Schwergewichtige Softwareentwicklungsprozesse

- (z.B. Wasserfallmodell, V-Modell)
- Die auf der vorigen Folie gezeigten Schritte werden nacheinander ausgeführt:
  - Anforderungen möglichst vollständig erfassen
  - danach Design vollständig entwerfen
  - danach mit der Realisierung beginnen
- ausführliche Dokumentation

Leichtgewichtige Softwareentwicklungsprozesse

- (z.B. Scrum, eXtreme Programming)
- Software änderbar gestalten
- Der Code gehört allen
- Direkte Kommunikation in kleinen Arbeitsgruppen
- Häufige Einbeziehung des Kunden während des gesamten Projekts
- Es wird schnell eine einfache, funktionierende Software gebaut, die dann weiterentwickelt wird

Leichtgewichtige Softwareentwicklungsprozesse

- Es wird frühzeitig und häufig lauffähige Software ausgeliefert.
- Spätere Anforderungen sind willkommen.
- Die Features mit dem höchsten Wert für den Kunden werden zuerst realisiert.
- Unnötige Dokumentation wird vermieden, der Code sollte sich selbst dokumentieren.

Leichtgewichtige Softwareentwicklungsprozesse

- Automatische Tests begleitend zur Entwicklung
- Vermeiden von Redundanz:
  Enthält das System bereits Code, der ähnliche Aufgaben erfüllt wie die aktuell zu realisierende, muss dieser Code so umgestaltet werden, dass er sowohl die "alte" als auch die "neue" Aufgabe erfüllt.
- Es wird täglich eine lauffähige Software gebaut (Daily Build)

Agiles Manifest

Individuen und Interaktionen	sind wichtiger als	Prozesse und Tools
Funktionierende Software	ist wichtiger als	Umfangreiche Dokumentation
Kooperation mit Projektbetroffenen	ist wichtiger als	Vertragsverhandlungen
Reaktion auf Änderungen	ist wichtiger als	Verfolgung eines festgelegten Plans

Quelle: www.agilemanifesto.org

SCRUM-KURZEINFÜHRUNG

Arbeitstechniken für das Management von Softwareprojekten

Scrum: Grundideen

- **Erwarte Überraschungen…**
  Änderungen an den Anforderungen sind ebenso normal wie noch zu präzisierende Anforderungen.

- **… aber vermeide böse Überraschungen!**
  Regelmäßige Überprüfung des Geleisteten (in den Scrum Meetings)

- "In einem neuen Softwaresystem werden die Anforderungen solange nicht komplett bekannt sein, bis die Anwender damit arbeiten." (Humphrey, 1995)

Wir verlieren den Staffellauf

"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)

Hirotaka Takeuchi und Ikujiro Nonaka, "The New New Product Development Game", *Harvard Business Review*, Januar 1986.

Scrum in 100 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 Geschäft setzt die Prioritäten. Selbst-organisierende Entwicklungsteams legen das beste Vorgehen zur Auslieferung der höchstprioren 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.

