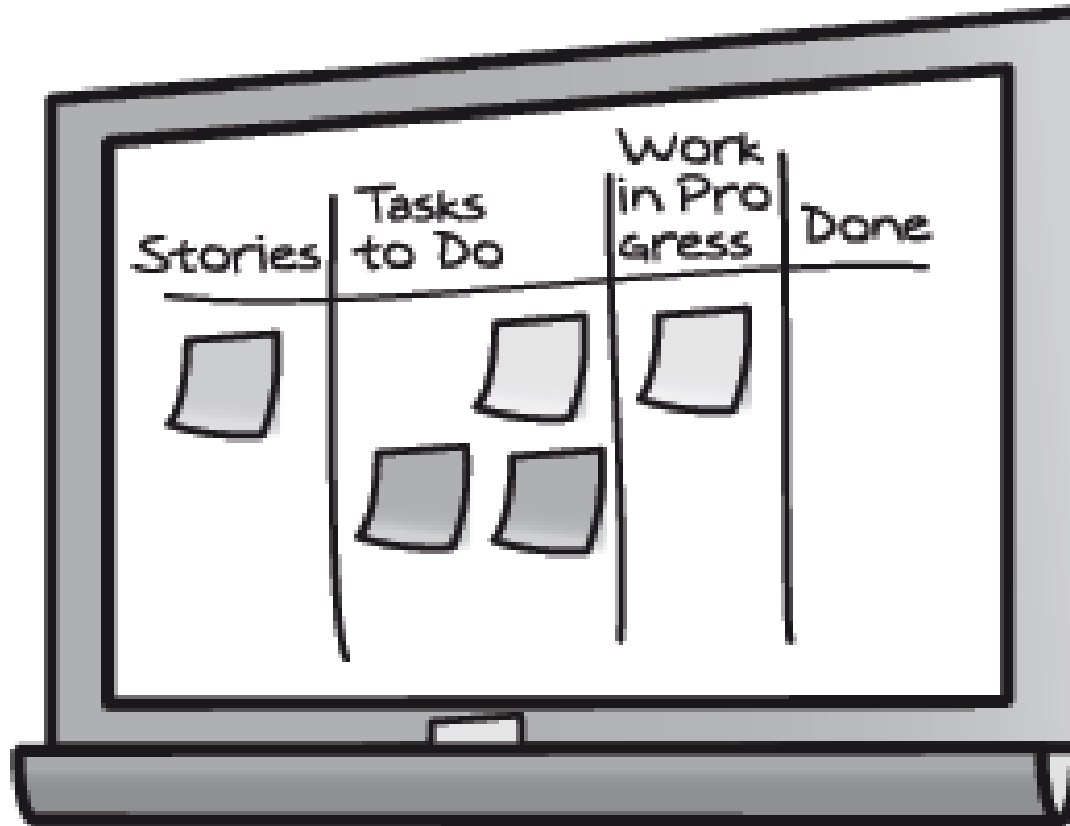
1. Selected Product Backlog am Taskboard unter „Stories“

