

Projektlabor Medizintechnik 2

Scrum in a Nutshell 2

Prof. Dr. Mark Hastenteufel

User Stories

Notation von User Stories

Als <Benutzer Rolle> will ich <das Ziel>[, so dass <Grund für das Ziel>].

As a <role> I want to <activity> [so that <business value>].

Beispiele:

Benutzer Rolle / Role	Ziel / Activity	Grund / Business value
Als Bewerber	will ich mich anmelden,	so dass ich mein Profil einstellen kann.
Als Headhunter	will ich nach Bewerbern suchen,	so dass ich Kontakt aufnehmen kann.
Als Bewerber	will ich Firmen bewerten,	so dass es sich unter anderen Bewerbern herumspricht.

Eigenschaften von User Stories: INVEST

- **I**ndependent (unabhängig)

Nicht immer machbar, Grundsatz der inkrementellen Entwicklung.
Beispiel:
Als Benutzer will ich mich registrieren.
Als Benutzer will ich mich anmelden.

- **N**egotiable (verhandelbar)

Details werden z.g. Zeit zwischen PO und Team verhandelt. Kein Vertrag!

- **V**alueable to the customer
(liefern Mehrwert)

Die Fertigstellen einer US liefert einen Mehrwert für den Kunden.
Als Entwickler möchte ich Logging Informationen speichern damit ich Fehler besser finden kann. ☹

- **E**stimatable (schätzbar)

Zu offen und schwammig → nicht schätzbar.
Als Benutzer möchte ich mein Profil managen. (☹ Was heißt das?)

- **S**mall (klein genug für einen Sprint)

User Stories sollen in einem Sprint von dem Team umsetzbar sein.

- **T**estable (testbar)

Akzeptanzkriterien zu einer User Story unterstützen das testen.
Als Benutzer möchte ich so schnell wie möglich ein Suchergebnis angezeigt bekommen. (☹ Was heißt so schnell wie möglich?)

Akzeptanzkriterien

Als Benutzer will ich mich anmelden.

Akzeptanzkriterien:

- Anmeldung mit email und Passwort
- Konto wird nach drei fehlerhaften Versuchen gesperrt
- Nach erfolgreicher Anmeldung: Seite mit meinen Kontodaten.

Akzeptanzkriterien

- detaillieren Verständnis der User Story.
- dienen zur Abnahme der User Story.
- dienen als Grundlage für ausführlichere Tests.

User Stories: Schneiden nach Daten

Beispiele:

Als Benutzer will ich mein Profil einstellen.

- Als Benutzer will ich mein Foto einstellen.
- Als Benutzer will ich meinen Lebenslauf als PDF einstellen.
- Als Benutzer will ich meine Erfahrungen und Kenntnisse als Freitext einstellen.

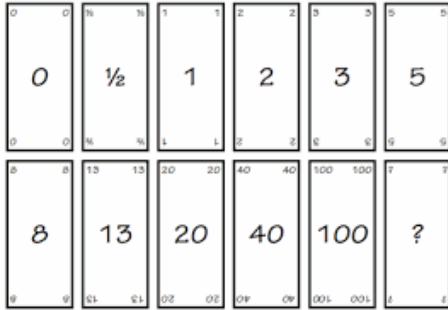
Als Benutzer will ich mit Kreditkarten zahlen können.

- Als Benutzer will ich mit Mastercard zahlen können.
- Als Benutzer will ich mit VISA zahlen können.
- Als Benutzer will ich mit Amex zahlen können.

Agiles Schätzen

Planning Poker

Planning poker: Method and tools



„Hardware“

Pointing Poker



Donate

\$10.00

Donate

Welcome to pointing poker (aka [planning poker](#))! Online, virtual and co-located agile teams use this application during their planning/pointing sessions to effectively communicate points for stories.

Pointing Session

Start a Session

... or ...

Session ID

Join a session

Retrospective Session

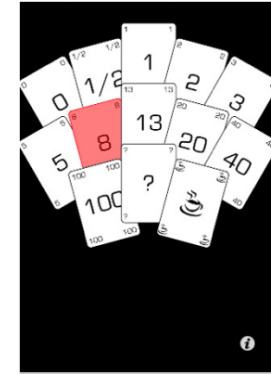
Start a Session

... or ...