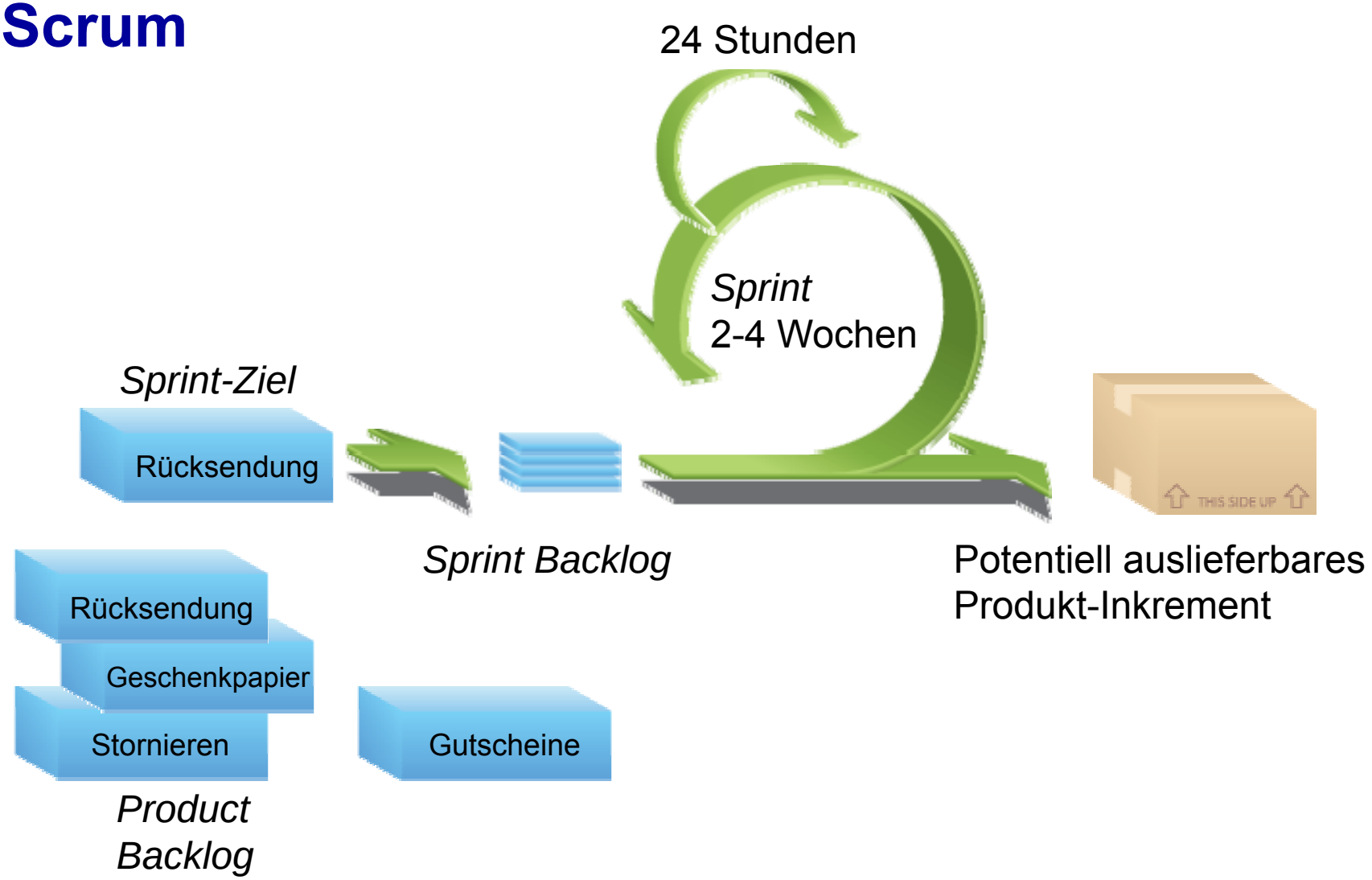
Scrum wurde benutzt bei…

- Microsoft
- Yahoo
- Google
- Electronic Arts
- High Moon Studios
- Lockheed Martin
- Philips
- Siemens
- Nokia
- Capital One
- BBC
- Intuit
- SAP
- Intuit
- Nielsen Media
- First American Real Estate
- BMC Software
- Ipswitch
- John Deere
- Lexis Nexis
- Sabre
- Salesforce.com
- Time Warner
- Turner Broadcasting
- Oce
- Allianz Deutschland