Das Sprint Backlog im Detail



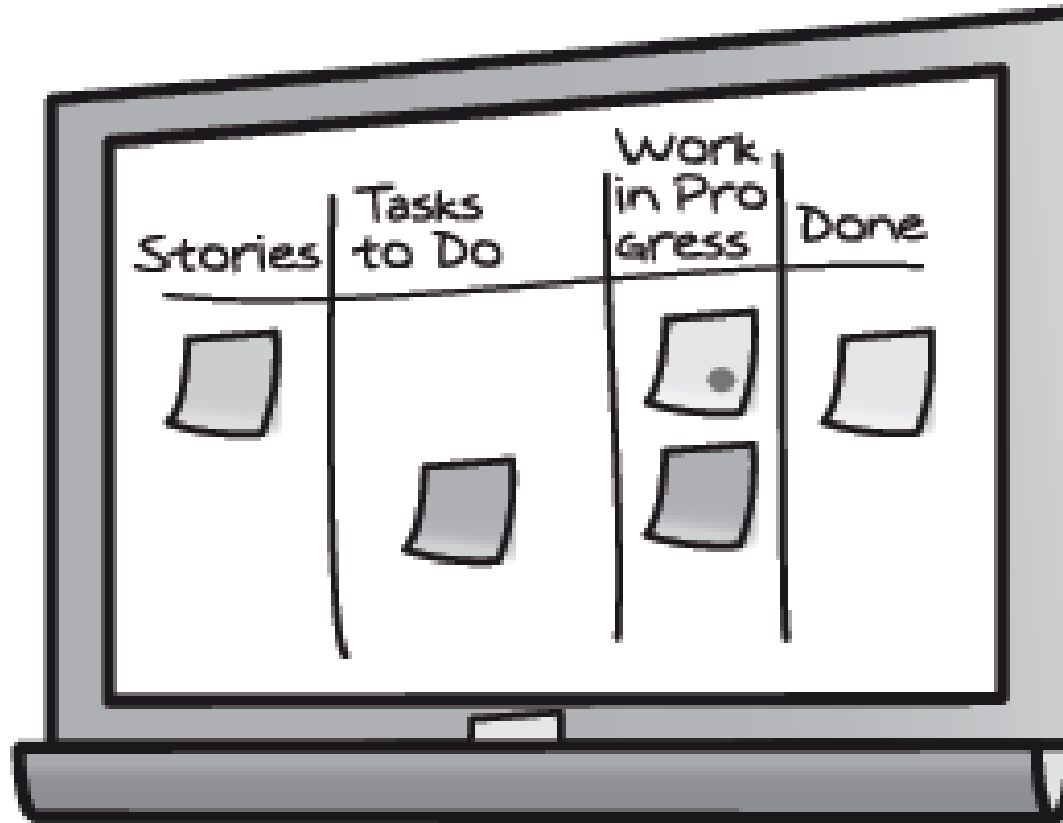
2. Stories aufgeteilt bearbeitbare 1-Tages „Tasks to Do“

Das Sprint Backlog im Detail



3. „Work in Progress“ zeigt Tasks, an denen jetzt gearbeitet wird, evtl. verändern sich Tasks, sind zerlegt

Das Sprint Backlog im Detail



4. Tasks sind „done“ oder sind mit Punkten (Impediments) markiert, wenn sie nicht fertig wurden

Youtube Video

- Scrum Rückblick
- <https://www.youtube.com/watch?v=OOdAwUXmL3E>

Programm vom 30.8.2019

- Repetition
- Ziele: Wie wird eine Scrum Projekt geschätzt?
 - User Stories 'Spielplätze' erstellen
(Gruppenarbeit: 2er Gruppen)
 - Relative Schätzung
 - Fibonacci Zahlenfolge
 - Planning Poker
 - Estimation Game
 - Magic Estimation

Spielplätze

- User Stories erstellen (2er Gruppen)
 - Zeit: 20 Minuten
 - Möglichst mehrere User Stories
 - Nicht vergessen:
Die Umsetzung einer User Story darf höchstens 1 Tag betragen.

Aufteilen in Sprint Backlog Tasks

Beispiel Hausreinigung



Relatives schätzen

- Relative Schätzung mit Beispiel
'CH – USA - D – FL'
- Wieviel Quadratkilometer haben diese Länder?
- Relativ zueinander?
- Je grösser der Abstand desto ungenauer!
- Fibonacci-Reihe?
- Abstand zwischen den Zahlen wird immer grösser:
1-2-3-5-8-13-21-34-55-89-144 -...
In Scrum : 0, ½, 1, 2, 3, 5, 8, 13, 20, 40, 100, ?, Kaffee

Planning Poker

1. Jeder aus dem Team gibt eine Schätzung für eine Userstory ab: Karte (Smartphone) ist verdeckt - es wird nicht diskutiert!
(Ist die User Story unklar, kann auch ein '?' gezeigt werden. Der Product Owner muss die Story dann erklären)
2. Sind alle parat, zeigt jeder seine Zahl. Sind die Zahlen nahe beieinander ist die Schätzung relativ sicher.
3. Bei grossen Unterschieden wird über die User Story diskutiert und es beginnt wieder bei Punkt 1 (oben).

