

SCRUM

Agiles Projektmanagement

Eine Alternative zum Wasserfallmodell

Deggendorf, November 2011

Vortrag im Rahmen der Vorlesung:

Aktuelle Themen der Wirtschaftsinformatik bei Prof. Dr. Herde im WS 2011/12

Inhalt

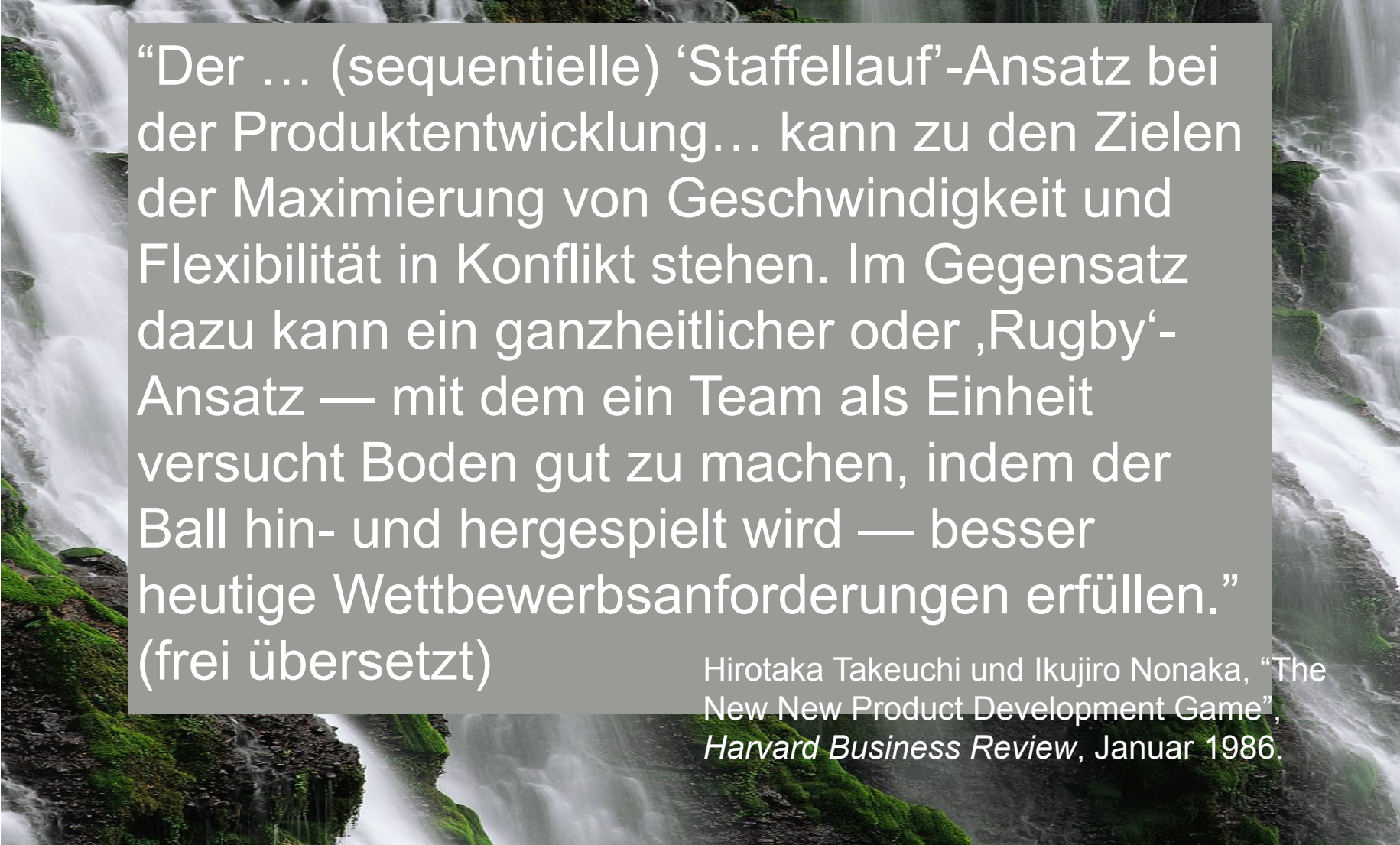
A SCRUM im Überblick

B Unterschiede zum Wasserfallmodell

C Beispiel aus der Praxis

A SCRUM im Überblick

Wir verlieren den Wettlauf – oder wenn Dämme brechen



“Der ... (sequentielle) ‘Staffellauf’-Ansatz bei der Produktentwicklung... kann zu den Zielen der Maximierung von Geschwindigkeit und Flexibilität in Konflikt stehen. Im Gegensatz dazu kann ein ganzheitlicher oder ‚Rugby‘-Ansatz — mit dem ein Team als Einheit versucht Boden gut zu machen, indem der Ball hin- und hergespielt wird — besser heutige Wettbewerbsanforderungen erfüllen.”
(frei übersetzt)

Hiroataka Takeuchi und Ikujiro Nonaka, “The New New Product Development Game”, *Harvard Business Review*, Januar 1986.

SCRUM in 80 Worten

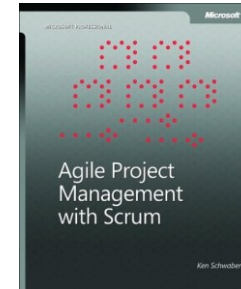
- Scrum ist ein agiler Prozess, der es erlaubt auf die Auslieferung der wichtigsten Geschäfts-Anforderungen innerhalb kürzester Zeit zu fokussieren.
- Scrum gestattet es schnell und in regelmäßigen Abschnitten (von zwei Wochen bis zu einem Monat) tatsächlich lauffähige Software zu inspizieren.
- Das Business setzt die Prioritäten. Selbst-organisierende Entwicklungsteams legen das beste Vorgehen zur Auslieferung der höchstpriorären Features fest.
- Alle zwei Wochen bis zu einem Monat kann jeder "lauffähige" Software sehen und entscheiden, diese so auszuliefern oder in einem weiteren Abschnitt zu ergänzen

Ursprung und Idee von SCRUM

Grundidee ist eine Entwicklung aus dem Bereich Maschinenbau. Verwandte Formen des agilen Vorgehens sind Kaizen und

Jeff Sutherland

Initiale Scrums bei Easel Corp., 1993
 IDX und über 500 Personen arbeiten mit Scrum

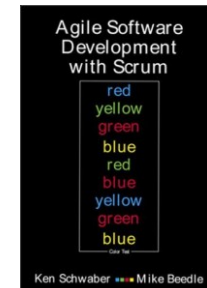


Ken Schwaber

- ADM
- Präsentiert Scrum auf der OOPSLA 96 mit Sutherland
- Autor von drei Büchern über Scrum

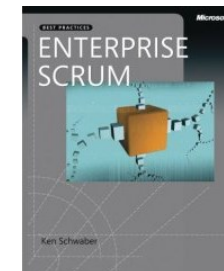
Mike Beedle

Scrum-Pattern in PLOPD4



Ken Schwaber und Mike Cohn

Scrum Alliance in 2002 gegründet; zuerst innerhalb der Agile Alliance



Quelle: TNS EMNID Kundenkarten-Studie 2010

Agiles Manifesto

Februar 2007

You Tube **Google**
 We are interested in better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

Individuals and interactions over processes

and tools

Working software over comprehensive



documentation



Customer collaboration over contract

negotiation

Responding to change over following a plan

That is, while there is value in the items on the right we value the items on the left more.



Quelle: Young & Rubicam, Brand Asset Valuator 2009

SCRUM ist schon bei folgenden Firmen im Einsatz

Quelle: Agile Allianz Stand 2010

Microsoft	Intuit
Yahoo	Nielsen Media
Google	First American Real Estate
Electronic Arts	BMC Software
High Moon Studios	Ipswitch
Lockheed Martin	John Deere
Philips	Lexis Nexis
Siemens	Sabre
Nokia	Salesforce.com
Capital One	Time Warner
BBC	Turner Broadcasting
Intuit	Oce
SAP	Allianz Deutschland
BMW	Mercedes Benz
PAYBACK	Deutsche Bank
ZF Friedrichshafen	Frankfurter Börse
Cisco	Metro Gruppe

... um das zu bauen ...

- Kommerzielle Software
- Inhouse-Entwicklungen
- Ausgesourcte Entwicklungen
- Festpreisprojekte
- Finanz-Applikationen
- ISO 9001-zertifizierte Applikationen
- Embedded systems
- 24x7 Systeme mit '99.999% uptime'-Anforderungen
- Den Joint Strike Fighter
- Videospiele
- 'FDA-approved', lebenskritische Systeme
- Satelliten-Kontrollsoftware
- Webseiten
- Handheld-Software
- Mobile Telefone
- 'Network switching'-Applikationen
- ISV Applikationen
- Einige der größten, in Anwendung befindlichen Applikationen

Charakteristisches zu SCRUM

Selbstorganisierende Team → Aufgaben werden im Team gelöst

Produkte werden in regelmäßigen Zeitabschnitten "ausgeliefert" → Sprints
 ... mit einer Dauer von 2 bis 4 Wochen

Anforderungen und Funktionen werden in Listenform visualisiert → Backlock
 ... Anforderungen dürfen sich ändern, aber nicht im Sprint

Keine Methoden festlegung, daher → Universell
 ... für jede Technologie anwendbar, fast egal für was