Charakteristika von Scrum

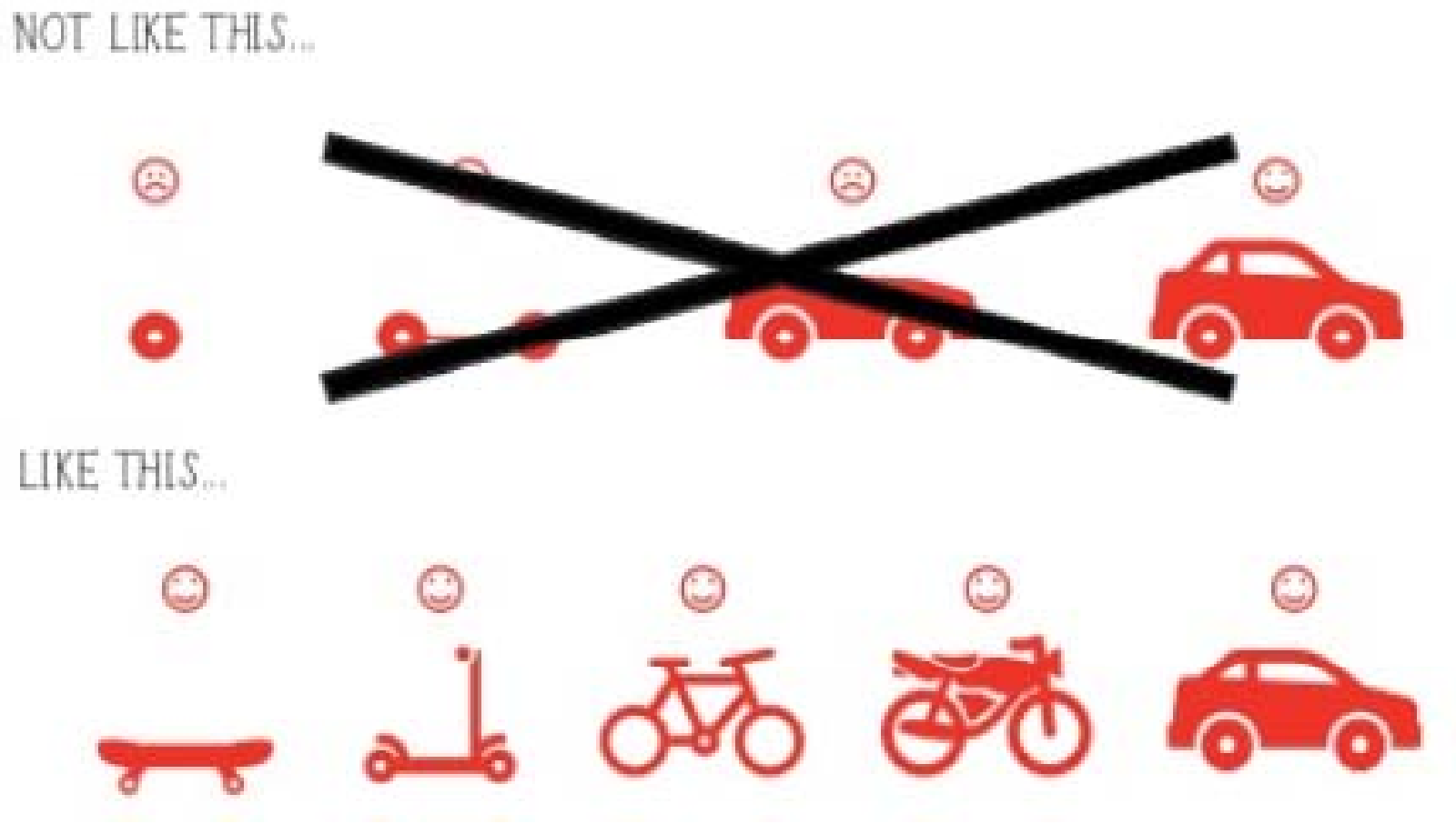
- Selbst-organisierende Teams
- Produkt schreitet in Serien / Abschnitten von monatlichen *Sprints* fort
- Anforderungen sind als Listeneinträge im *Product Backlog* festgehalten
- Die innerhalb des nächsten Sprints zu lösenden Aufgaben kommen ins *Sprint Backlog*.

Scrum


24 Stunden
Sprint
2-4 Wochen
Sprint-Ziel
Rücksendung
Sprint Backlog
Potentiell auslieferbares
Produkt-Inkrement
THIS SIDE UP
Rücksendung
Geschenkpapier
Stornieren
Gutscheine
Product
Backlog

Die Sprints

- Scrum-Projekte schreiten in Serien von Sprints voran
- Die typische Sprintdauer betragt 2 – 4 Wochen (für unser ZUUL-Projekt: 2 Wochen)
- Eine konstante Dauer führt zu einem besseren Rhythmus
- Das Produkt wird während des Sprints entworfen, kodiert und getestet
- Am Ende des Sprints steht immer lauffähige Software!
- Möglichst keine Änderungen an den Aufgaben während eines Sprints!