Programm 6.9.2019

- Schätzen in Scrum
- Ankündigung Prüfung 20.9.2019

Repetition

- Auf welcher Basis schätzen wir Projekte in Scrum?
 - Unechte Fibonacci-Reihe
- Welche Arten zum schätzen habe wir kennengelernt?
 - Planning Poker
 - Estimation Game
 - Magic Estimation

Schätzen in Scrum

Damit die Projektkosten geschätzt werden können, müssen alle (!) User Stories vom ganzen Product Backlog geschätzt werden.

- Entweder mit:
 - **Planning Poker**
- oder mit:
 - **Estimation Game**
- oder mit:
 - **Magic Estimation**

Planning Poker

1. Jeder aus dem Team gibt eine Schätzung für eine Userstory ab: Karte (Smartphone) ist verdeckt - es wird nicht diskutiert!
(Ist die User Story unklar, kann auch ein '?' gezeigt werden. Der Product Owner muss die Story dann erklären)
2. Sind alle parat, zeigt jeder seine Zahl. Sind die Zahlen nahe beieinander ist die Schätzung relativ sicher.
3. Bei grossen Unterschieden wird über die User Story diskutiert und es beginnt wieder bei Punkt 1 (oben).

Magic Estimation

1. Zuerst wird die Scrum-Poker Karten nach den Grössen gelegt (Schätzskala)
2. Dann werden die User Stories zu den entsprechenden Poker-Karten gelegt.
3. Jeder vom Team legt die User Story auf die Skala – die User Stories können von andern Team Mitgliedern verschoben werden
4. Wird eine User Story hin und hergeschoben, nimmt der Product Owner die User Story weg. Die wird später nochmals besprochen!

Estimation Game

1. Jemand aus dem Entwicklerteam liest die Story vor und legt die Karte hin.
2. 2.te Story wird vorgelesen und nach einer Komplexität unter oder über die erste Karte gelegt.
3. Der dritte kann die Reihenfolge verändern, oder eine neue Karte nehmen und legen.
4. Sämtliche Karten werden vorgelesen und eine Reihenfolge bestimmt (es gibt dabei keine Diskussion)
5. Dann werden die User Stories mit den Scrum-Poker Karten bewertet.

Story Points

- Die Werte der Poker Karten (für alle User Stories) werden zusammengezählt, das ergibt die Anzahl Story Points – Das ist die initiale Schätzung des Projekts.
- Um den Kunden einen Preis für das Projekt sagen zu können, wird **eine** User Story ausgesucht, bei der man genau weiss wie lang man braucht, diese zu realisieren. (In Teilschritte brechen, Beispiel Hausreinigung)
- Dann weiss man die Zeit (CHF/h), die die User Story braucht, um umgesetzt zu werden (Anzahl Story Points). Und man hat die Anzahl Story Points für das gesamte Projekt.