& generative Regeln um ein agiles Umfeld zu ermöglichen
 ... funktionsbezogen

... potenziell auslieferbar

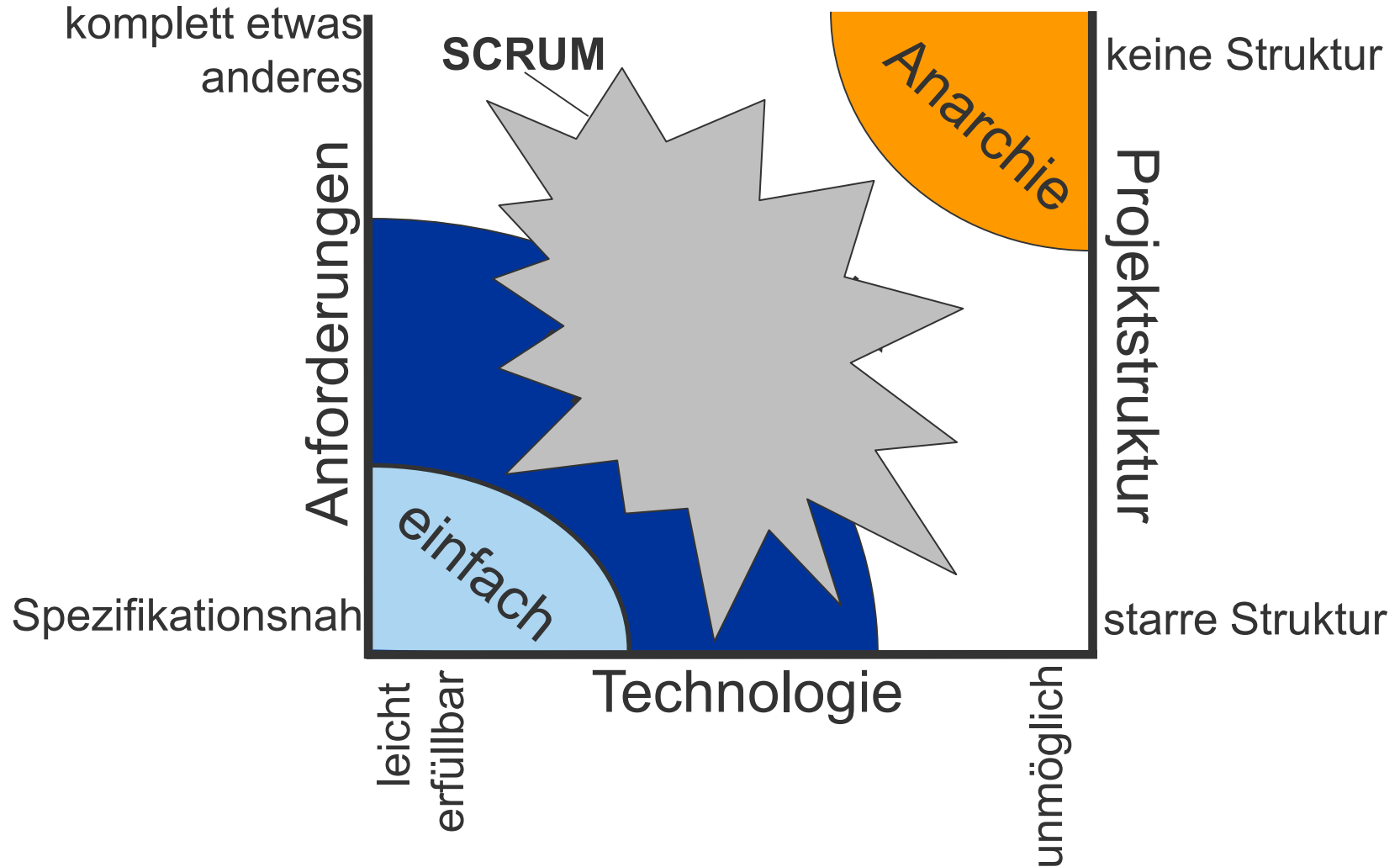
... getestet

& Agile Prozesse

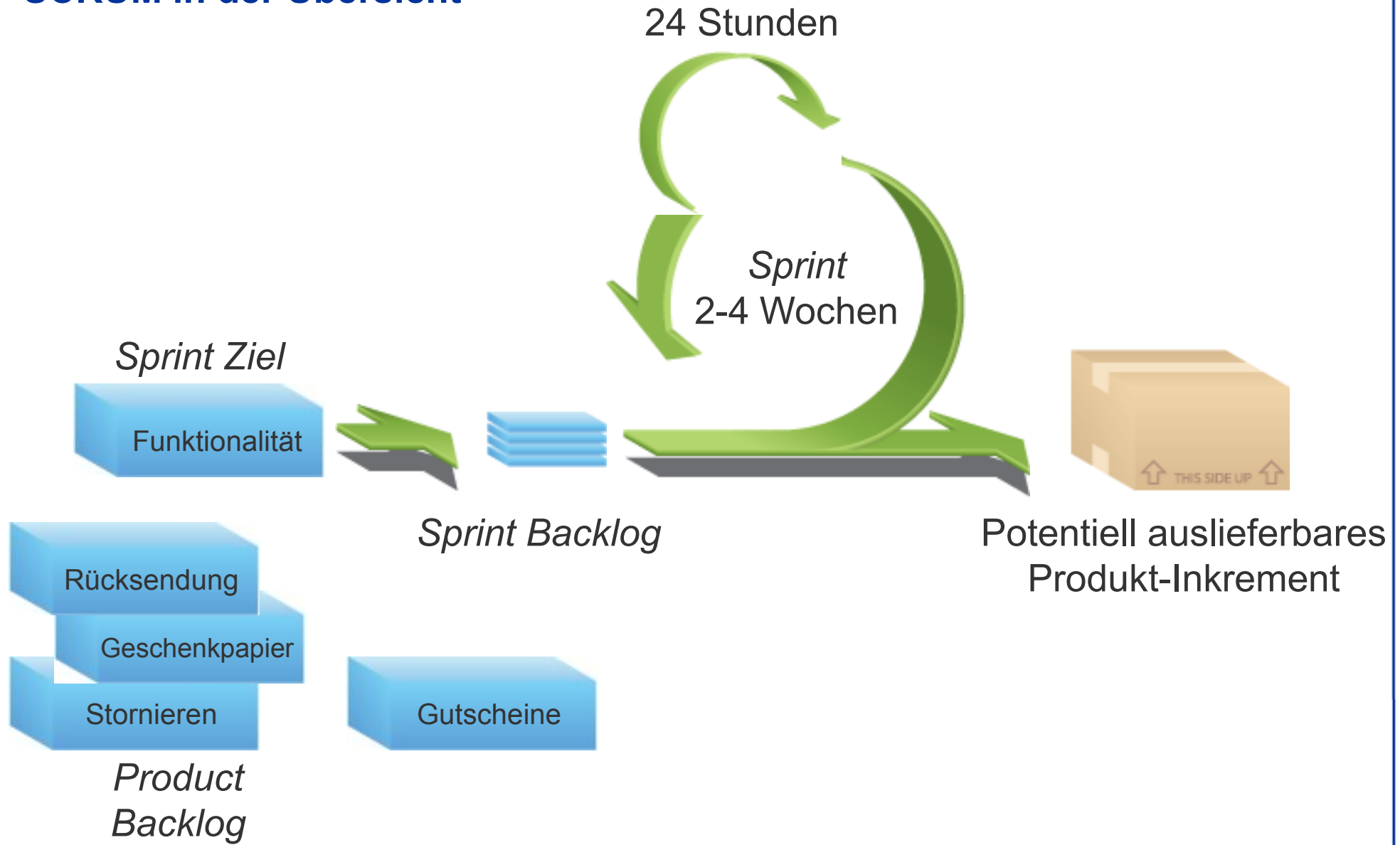
... das Team entscheidet wie

... der Kunde bestimmt was

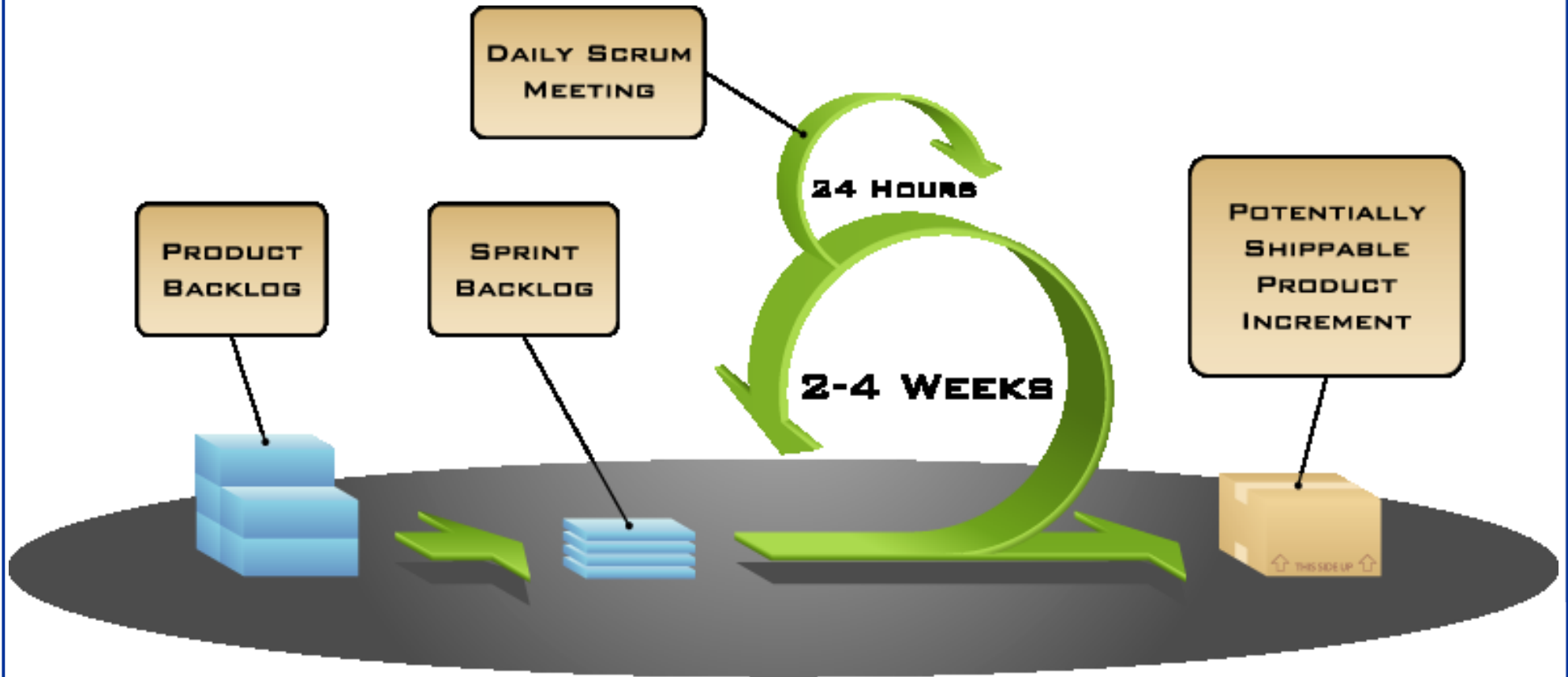
Spannungsbogen zwischen verschiedenen Projektmanagementmethoden