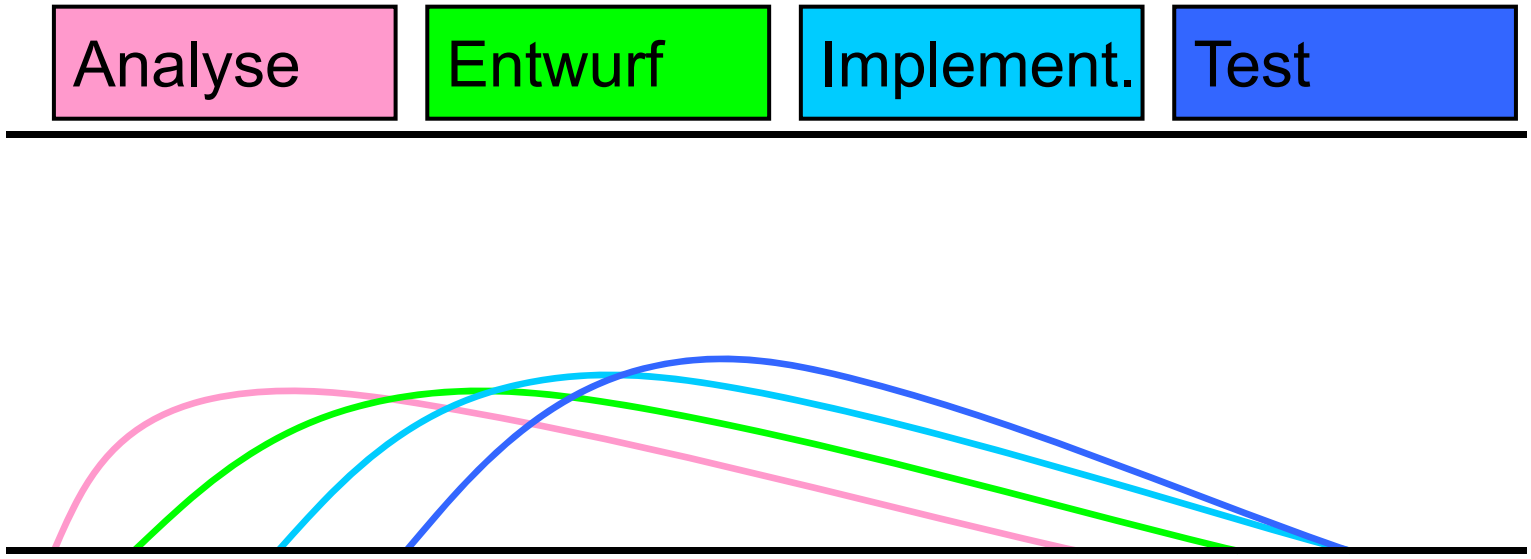

NOT LIKE THIS...
LIKE THIS...

Keine sequentielle Entwicklung

- Analyse
- Entwurf durch Architekten
- Implementierung und Tests
- Einführung
- Wartung

Anstatt alles im Ganzen hintereinander ...

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



Quelle: “The New New Product Development Game”, Hirotaka Takeuchi and Ikujiro Nonaka, *Harvard Business Review*, January 1986.

Scrum - der Rahmen

Rollen

- Produkt Owner
- Scrum Master
- Team

Meetings

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

Artefakte

- Product Backlog
- Sprint Backlog
- Burndown-Diagramm

Scrum - der Rahmen

Rollen

- Produkt Owner
- Scrum Master
- Team

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

Artefakte

- Product Backlog
- Sprint Backlog
- Burndown-Diagramm

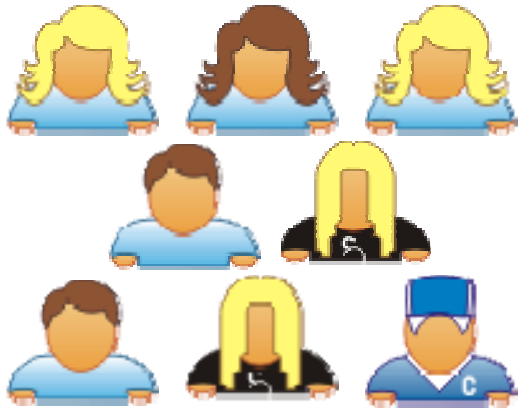
Der Product Owner

- Vertritt den Kunden
- Definiert Produkt-Features
- Bestimmt Auslieferungsdatum und Inhalt
- Ist verantwortlich für das finanzielle Ergebnis des Projekts
- Priorisiert Features
- Passt Features und Prioritäten nach Bedarf für jeden Sprint an
- Akzeptiert oder weist Arbeitsergebnisse zurück