Schätzung Spielplätze IT2b_2

<u>Scrum Poker</u>	<u>Maple Est.</u>	<u>Estimation Game</u>
40	40	0,5
5	100	2
8	20	3
5	100	5
	5	5
2	5	13
	3	40
100	2	100
20	1	100
100		
1		
<hr/> 281	<hr/> 276	<hr/> 268,5

Schätzung Spielplätze IT2b_1

<u>Planning Folie</u>	<u>Magic Estimation</u>	<u>Estimation Game</u>
20 +	1	2
1 +	2	2
2 +		
20 +	5	5
8 +	8	13
100 +	13	20
20 +	13	40
13 +	20	100
<hr/>	<hr/>	<div style="border: 1px solid black; padding: 5px; display: inline-block;">185</div>
= 204	100	
	162	

YouTube-Video zum Schätzen (53')

https://www.youtube.com/watch?v=n8_WeJcMJkc

Programm vom 13.9.2019

Unterrichtsziele:

- Repetition
- Erstellen einer Präsentation über agile Methoden
- Präsentation andere agile Methoden
- Ankündigung Prüfung am 20.9.2019

Welche agilen Methoden kennen Sie?



Andere agile Methoden

Auftrag Gruppen zu je 2 Personen erstellen zu den drei Methoden (ohne Scrum) eine Präsentation mit Power Point von mindestens 5 Folien (Schriftgrösse min. 22Pkt), Quellen Wikipedia, Internet...

- eXtreme Programming
- Feature Driven Development
- Kanban

Anschliessende werden die Lösungen präsentiert.

Andere agile Methoden

Auftrag Gruppen zu je 2 Personen erstellen zu den vier Methoden eine Präsentation mit Power Point von mindestens 10 Folien (Schriftgrösse min. 22Pkt), Quellen Wikipedia, Internet...

- eXtreme Programming
- Feature Driven Development
- Kanban

Anschliessende werden die Lösungen präsentiert.

SCRUM - Prüfung

- Die erste Prüfung findet am 20.9.2019 statt:
 - Theorie und Praxis
- Die Präsentation ist auf Laufwerk U (nur Lesen).
- An den Prüfungen darf die Präsentation **nicht** verwendet werden.
- Selbst erstellte handgeschriebene Unterlagen 1 Seite A4 sind erlaubt.
- An einer Nachholprüfung dürfen keinerlei Unterlagen verwendet werden und die Prüfung wird schwieriger!

Programm vom 20.9.2019

Unterrichtsziele:

- Fertigstellen der Präsentation
- Präsentation andere agile Methoden
- Prüfung 45'

Programm 4. Oktober 2019

Unterrichtsziele:

- Versionsverwaltungssysteme kennen
- Git anwenden

Was wir benötigen!

- Wenn verschiedene Entwickler an einem(!) Projekt arbeiten, müssen die verschiedenen Softwarepakete, an denen gearbeitet wird, vereint werden!
- Wie können wir das bewerkstelligen?
- Wir verwenden Versionsverwaltung-Systeme
- Ein gemeinsam genutztes Versionsverwaltungssystem, zum Beispiel: <https://git-scm.com/>