SCRUM in der Übersicht



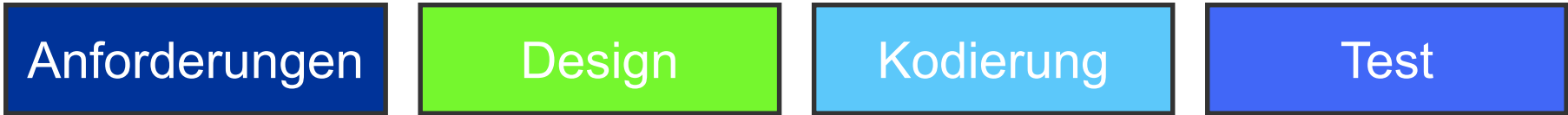
Im Detail



COPYRIGHT © 2006, MOUNTAIN GOAT SOFTWARE

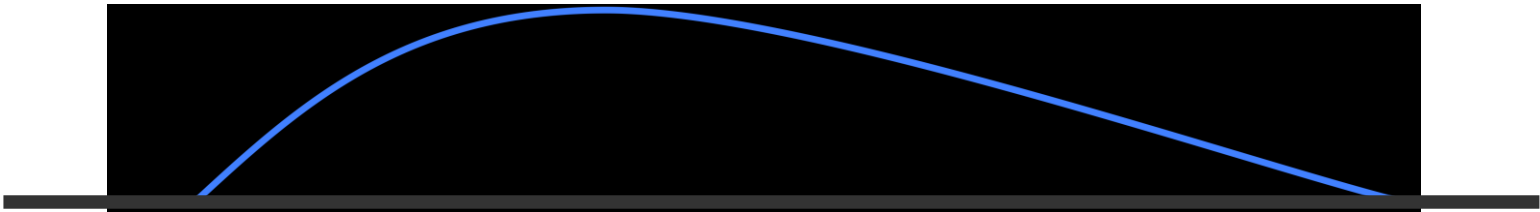
B Unterschiede zum Wasserfall

Vorgehensmodell



Anstatt alles im Ganzen hintereinander ...

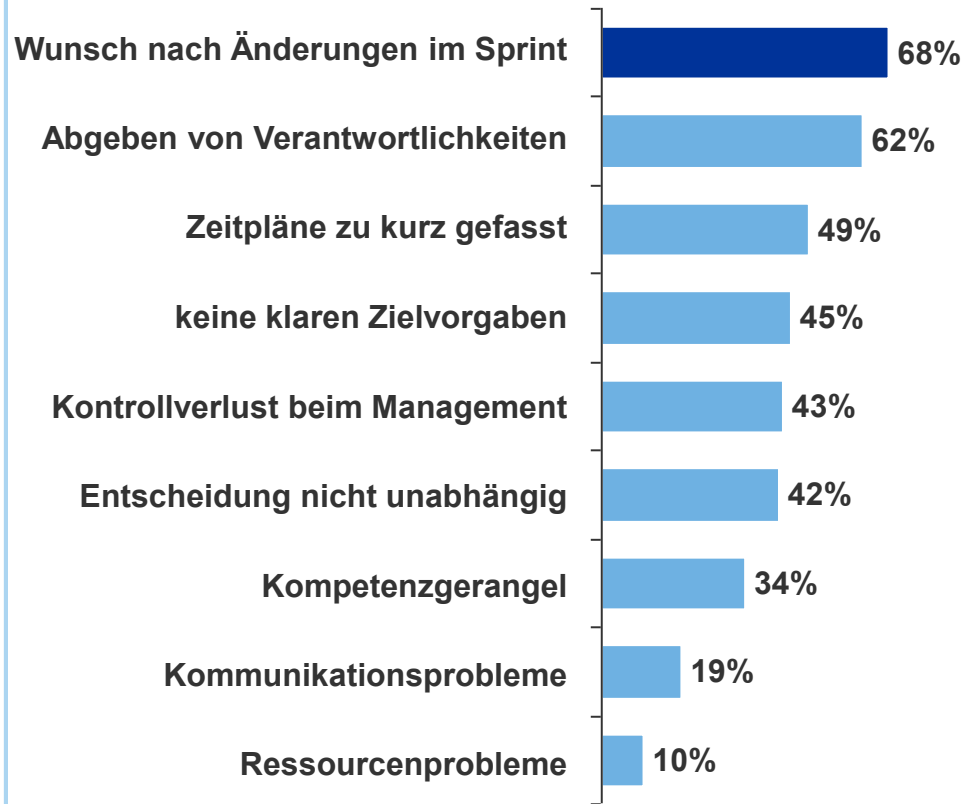
... tun Scrum-Teams ein bisschen von allem die ganze Zeit über



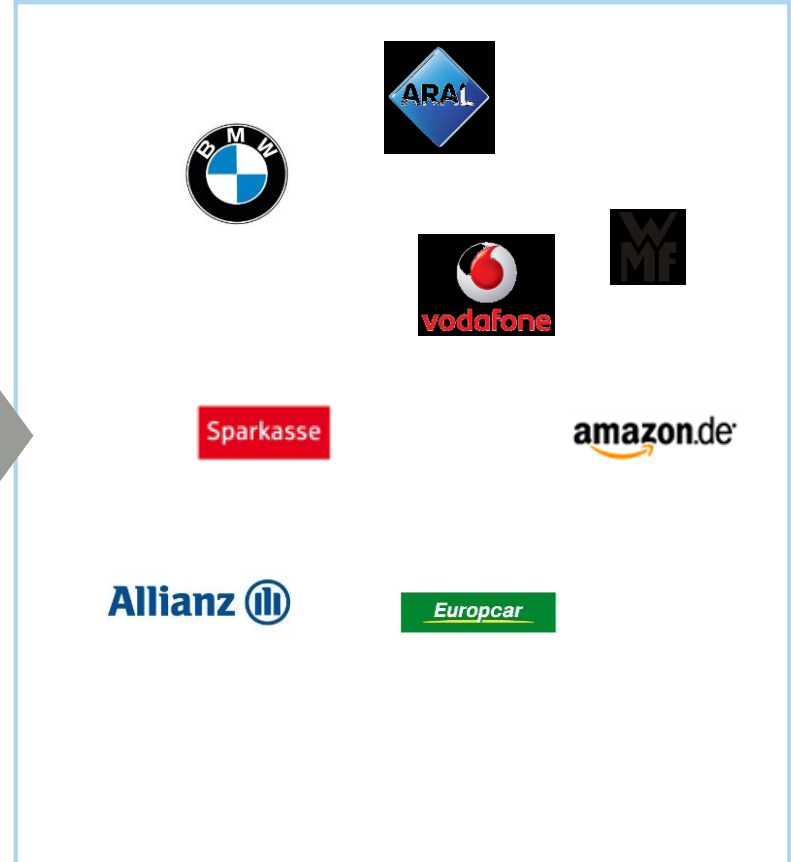
Quelle: "The New New Product Development Game" von Takeuchi und Nonaka. *Harvard Business Review*, January 1986.

Herausforderungen

Probleme die weiterhin bestehen¹



befragte Projektleiter von



1) Eigene Erhebung aus acht Projekten im Zeitraum von 2007 - 2011

Das Setup

Rollen

- Produkt-Owner
- ScrumMaster
- Team

Meetings

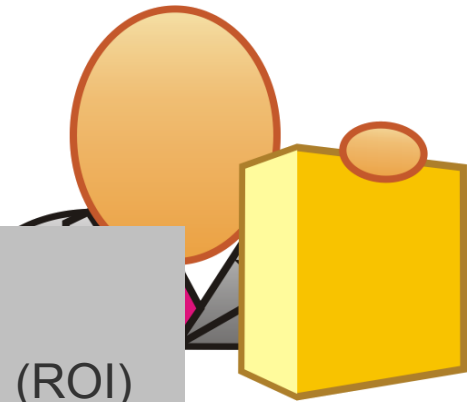
- Sprint-Planung
- Sprint-Review
- Sprint-Retrospektive
- Tägliches Scrum-

Artefakte

- Product Backlog
- Sprint Backlog
- Burndown-Diagramm

Der Product Owner

Definiert Produkt-Features
 Bestimmt Auslieferungsdatum und Inhalt
 Ist verantwortlich für das finanzielle Ergebnis des Projekts (ROI)
 Priorisiert Features abhängig vom Marktwert
 Passt Features und Prioritäten nach Bedarf für jeden *Sprint* an
 Akzeptiert oder weist Arbeitsergebnisse zurück



Der Scrum Master



Repräsentiert das Management gegenüber dem Projekt
 Verantwortlich für die Einhaltung von Scrum-Werten
 und -Techniken
 Beseitigt Hindernisse
 Stellt sicher, dass das Team vollständig funktional und
 produktiv ist
 Unterstützt die enge Zusammenarbeit zwischen allen
 Rollen und Funktionen
 Schützt das Team vor äußeren Störungen

Das Team

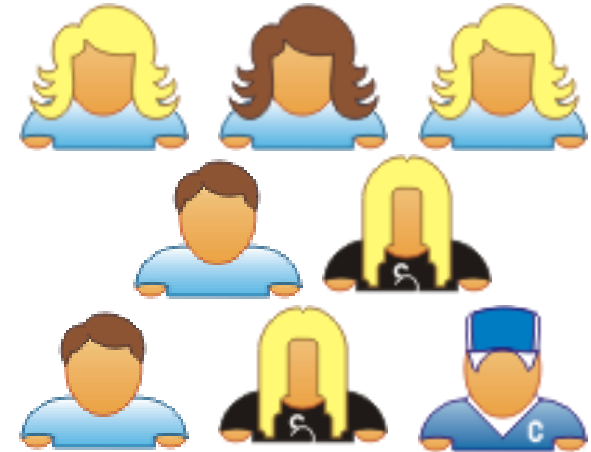
Typischerweise 5-9 Personen

Funktionsübergreifend:
 QS, Programmierer, UI-Designer, etc.