Session ID

Join a session

Online collaboration tool, eg:
<https://www.pointingpoker.com/>



Planing Poker Apps,
see App Store

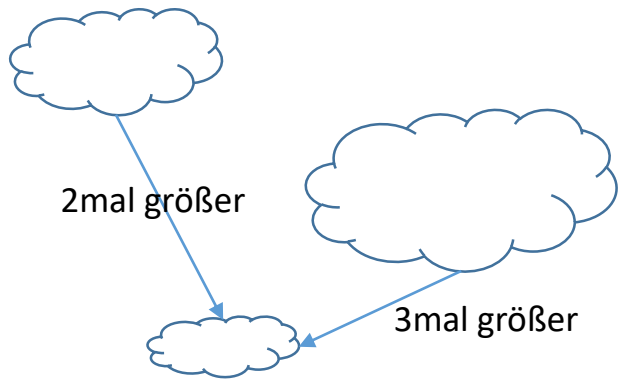
Grundlage des agilen Schätzens ist der (priorisierte) Backlog mit User Stories.
Reihenfolge des Schätzens entspricht der Priorisierung.

Geschätzt wird die **relative Größe** einer Story (in Relation zu den anderen Stories)
Oft verwendet: 1,2,3,5,8,13,21, 34, ...

Vorgehen beim agilen Schätzen mit Planning Poker:

- Referenz Stories festlegen: Was ist eine 1? Was ist eine 5?
- Max. zwei Runden
- Priorisiertes Vorgehen anhand Backlog
- Alle Rollen müssen involviert sein
- Product Owner erklärt Story (timeboxed, z.B 1 min)
- Erste Diskussionsrunde mit PO (timeboxed, z.B. 1 min)
- Erste SchätZRunde
- Diskussion auf Basis der min. und max. Schätzungen
- Zweite SchätZRunde
- Einigung

Von der Größe zur Zeit



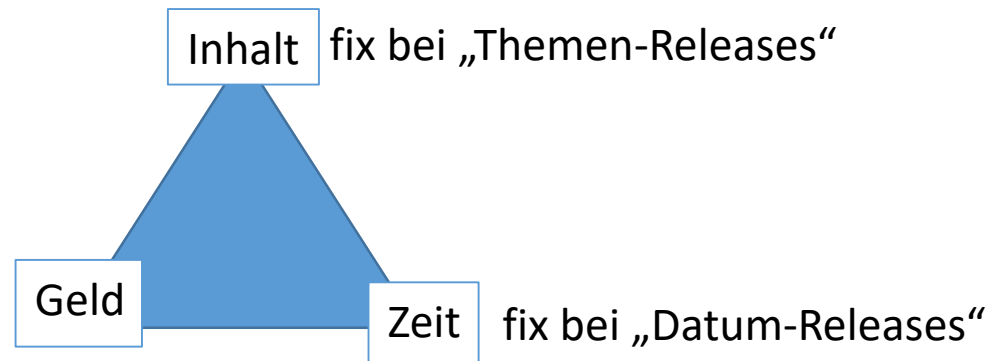
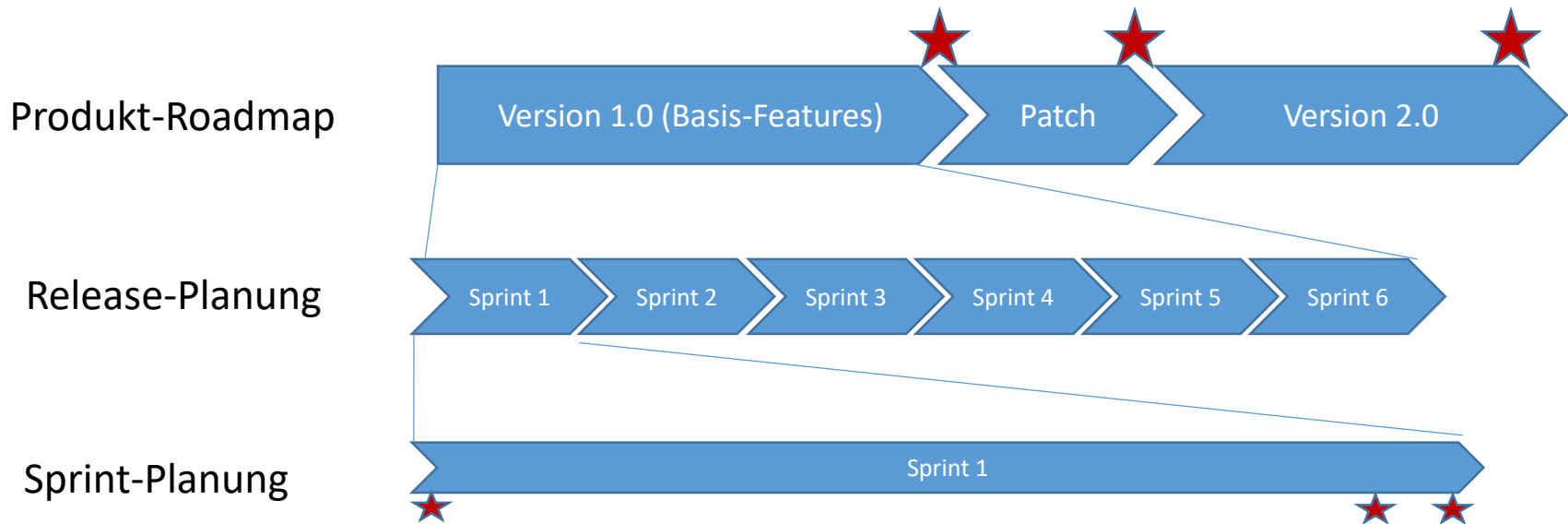
relative Größe