Git

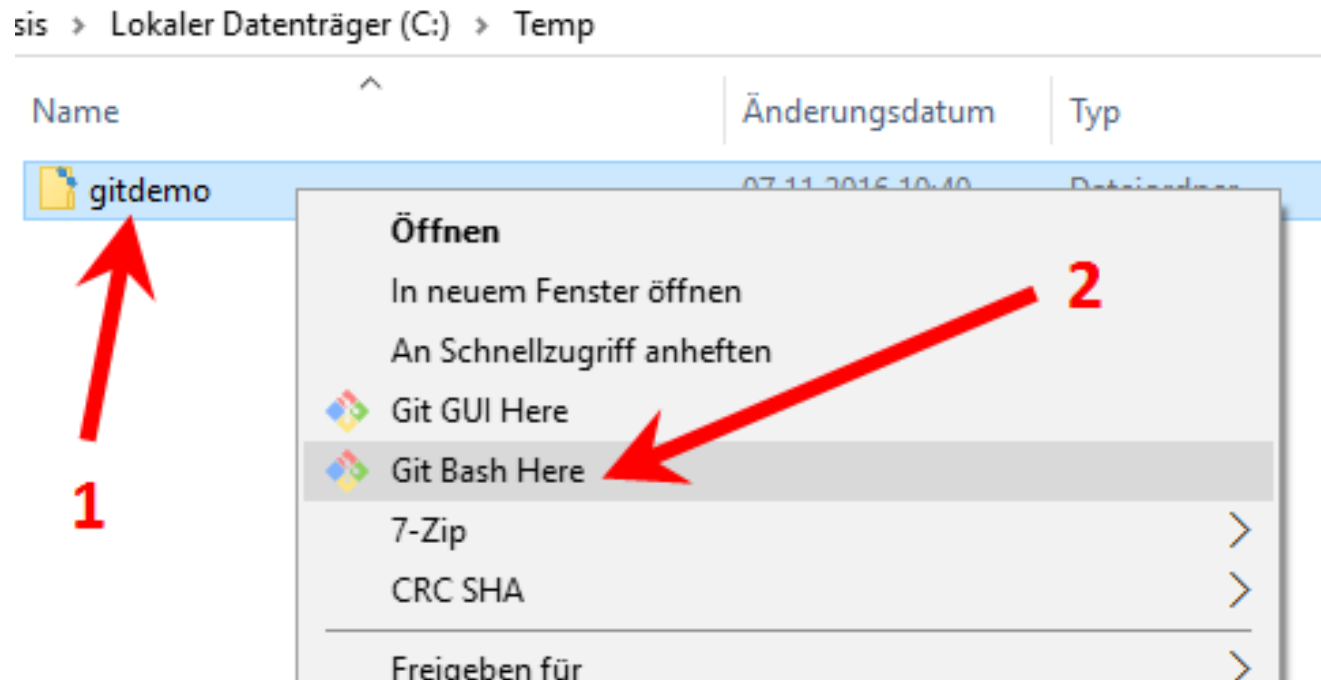
- 'Git' nur für agile Software?
- Grundsätzlich für alles, was Personen gemeinsam erstellen.
- Git in der Cloud...
 - [BitBucket](#)
 - [GitHub](#)
 - [GitLab](#)

Git Workshop

- Schritt für Schritt – Kapitel 1 bis Kapitel 2.6
– bis und mit Seite 21
- Lesen und Beispiele durchführen
- Wichtigste Git-Befehle notieren
- Sie müssen damit rechnen, dass der eine oder andere Git-Befehl an einer Prüfung vorkommen wird

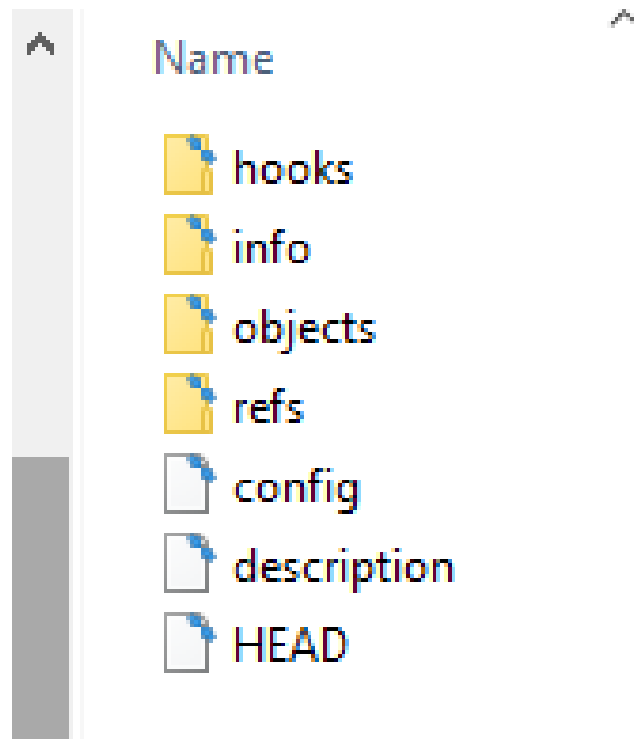
gitdmo erstellen

- Git starten



gitdemo und .git öffnen

rsis > Lokaler Datenträger (C:) > Temp > gitdemo > .git >



Repetition git Befehle

- > git config --global user.name hmustermann
- > git config --global user.email hans@mustermann.de
- > git init
- > git add foo.txt bar.txt
- > git commit --message "wird im log angezeigt"
- > git status
- > git diff
- > git log
- > git clone
- > git pull
- > git push

Hausaufgaben

- Nächstes Mal das Buch 'Scrum' oder entsprechendes pdfmitnehmen!