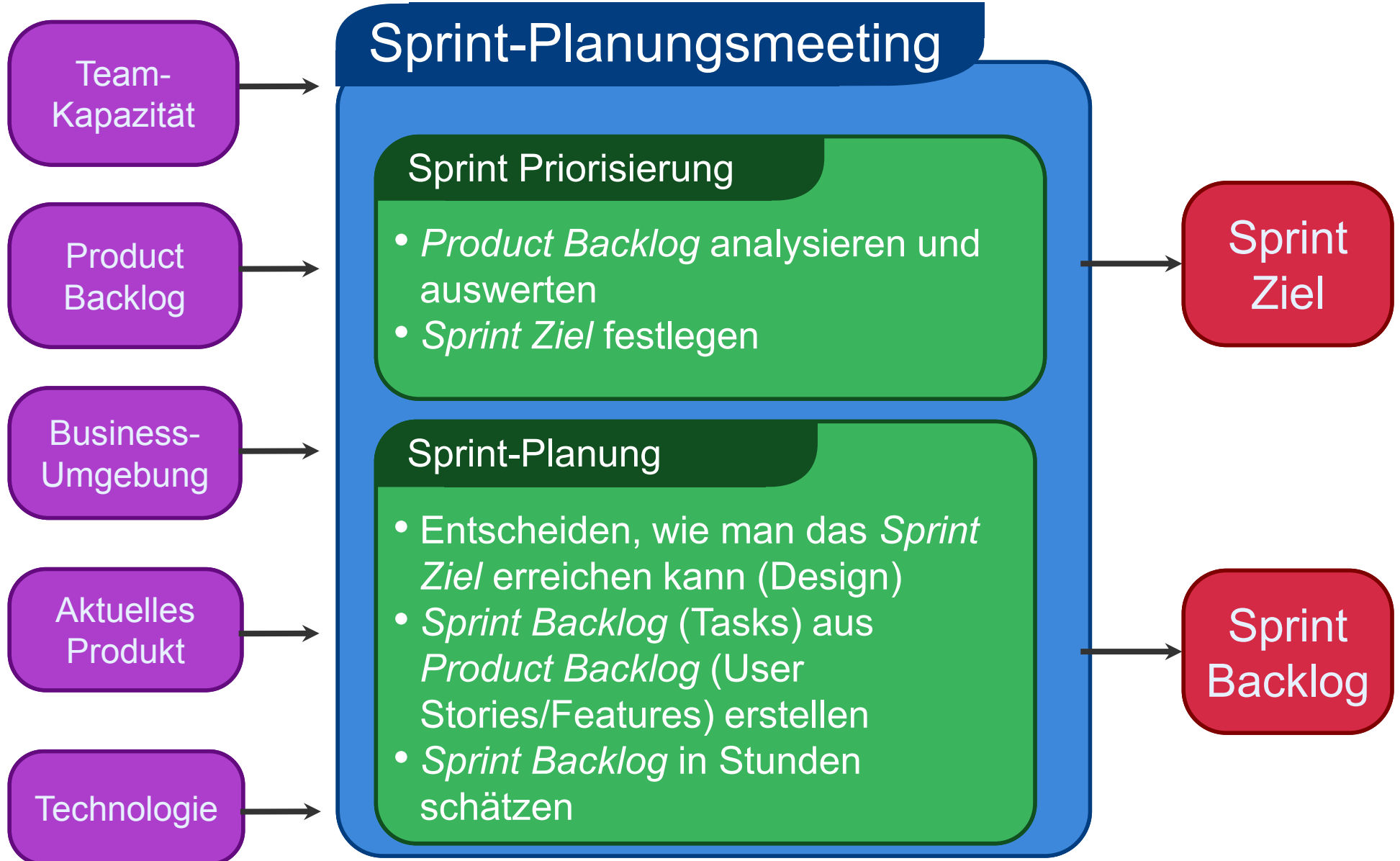
Mitglieder sollten Vollzeitmitglieder sein
 Wenige Ausnahmen (z.B. Systemadministratoren)

Teams organisieren sich selbst
 Ideal: keine Titel (aber manchmal nicht vermeidbar)

Mitgliedschaft kann sich nur zwischen *Sprints* verändern



Die Meetings



Das Sprint Planning

Team wählt Einheiten, zu deren Implementierung es sich verpflichten kann, aus dem *Product Backlog* aus

Sprint Backlog wird erstellt

Tasks werden identifiziert und geschätzt (1-16 Stunden)

Dieses wird gemeinschaftlich getan, nicht vom ScrumMaster allein

Highlevel-Design wird berücksichtigt

As a customer I want to find the next available store on a easy to use map

Sprint Backlog und Tasks

Code the middle tier (8 hours)
 Code the user interface (4)
 Write test fixtures (4)
 Code the foo class (6)
 Update performance tests (4)

Daily Meetings

Daily Scrum

Parameter:

Täglich
15 Minuten lang

Stand-up

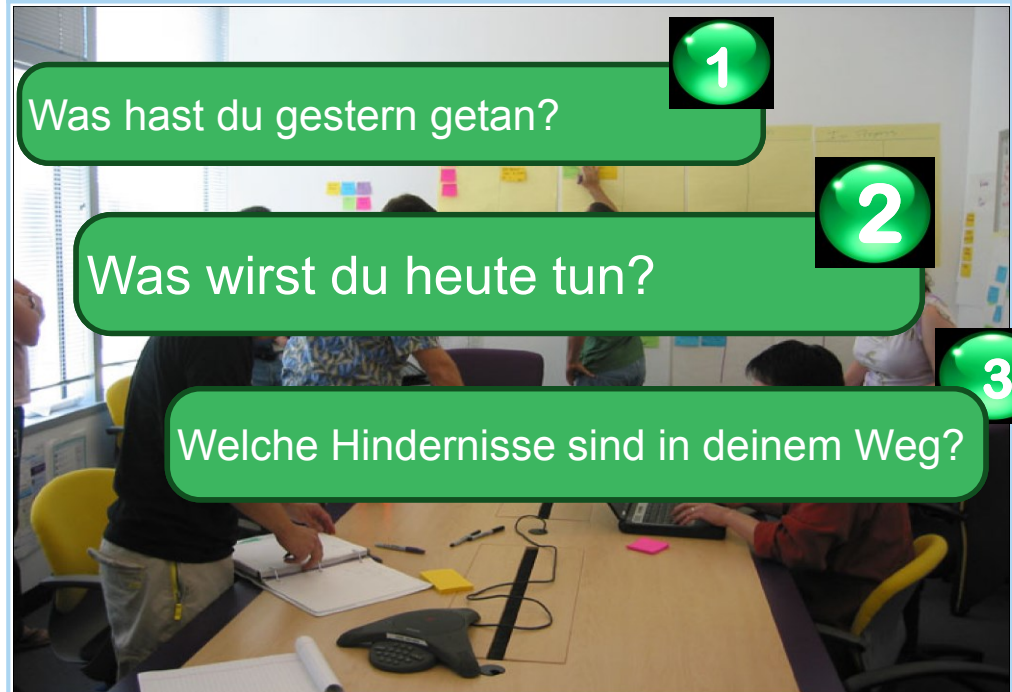
Nicht zur Problemlösung

Alle sind eingeladen

Aber nur Team-Mitglieder, der ScrumMaster, und der Produkt-Owner dürfen reden

Hilft, andere/überflüssige Meetings zu vermeiden

3 wichtige Fragen zur Absprache des Teams



Diese sind kein Statusberichte für den ScrumMaster, sondern Verpflichtungen in Anwesenheit der Kollegen

Sprint Reviewmeeting

Das Team präsentiert, was es während eines *Sprints* erreicht hat

Typischerweise in Form einer Demo der neuen Features oder der zugrunde liegenden Architektur

Informell

„Zwei Stunden zur Vorbereitung“-Regel

Keine Folien

Das ganze Team nimmt teil

Laden Sie die ganze Welt ein!



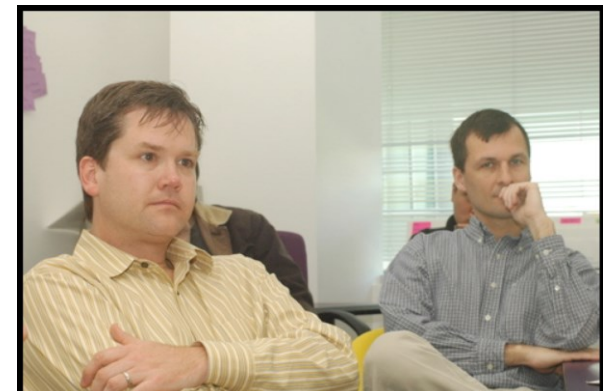
Sprint Retrospektive

Prüfen Sie regelmäßig, was gut und nicht so gut funktioniert

Typischerweise 15–30 Minuten lang
Nach jedem *Sprint*

Das ganze Team nimmt teil

ScrumMaster
Produkt-Owner
Team



Vielleicht Endkunden und andere Personen (aber Vorsicht!)

Das Produkt Backlog

- Die Anforderungen
- Eine Liste aller gewünschten Projektarbeiten
- Idealerweise soll jeder Eintrag wertvoll für Benutzer des Produktes oder Kunden sein

Vom Produkt-Owner priorisiert

Zu Beginn jedes *Sprints* re-priorisiert



Product Backlog

Backlog item	Estimate
Allow a guest to make a reservation	3
As a guest, I want to cancel a reservation.	5
As a guest, I want to change the dates of a reservation.	3
As a hotel employee, I can run RevPAR reports (revenue-per-available-room)	8
Improve exception handling	8
...	30
...	50

Die Sprint Ziele

- Kurze Angabe dessen, worauf sich die Arbeiten während des *Sprints* fokussieren

Database Application

Make the application run on SQL Server in addition to Oracle.

Life Sciences

Support features necessary for population genetics studies.

Financial Services

Support more technical indicators than company ABC with real-time, streaming data.