Zeit (Kosten)

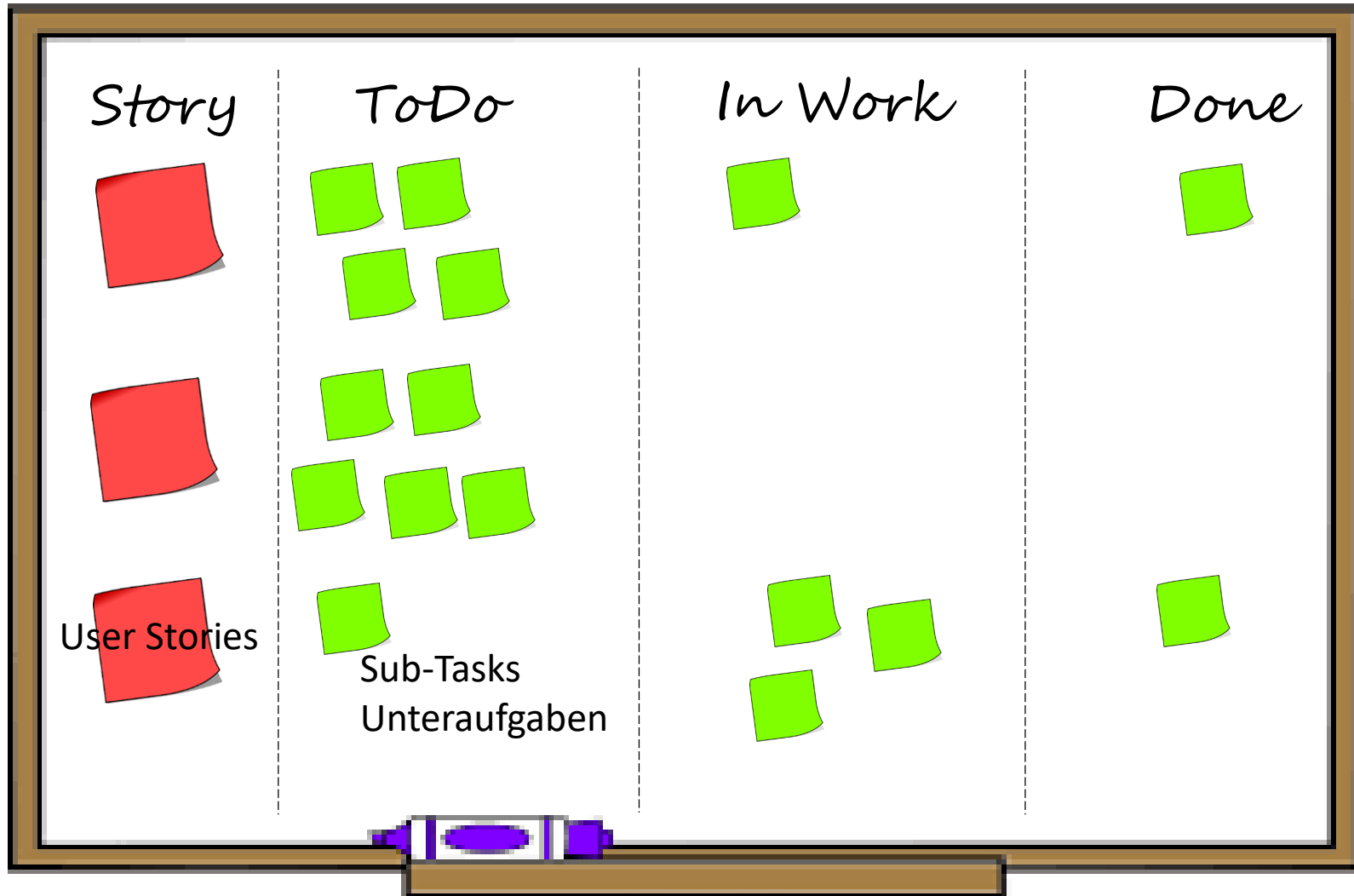
Agile Planung

Planung auf verschiedenen Ebenen



Ergebnis einer Sprint-Planung: Sprint-Backlog

Grundlage der Sprint-Planung ist der priorisierte Produkt-Backlog. Hieraus wird ein Sprint-Backlog (ausgewählte User Stories plus zugehörige Sub-Tasks) erstellt.