Letzte Präsentationen IT 2b_1

- Bitte alle Präsentationen auf Netzlaufwerk laden!
- Präsentation Gruppe
 - Shannon Schürch, Jonathan Genet
- Noten werden nächstes Mal bekanntgegeben

Letzte Präsentationen IT 2b_2

- Bitte alle Präsentationen auf Netzlaufwerk laden!
- Präsentationen Gruppen
 - Hechenberger Nicolas, Lauter Silvester
 - Lazic Ksenija, Vukorepa René
 - Peyer Clemens, Rawyler Hubert
- Noten werden nächstes Mal bekanntgegeben

Repetition git Befehle

- > git config --global user.name hmustermann
- > git config --global user.email hans@mustermann.de
- > git init
- > git add foo.txt bar.txt
- > git commit --message "wird im log angezeigt"
- > git status
- > git diff
- > git log
- > git clone
- > git pull
- > git push

Das Entwicklerteam

- Lesen im Buch 'Scrum':
4.3 Das Team – Die Spezialisten (Seite 65 – 75)
ev.
2.1 Das Entwicklungsteam – die Spezialisten (Seite 64 - 74)
- Entwicklungsteam bilden
- Scrum Projekt: Sichere Passwörter

Entwicklungsteams bilden

IT2b_1

- Entwicklungsteams à 2 bis 3 Personen werden gebildet:

Twerenbold Dario	Ochmann Phil	
Brendle Joel	Bühler Nicola	Filli Ursin
Jaiswal Sidharth	Khankah Ruanin	Sivakumaran Sivashan
Genet Jonathan	Kägi Artem	Schürch Shannon

Sprint Backlog Tasks

- Nachdem die Anforderungen für den nächsten Sprint vom Product Owner ausgewählt wurden, werden sie in **Aufgaben**, die innerhalb eines Tages erledigt werden können, unterteilt. Deshalb nennen wir die **Bestandteile** des Sprint Backlogs auch *Sprint Backlog Tasks*. Genau wie im Product Backlog wird für jedes Bestandteil des Sprint Backlogs eine **Aufwandsschätzung** durchgeführt. Damit wissen wir welche Tasks in den aktuellen Sprit aufgenommen werden können (Time Box).

User Story

Auftrag:

- In 2er oder 3er Gruppen (5 agile Teams), jede Gruppe erstellt zum agilen Projekt 'Sichere Passwörter' eine User Story.
- 1 Person des Teams ist dabei der Product Owner.

Programm vom 1. November 2019

- Wir arbeiten am Projekt 'Sichere Passwörter'.
- Es wird mit 'Scrum' gearbeitet.
- Das Projekt wird mit PHP erstellt.
- Jede Gruppe hat Ihr Sprint Backlog bereit und offen
- Ist das Sprintbacklog nicht vorhanden gibt es 0.5 Noten Abzug

Hochladen auf GitHub

- GitHub auf Wikipedia:
- <https://de.wikipedia.org/wiki/GitHub>
- Anleitung Hochladen auf GitHub:
<https://legacy.thomas-leister.de/github-fuer-anfaenger-repository-anlegen-und-code-hochladen/>