Management des Sprint Backlogs

Team-Mitglieder wählen Tasks aus (Arbeit wird nie zugewiesen)

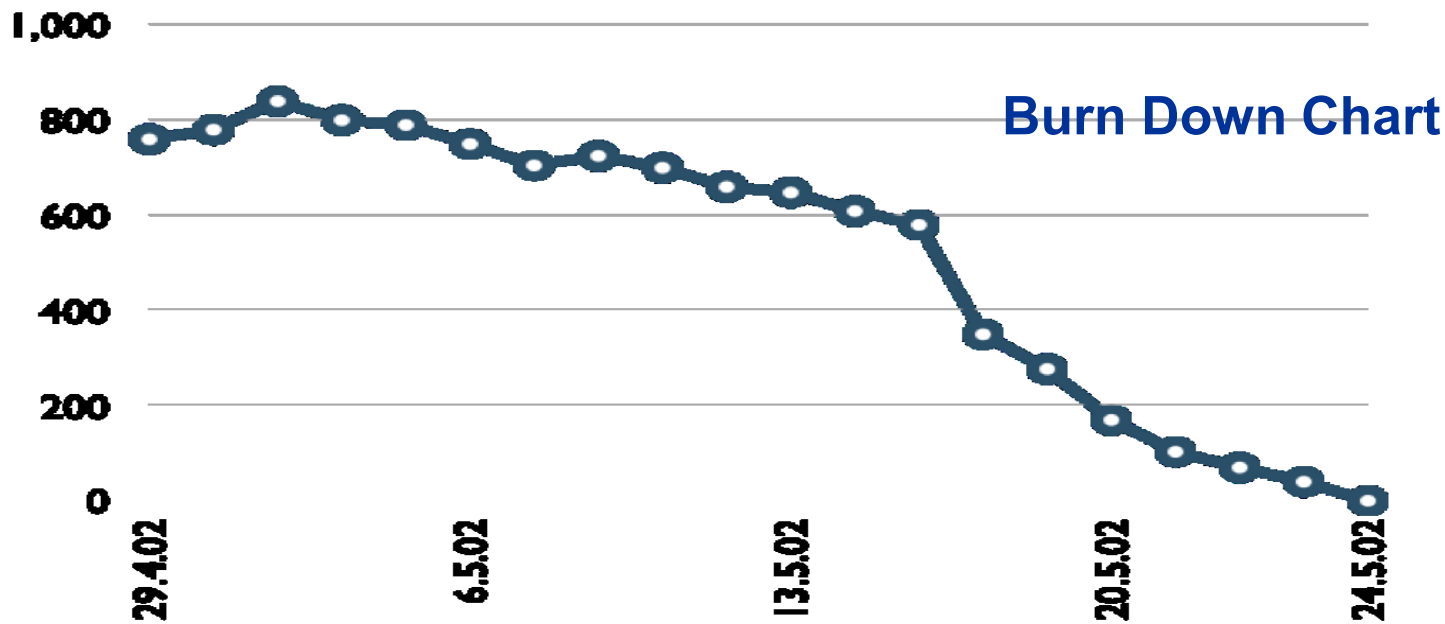
Die geschätzte restliche Arbeit wird täglich aktualisiert

Jedes Team-Mitglied kann Tasks hinzufügen, löschen oder ändern

Neue, für den *Sprint* benötigte Arbeit taucht auf

Wenn Arbeit unklar ist, definieren Sie eine Task mit einer größeren Zeitschätzung und brechen diese später herunter

Updates Sie verbleibende Arbeit sobald Sie mehr wissen



Skalierbarkeit

Typische Teams bestehen aus 7 ± 2 Personen
 Teams von Teams ermöglichen Skalierbarkeit

Faktoren des Skalierens

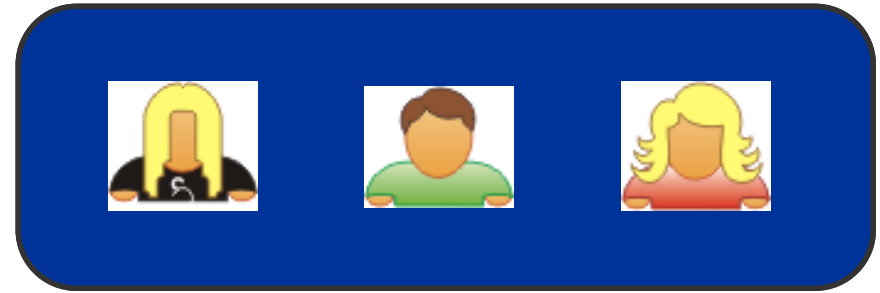
Typ der Anwendung

Teamgröße

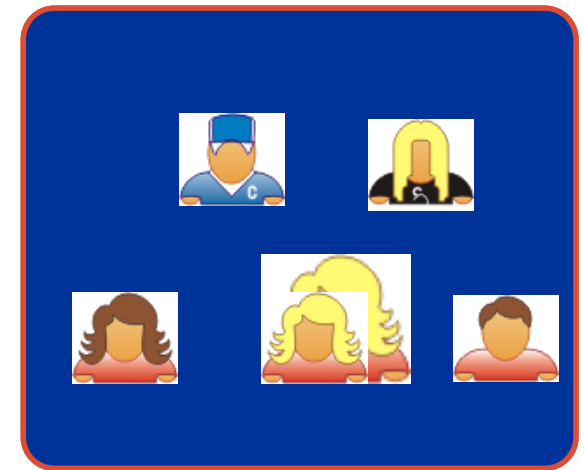
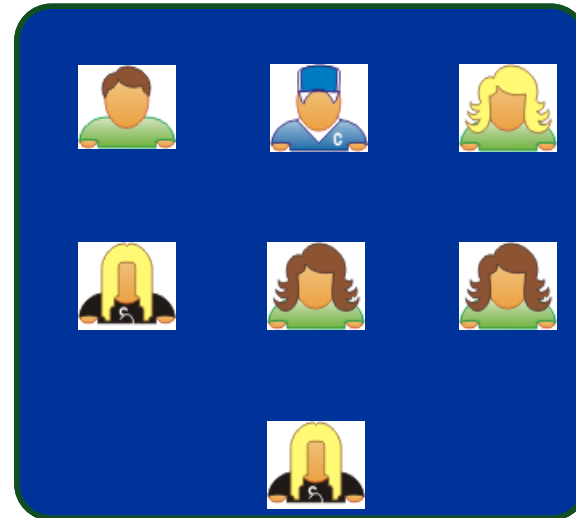
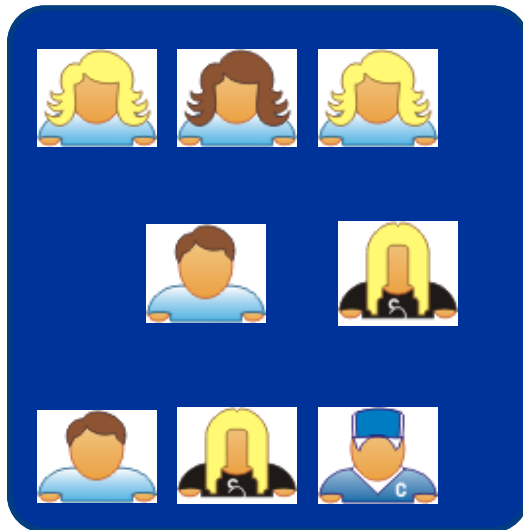
Teamverteilung (örtlich)

Projektdauer

Scrum ist mehrmals für 500-Personenprojekte verwendet worden



Scrum of Scrums

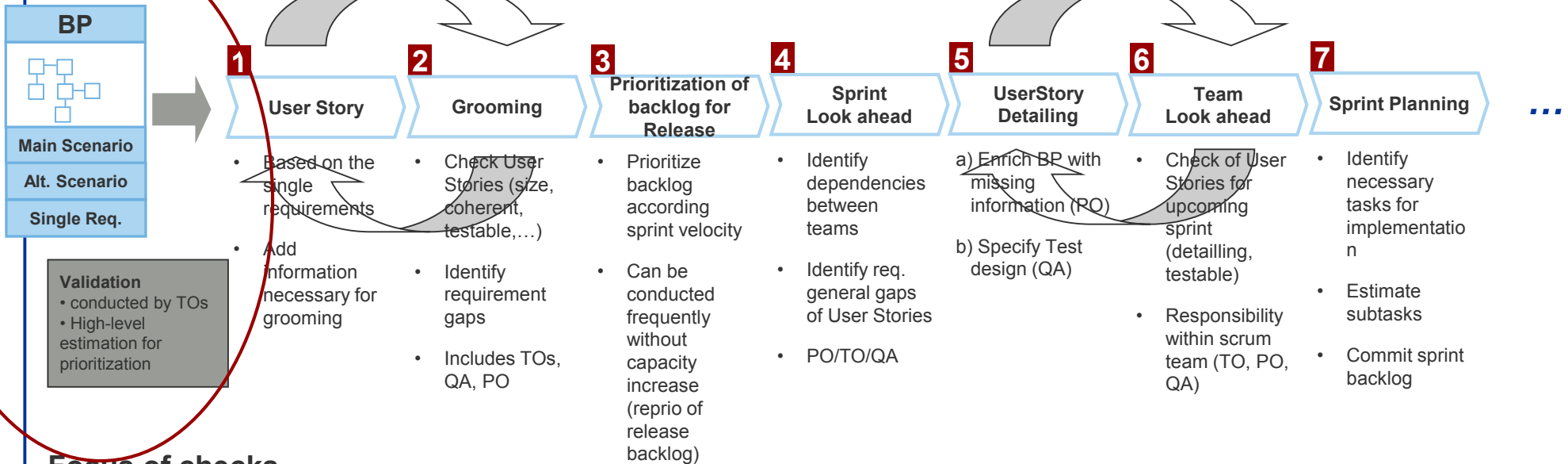


C Praxisbeispiele und Planning Poker ...

What happens before User Stories are written?