Release-Burndown

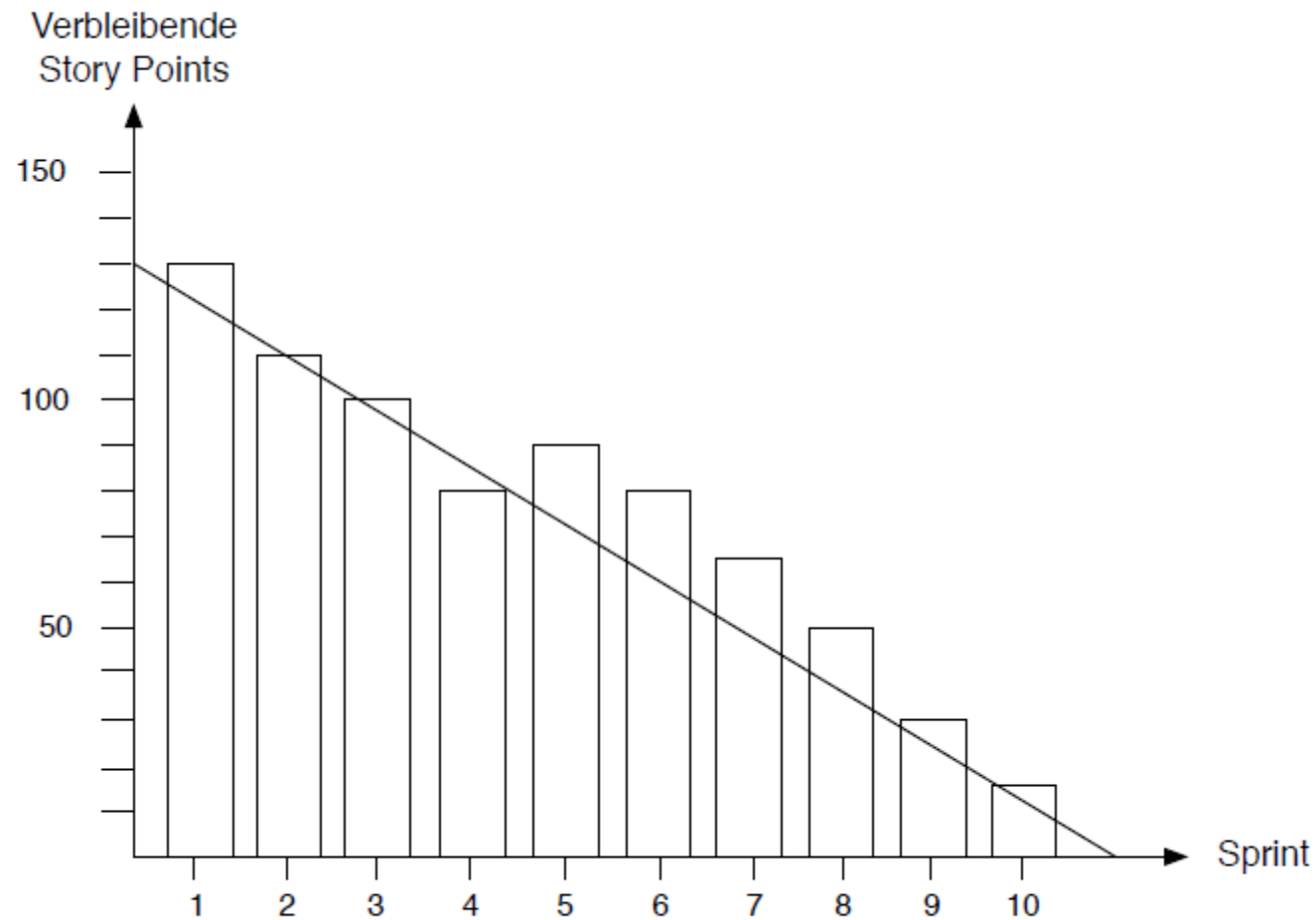


Abbildung 12.5: Das Release-Burndown-Chart dokumentiert den Projektfortschritt

Eine Methode zur Durchführung von Retrospektiven

Retrospektiven

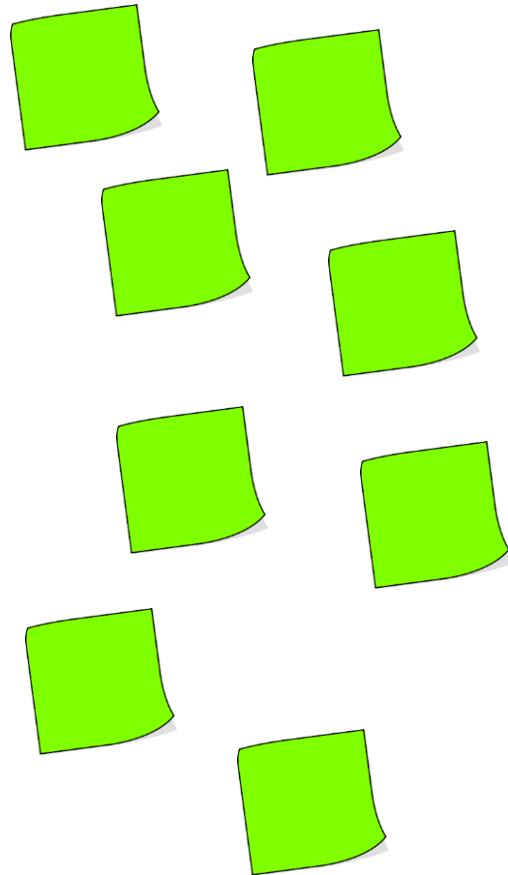