Der Scrum Master

- Repräsentiert das Management gegenüber dem Projekt, ist aber kein Chef oder Projektleiter
- Verantwortlich für die Umsetzung der Scrum-Regeln (moderiert z.B. Meetings)
- Stellt sicher, dass das Team produktiv arbeiten kann
- Schützt das Team vor äußeren Störungen
- Unterstützt die enge Zusammenarbeit zwischen allen Rollen und Funktionen
- ist kein Team-Mitglied (in der Theorie…)

Das Team



- Typischerweise 5-9 Personen
- Funktionsübergreifend:
  - Qualitätssicherung, 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

Scrum - der Rahmen

Rollen

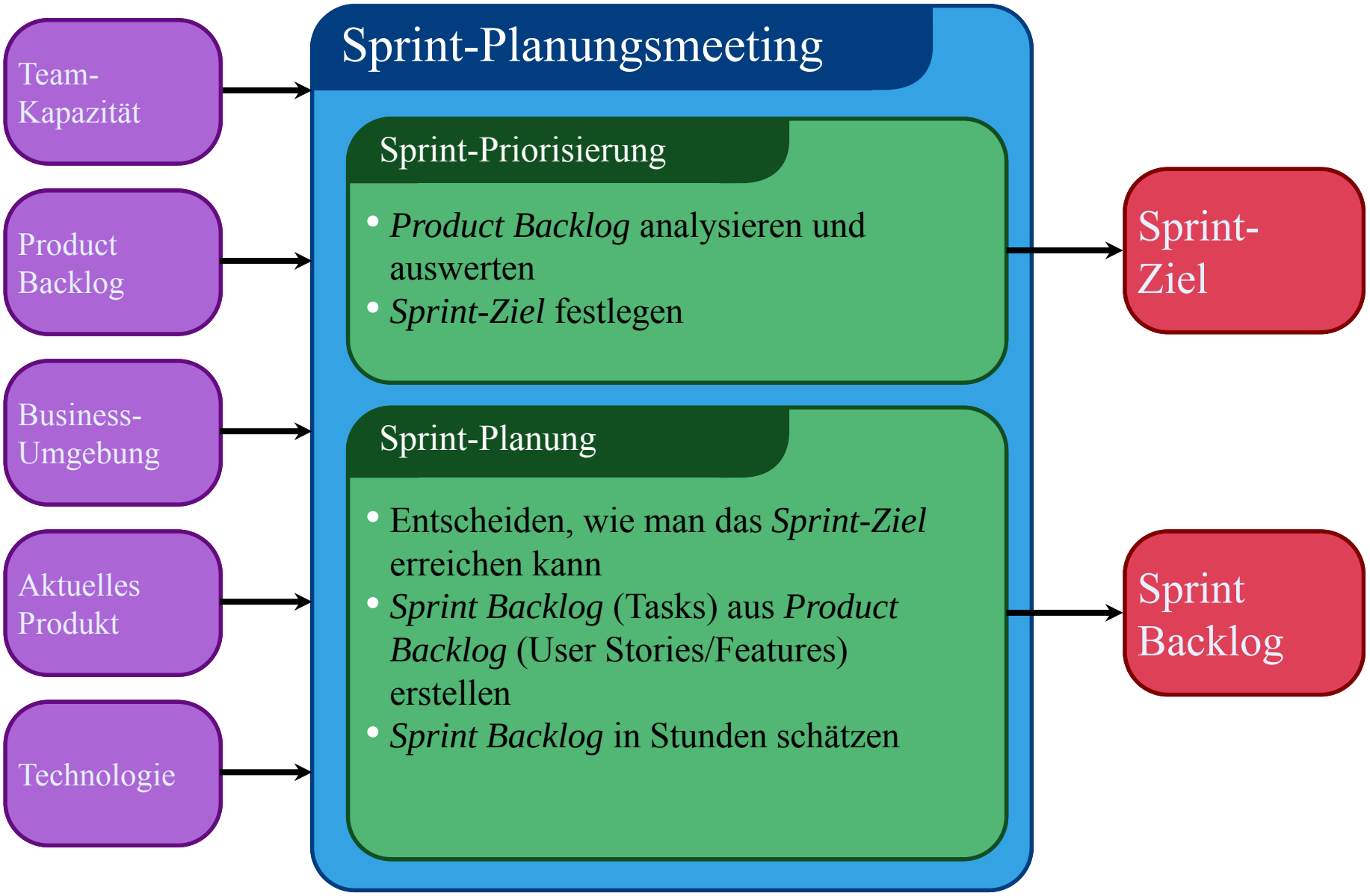
- Produkt-Owner
- ScrumMaster
- Team

Meetings

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

[illegible]