Release preparation

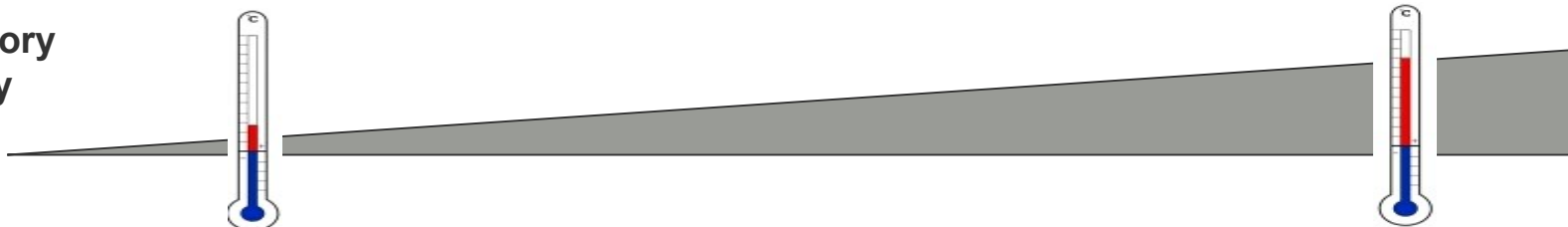
Sprint Preparation



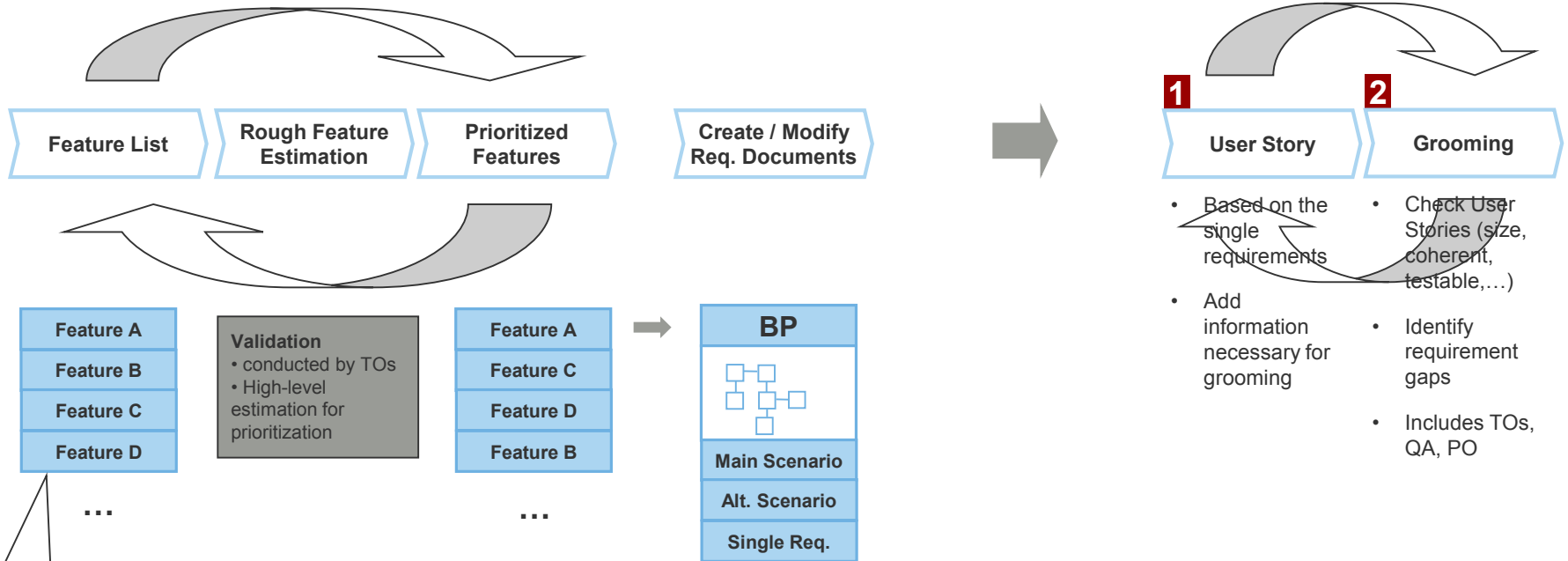
Focus of checks

<ul style="list-style-type: none"> Consistency Feasibility 		<ul style="list-style-type: none"> Appropriated size Testability Coherent 		<ul style="list-style-type: none"> Completeness Test Design 	
--	--	--	--	---	--

User Story Maturity

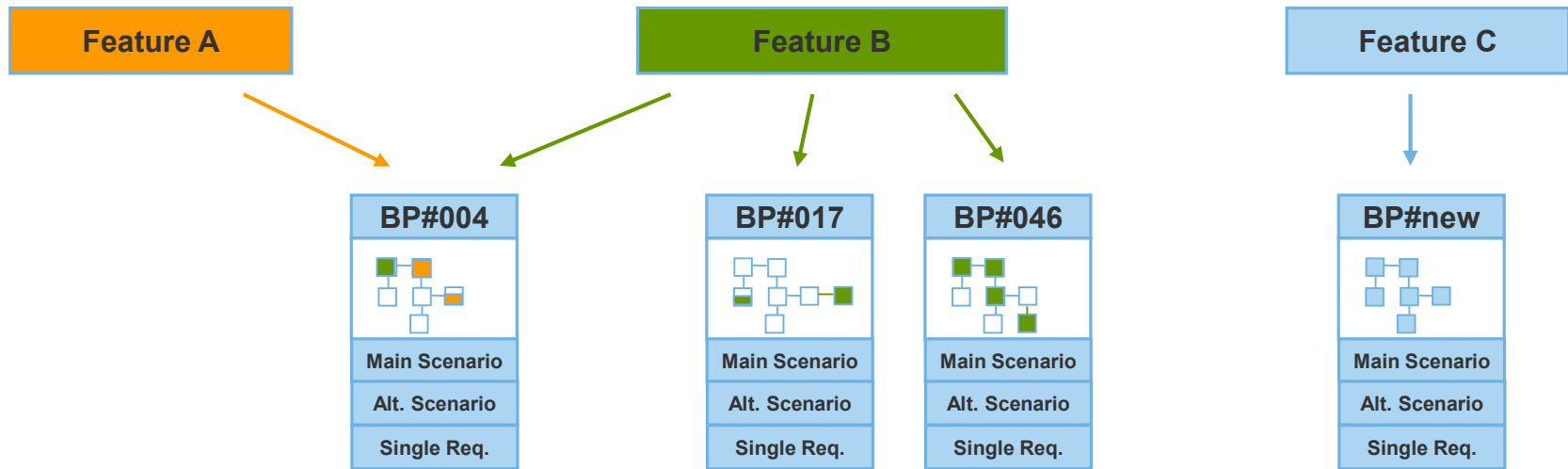
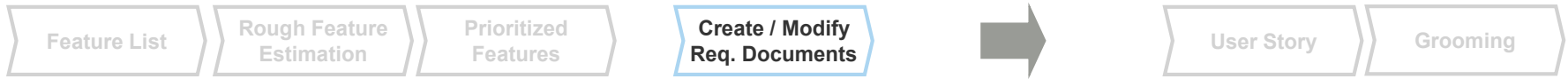


Prioritized Feature Lists drive the modification of req. documents

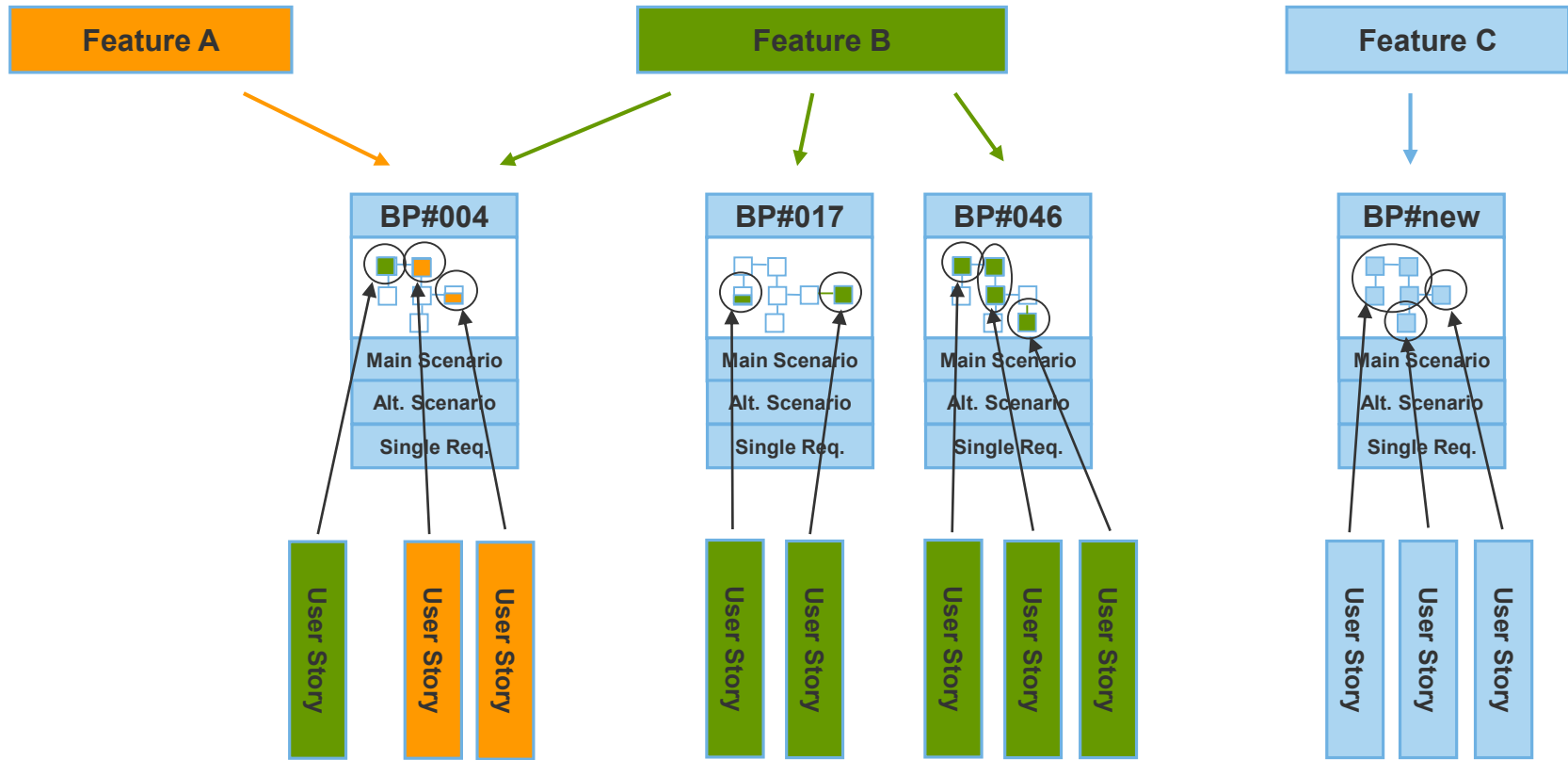
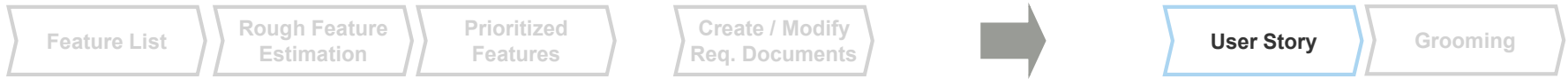


Feasibility of Features is pre-checked by Architects in Product Gateway (see Feature Lifecycle Process / Markus Kleinfelder)