Vorbereitung:

- Regeln setzen
- Vertrauen aufbauen
- Teilnehmerkreis definieren: keine Teammanager, Vorgesetzte, ...
- Ablauf erklären

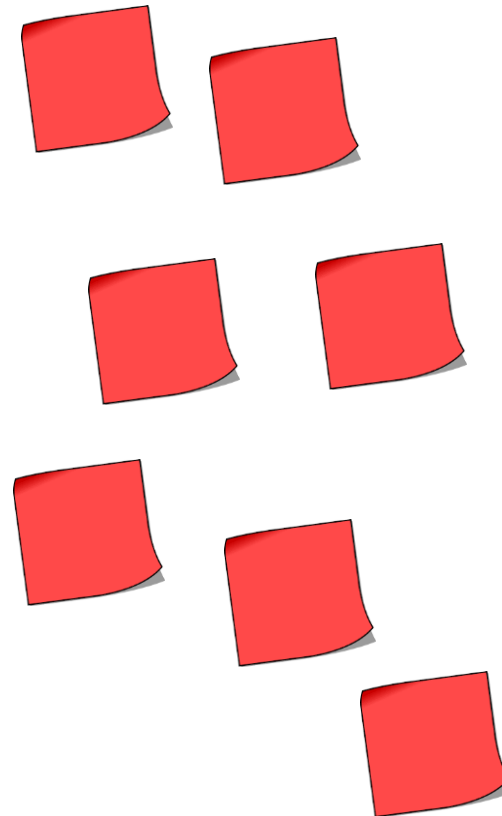


Retrospektive (1): Daten sammeln

Was war gut?
Was sollte man beibehalten?