Sichere Passwörter

Original

SecureCode: 3# ID: 8344910

1 2 3 4 5 6 7 8 9 10 11

1 2 3 4 5 6 7 8 9 10 11

SAVERNOVA

© Copyright 2007 SAVERNOVA™ All rights reserved www.savernova.com

Back Up

SecureCode: 3# ID: 8344910

1 2 3 4 5 6 7 8 9 10 11

1 2 3 4 5 6 7 8 9 10 11

SAVERNOVA

© Copyright 2007 SAVERNOVA™ All rights reserved www.savernova.com

Sichere Passwörter

- Wir werden ein agiles Projekt in PHP erstellen. Und zwar gemäss der untenstehenden Webseite:
- Die Passwortkarte:
<http://www.tagesanzeiger.ch/digital/daten/sichere-passwoerter-und-wie-man-sie-sich-merkt/story/18097263>
- Die horizontale und vertikale Anzahl Buchstaben/ Zahlen/Sonderzeichen kann soll ausgewählt werden.

Programm vom 01.11 2019

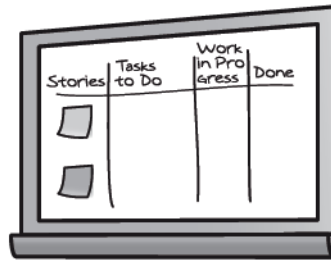
- Am Projekt 'Sichere Passwörter' weiter arbeiten.
- Es wird mit 'Scrum' gearbeitet
- Jede Gruppe hat Ihr Sprint Backlog bereit und offen!
- Ist das Sprintbacklog nicht vorhanden gibt es 0.5 Noten Abzug

User Stories erstellen

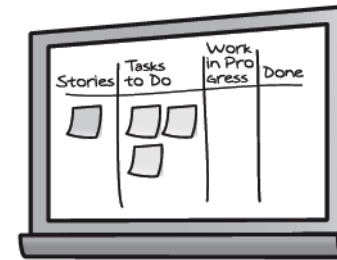
- Das Entwicklerteam erstellt mindestens 4-5 User Stories
- und zerlegt die User Stories in 1 Tages 'task to do'.

Product Backlog und Sprint Backlog

- Wir erstellen gemeinsam das Product Backlog mit den User Stories.
- Im Daily Scrum wird das Sprint Backlog aktualisiert:



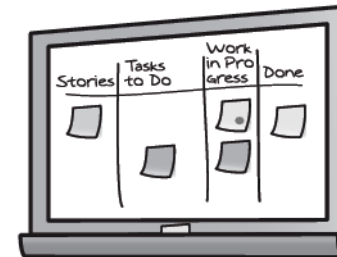
1. Selected Product Backlog am Taskboard unter „Stories“



2. Stories aufgeteilt in bearbeitbare 1-Tages „Tasks to Do“



3. „Work in Progress“ zeigt Tasks, an denen jetzt gearbeitet wird, evtl. verändern sich Tasks, sind zerlegt



4. Tasks sind „done“ oder sind mit Punkten (Impediments) markiert, wenn sie nicht fertig wurden

Aufteilen in Sprint Backlog Tasks

Beispiel Hausreinigung



Auftrag

- In 2er oder 3er Gruppen (5 agile Teams), jede Gruppe erstellt zum agilen Projekt 'Sichere Passwörter' eine User Stories.
- 1 Person des Teams ist dabei der Product Owner.
- Die horizontale und vertikale Anzahl Buchstaben/ Zahlen/Sonderzeichen kann ausgewählt werden.

Sichere Passwörter

Original

SecureCode: 3# ID: 8344910

1 2 3 4 5 6 7 8 9 10 11

1 2 3 4 5 6 7 8 9 10 11

SAVERNOVA

© Copyright 2007 SAVERNOVA™ All rights reserved www.savernova.com

Back Up

SecureCode: 3# ID: 8344910

1 2 3 4 5 6 7 8 9 10 11

1 2 3 4 5 6 7 8 9 10 11

SAVERNOVA

© Copyright 2007 SAVERNOVA™ All rights reserved www.savernova.com