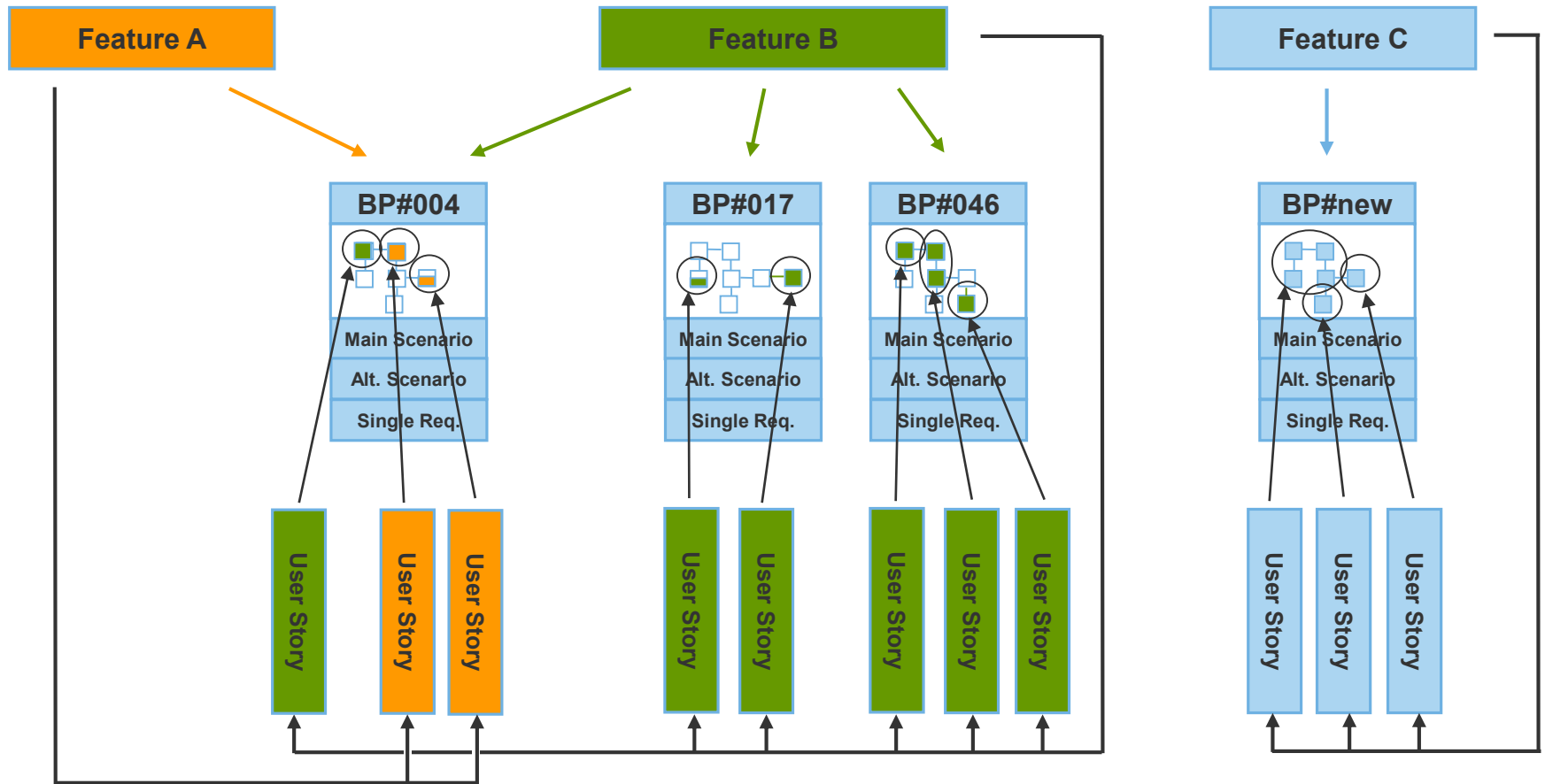
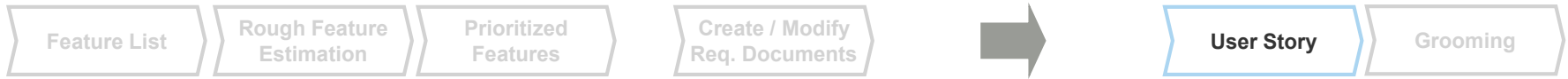
Features have impact on existing and / or new requirements documents



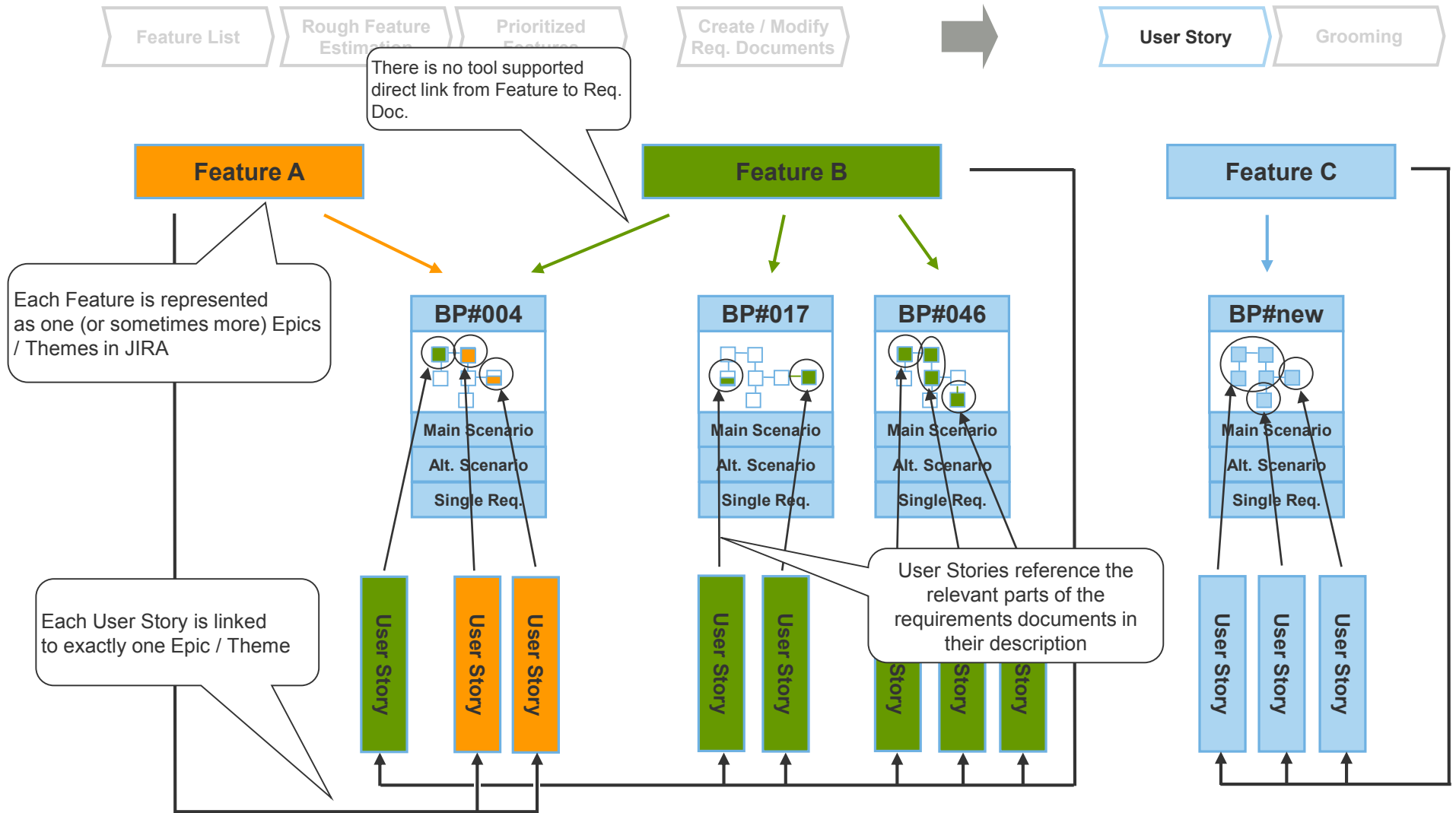
User Stories are created to communicate business requirements to the scrum teams



Each User Story belongs to exactly one Feature



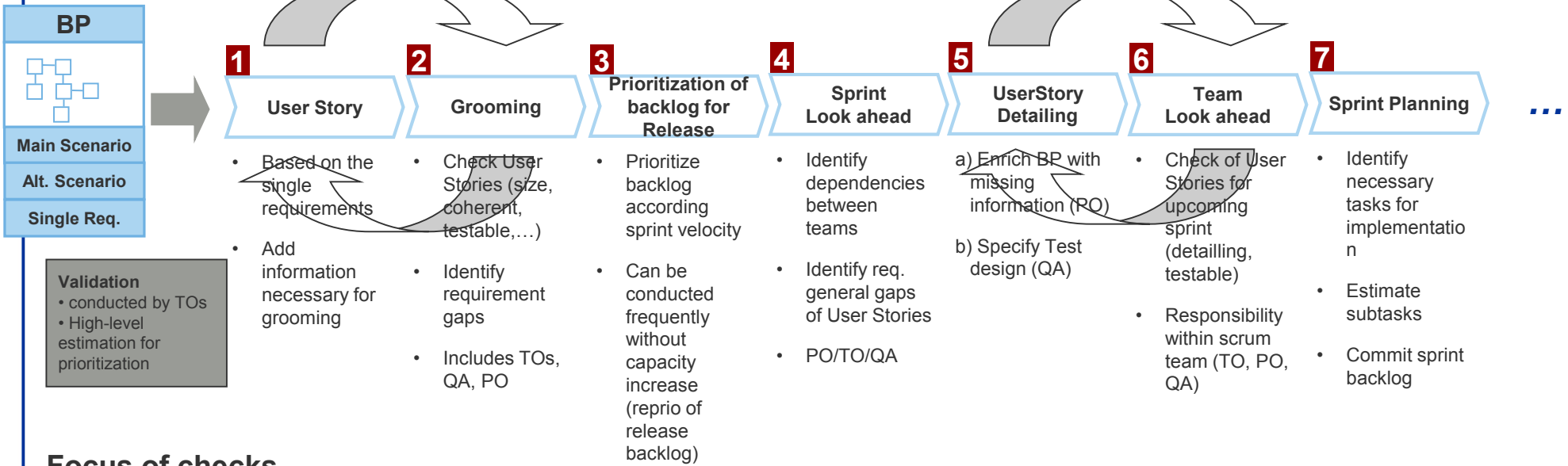
Links between Features, Requirements Documents and User Stories