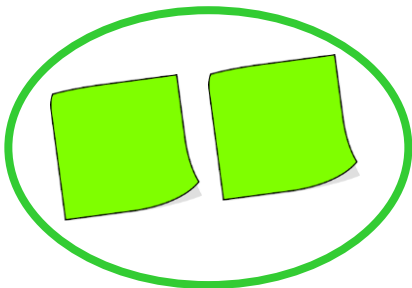
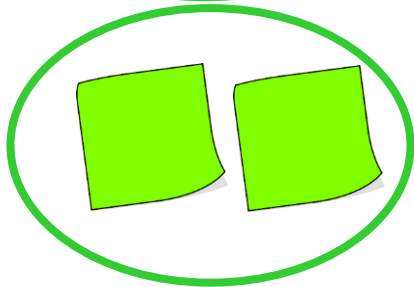
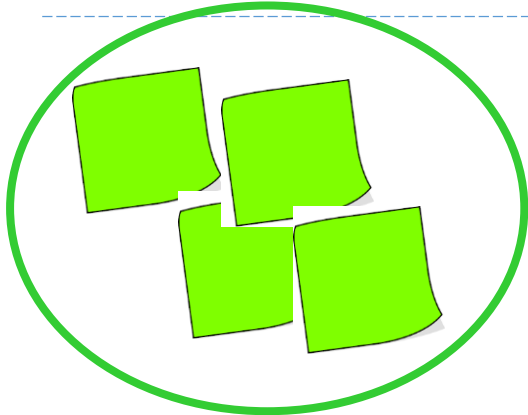


Was war nicht gut?
Was sollte man ändern?

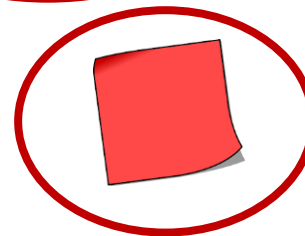
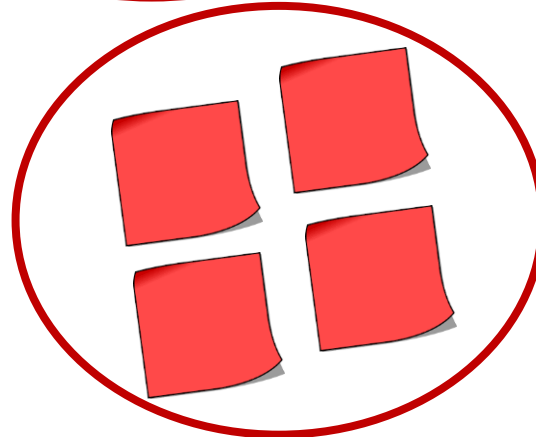
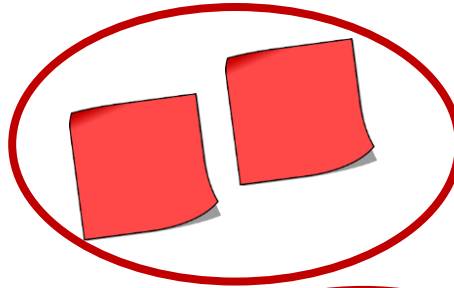


Retrospektive (2): Gruppieren

Was war gut?
Was sollte man beibehalten?

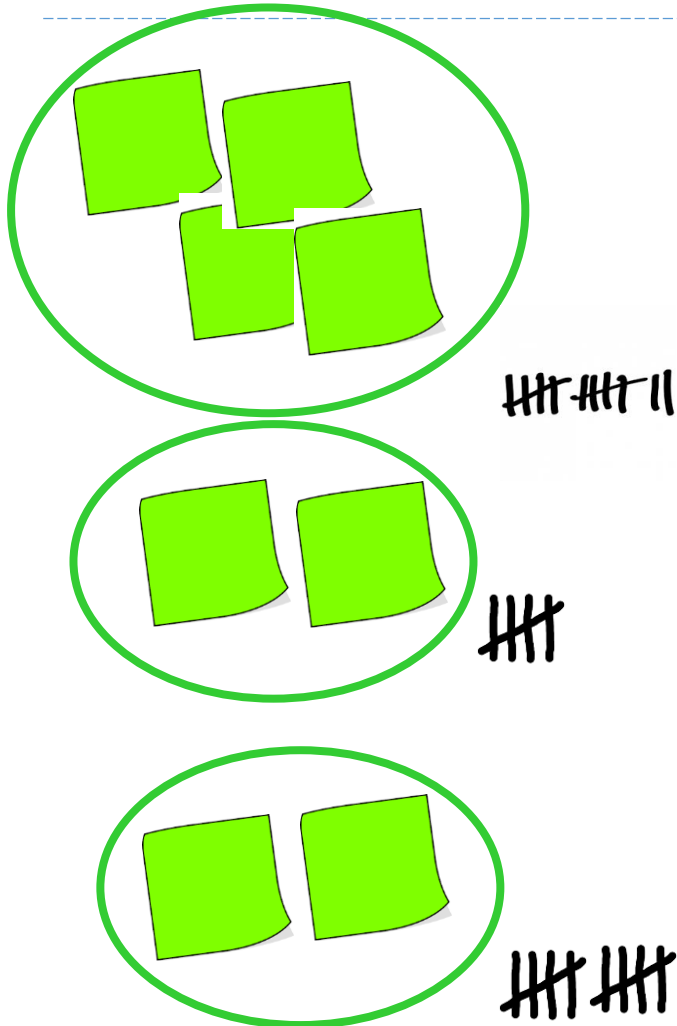


Was war nicht gut?
Was sollte man ändern?

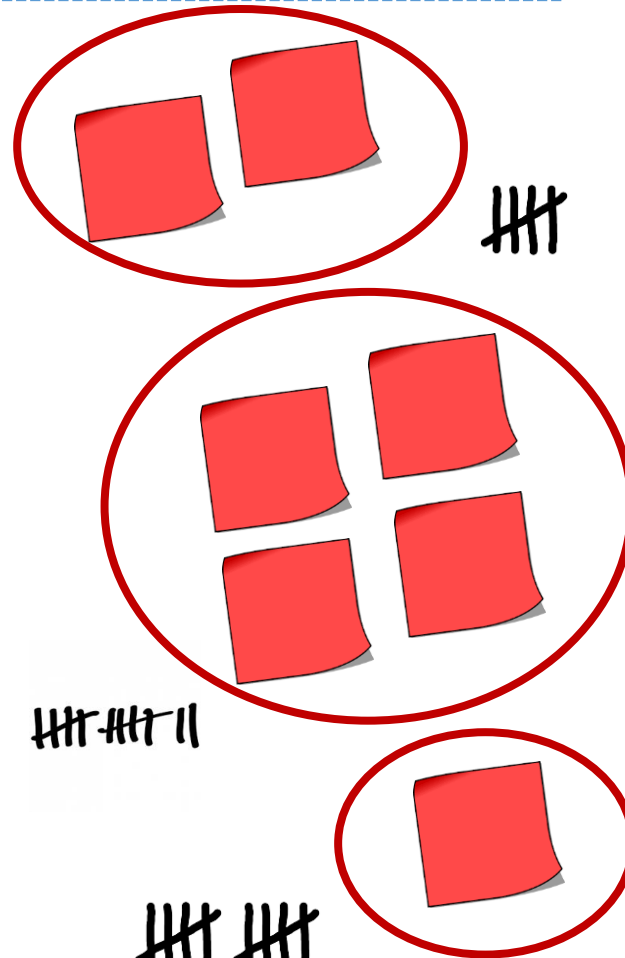


Retrospektive (3): Priorisieren

Was war gut?
Was sollte man beibehalten?