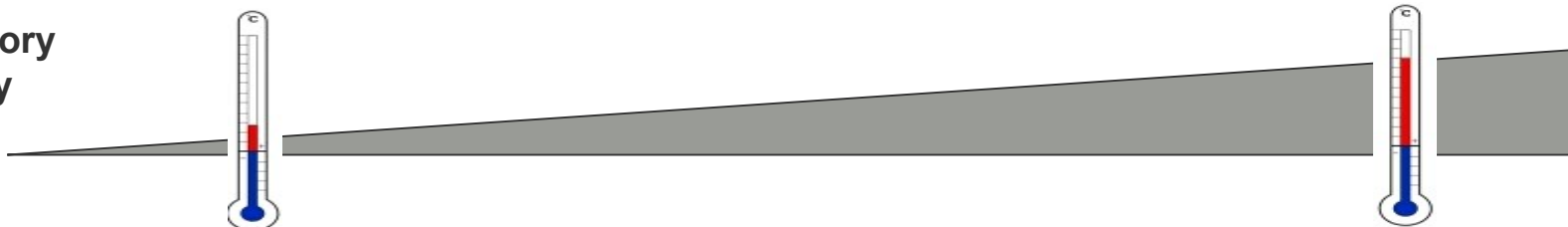
Requirement Implementation Process

Release preparation

Sprint Preparation



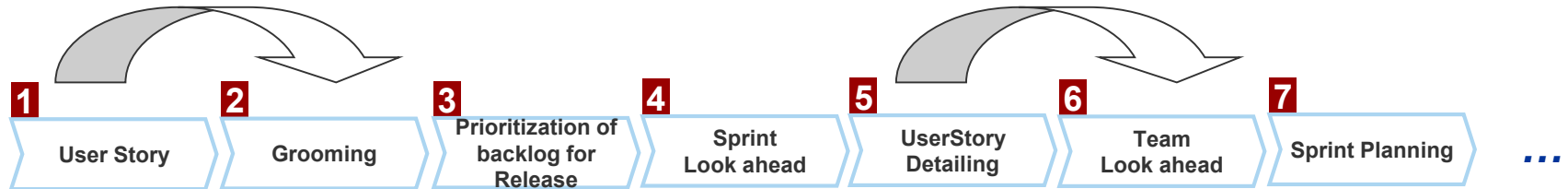
User Story Maturity



Requirement Implementation Process – Responsibilities & User Story Lifecycle

Release preparation

Sprint Preparation



Responsibilities

- 1 User Story**
 - PO creates User Story
 - Corresponding TOs, QAs supports to check User Stories
 - N.Teams - >TOs of Customer
 - S. Teams -> Scrum Master/Tech.L ead
- 2 Grooming**
 - POs of the to be groomed US
 - Corrsponding TOs and Key TOs
 - Corrsponding QA
- 3 Prioritization of backlog for Release**
 - CPO, CTO, LSA, selected POs and TOs
- 4 Sprint Look ahead**
 - CPO, TOs, Pos, QA
- 5 UserStory Detailing**
 - PO, TO and/or selected ressources of team, QA Owner or QA ressource of team
- 6 Team Look ahead**
 - PO, TO and/or selected ressources of team, QA Owner or QA ressource of team
- 7 Sprint Planning**
 - Scrum Team including PO, TO, QA

Jira User Story Life Cycle



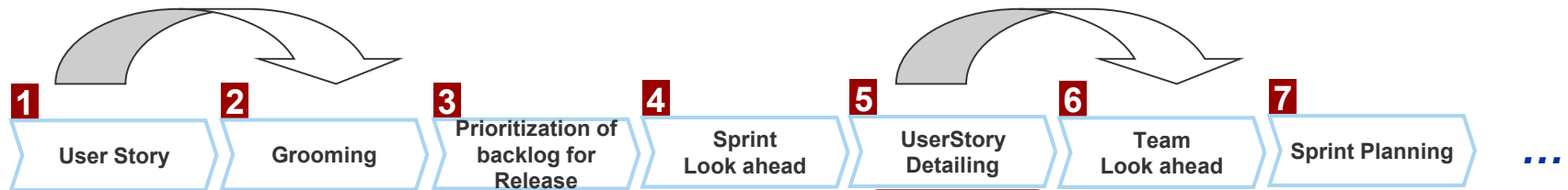
<p>Create User Stories for Features (referring to req. docs)</p> <p>Incorporate feedback regarding grooming readiness into US</p>	<p>Present US in Grooming</p>	<p>Maintain backlog (prioritization)</p>	<p>Add missing details to req docs and US</p> <p>Incorporate feedback regarding level of detail into US</p>	<p>Explain new US to scrum team in advance (if necessary)</p> <p>Review test design</p>	<p>Clarify questions during sprint</p> <p>Present US in sprint planning</p>
---	-------------------------------	--	---	---	---

PO Responsibilities

Direct communication of PO with scrum team (or representatives) is very important

Release preparation

Sprint Preparation



Responsibilities

- PO creates User Story
- Corresponding TOs, QAs supports to check User Stories
- N Teams -> TOs of Project
- S. Teams -> Scrum Master/Tech.L ead

Discuss User Story "slicing" per Feature in time before Grooming, so feedback can be considered

- POs of the to be groomed US
- Corrsponding TOs and Key TOs
- Corrsponding QA

- CPO, CTO, LSA, select POs and

Discuss level of detail of US. Discuss if design is prepared.

Discuss test design

PO, TO and/or selected rersources of team, QA Owner or QA rersource of team

- PO, TO and/or selected rersources of team, QA Owner or QA rersource of team

Scrum Team including PO, TO, QA

Answer questions during sprint

Jira User Story Life Cycle



PO Responsibilities

Create User Stories for Features (referring to req. docs)

Incorporate feedback regarding grooming readiness into US

Present US in Grooming

Maintain backlog (prioritization)

Add missing details to req docs and US

Incorporate feedback regarding level of detail into US

Explain new US to scrum team in advance (if necessary)

Review test design

Clarify questions during sprint

Present US in sprint planning

What is a User Story?

What criteria should a User Story fulfill?

- Goals of a User Story
 - Means of communication from PO to scrum team:
Which business requirements are expected to be developed and tested?
 - Unit of planning for scrum team:
Which team should implement the story in what sprint?
 - Unit of progress tracking:
What (parts of) business requirements have been successfully implemented?

- Most important criteria of a User Story
 - Adequately sized, so it can be implemented in one Sprint
 - Testable on its own (not dependent on other stories)

- *Note: What is a Story not?*
 - *A means for structuring Integration Tests or User Acceptance Tests on the level of business processes*

This can be difficult. PO needs to discuss and decide User Story size and scope together with scrum team (development and QA).

What should the content be of a (Global Core) User Story?

1. Summary (“As a <type of user>, I want <some goal> so that <some reason>.”)
2. Scope of the Story
 - Short description of business goal and context (ideally not more than 2-3 sentences)
 - optionally explicit scope boundaries (what is not scope of the story)
3. Requirements Details
 - Using references into the business requirements documents (avoid redundancy, no copy&paste of the original requirements)
 - Precisely defining scope (exactly what parts of the referenced requirements documents are in scope of this story?)
 - *Important: include working ("clickable") links to the requirements documents (see following [slide](#) about subversion on how to do this)*
4. Testideas and –hints
 - aspects that are especially important to test
5. Additional and technical notes relevant for understanding and estimating the story, e.g.
 - assumptions about the technical solution
 - references to already implemented stories that have similarities to this one
 - any other important results of discussions between PO, Dev, QA

Entered by PO when creating the story.

Entered by PO before QA starts test design

Entered as a result of discussions between PO, Dev, QA

How to fill out the JIRA User Story fields and for a template for the story description, see <http://display/PBINT/PO-Guideline+for+handling+User+Stories>

How to "slice" a Feature into User Stories?

- User Stories should, whenever possible, be "sliced" according to business content
 - not by technical aspects (e.g. architectural layers)
- Remember: a User Story should be testable on its own
 - i.e. the PO needs to be able to decide after a sprint, if the story was successfully implemented
 - this is easier if the scope of the story is defined in terms of business requirements, not technologies
 - (even though such "small" stories might feel strange at the beginning for POs new to scrum)
- Some ideas for starting the User Story "slicing"
 - Happy Case as one Story, exceptional scenarios as additional stories
 - Basic process with only the absolutely necessary steps as one story ("Durchstich"), the more complicated aspects (e.g. sophisticated validations, special cases, ...) as additional stories"
 - Reduced functionality with assumptions about default values as the first story, further data entry in additional stories
- PO should make the first draft of the User Story slicing
 - But PO needs to discuss and get feedback by Dev and QA before taking the stories into grooming

Quellenverzeichnis:

Auszüge aus der Präsentation von:
Mike Cohn

mike@mountaingoatsoftware.com

www.mountaingoatsoftware.com

(720) 890-6110 (office)

Teile dieser Präsentation wurden (aus der deutschen Version von) "An Introduction to Scrum" von Mike Cohn, übersetzt von Simon Roberts und Birgit Panzram entnommen

Takeuchi, Hirotaka; Nonaka, Ikujiro (January–February 1986).

"The New New Product Development Game" (PDF). Harvard Business Review. Retrieved 2010-06-09.

Schwaber, Ken; Beedle, Mike (2002).

Agile software development with Scrum. Prentice Hall. ISBN 0130676349.

Scrum, Scrum Developer Courses, Scrum Knowledge Assessment, Scrum Guide, Ken Schwaber - Scrum Guides". Scrum.org. 2009.

<http://www.scrum.org/scrumguides/>

Linda Rising, Norman S. Janoff; IEEE SOFTWARE

Ausgabe: J u l y / A u g u s t 2000 <http://members.cox.net/risingl1/Articles/IEEEScrum.pdf>

- *Agile and Iterative Development: A Manager's Guide* von Craig Larman
- *Agile Estimating and Planning* von Mike Cohn
- *Agiles Projektmanagement mit Scrum* von Ken Schwaber
- *Scrum - Agiles Projektmanagement erfolgreich einsetzen* von Roman Pichler
- *Agile Retrospectives* von Esther Derby und Diana Larsen
- *Agile Software Development Ecosystems* von Jim Highsmith
- *Agile Software Development with Scrum* von Ken Schwaber und Mike Beedle
- *The Enterprise and Scrum* von Ken Schwaber
- *User Stories Applied for Agile Software Development* von Mike Cohn
- Artikel auf www.scrumalliance.org