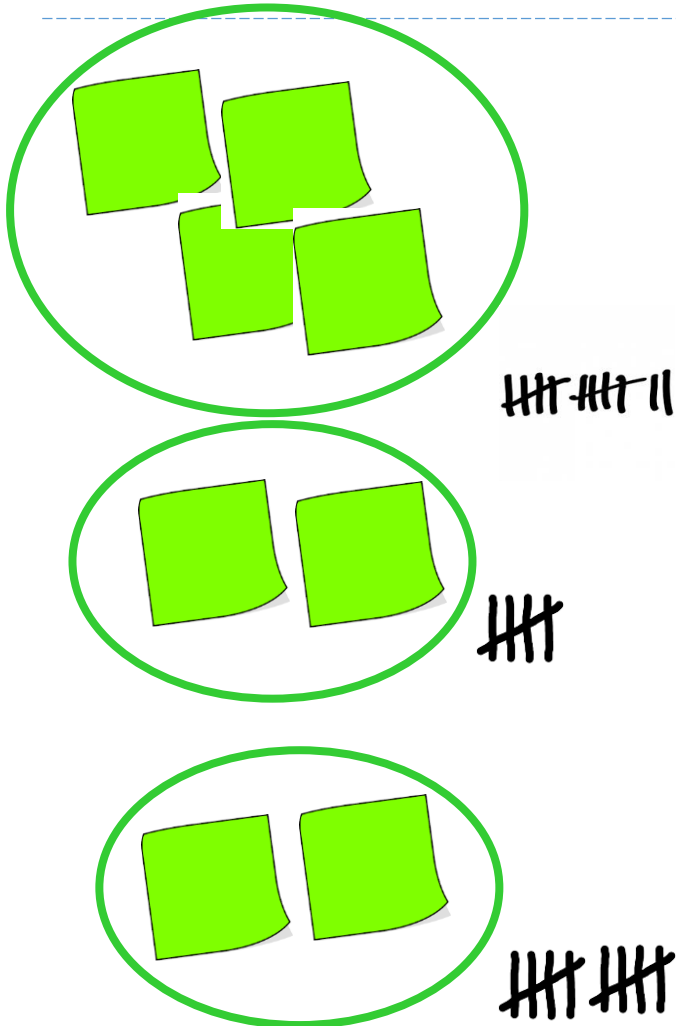


Was war nicht gut?
Was sollte man ändern?

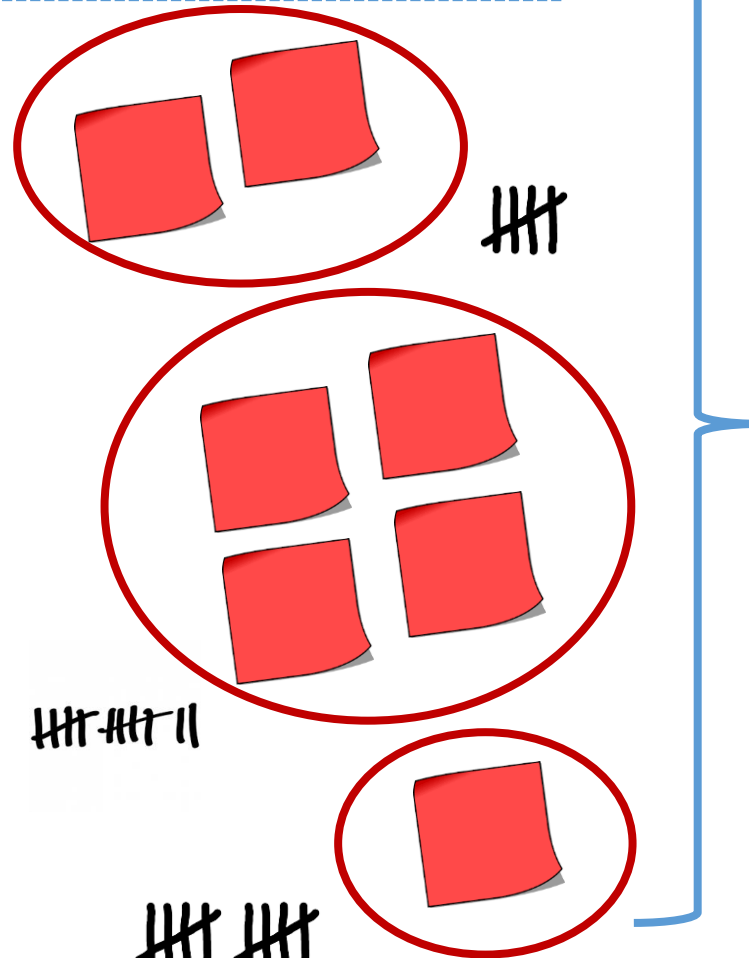


Retrospektive (4): Aktionen definieren

Was war gut?
Was sollte man beibehalten?



Was war nicht gut?
Was sollte man ändern?



Aktionsplan:

- 1)
- 2)
- 3)