- Product Backlog
- Sprint Backlog
- Burndown-Diagramm


Team-Kapazität
Product Backlog
Business-Umgebung
Aktuelles Produkt
Technologie
Sprint-Planungsmeeting
Sprint-Priorisierung
• *Product Backlog* analysieren und auswerten
• *Sprint-Ziel* festlegen
Sprint-Planung
• Entscheiden, wie man das *Sprint-Ziel* erreichen kann
• *Sprint Backlog* (Tasks) aus *Product Backlog* (User Stories/Features) erstellen
• *Sprint Backlog* in Stunden schätzen
Sprint-Ziel
Sprint Backlog

Die Sprint-Planung

- Team wählt aus dem Product Backlog aus, was im nächsten Sprint umgesetzt werden soll. Dies kommt ins Sprint Backlog.
- Sprint Backlog wird gemeinschaftlich (!) erstellt
  - Tasks werden identifiziert und geschätzt (1-16 Stunden)
  - Es muss klar definiert sein, wann eine Task "fertig" ist!

Feature	Tasks mit Aufwandsschätzung

In unserem ZUUL-Abenteuer soll es einen magischen Gegenstand geben, bei dessen Berührung der Held an einen zufälligen Ort teleportiert wird.

Klasse für mag. Gegenstand (1 Stunde)
Methode "Teleportiere" (2 Stunden)
Tests für mag. Gegenstand (1 Stunde)
Grafik für mag. Gegenstand (2 Stunden)

Das tägliche Scrum-Meeting

- Täglich
- 15 Minuten lang
- im Stehen

- Nicht zur Problemlösung
  - Alle sind eingeladen
  - Aber nur Team-Mitglieder, der ScrumMaster, und der Produkt-Owner dürfen reden
- Hilft, andere/überflüssige Meetings zu vermeiden
- **NICHT** durch E-Mail ersetzbar!

Jeder beantwortet 3 Fragen

1. Was hast du gestern getan?
2. Was wirst du heute tun?
3. Welche Hindernisse sind dir im Weg?

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

Das Sprint-Review-Meeting

- Das Team präsentiert, was es während eines Sprints erreicht hat
- Typischerweise in Form einer Demo der neuen Features oder der zugrunde liegenden Architektur
- Informell
  - Regel: ‚Zwei Stunden zur Vorbereitung‘
  - Keine Folien
- Das ganze Team nimmt teil
- Laden Sie die ganze Welt ein!

Die Sprint-Retrospektiven

- Prüfen Sie regelmäßig, was gut und nicht so gut funktioniert (im Sprint selbst und bei der Planung)
- Ziel: Verbesserungsmöglichkeiten finden
- Typischerweise 15–30 Minuten lang
- Nach jedem Sprint
- Das ganze Team nimmt teil
  - Scrum Master
  - Produkt-Owner
  - Team
  - Vielleicht Endkunden und andere Personen (aber Vorsicht!)

Beginnen / aufhören / weitermachen

- Das gesamte Team kommt zusammen und diskutiert, wie es sich verbessern möchte:

beginnen mit ...

aufhören mit ...

weitermachen mit ...

Diese ist eine von vielen Methoden um Retrospektiven durchzuführen

Maßnahmen (Tasks) aus Retrospektive ableiten

Beschreibung für einmalige Maßnahmen:

Kurztitel

Beschreibung des Problems

Relevanz

Maßnahme

Verantwortlicher

Erledigungsdatum

Rollen

- Produkt-Owner
- ScrumMaster
- Team

Meetings

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

Artefakte

- Product Backlog
- Sprint Backlog
- Burndown-Diagramm

Der Product Backlog

Product Backlog

- Liste der offenen Anforderungen
- Eine Liste aller gewünschten Projektarbeiten
- Idealerweise soll jeder Eintrag wertvoll für Benutzer des Produktes oder Kunden sein
- Zu Beginn jedes Sprints neu priorisiert, und zwar vom Product Owner
- Beachte: Es können neue Anforderungen hinzukommen (seltener bestehende entfallen)

Product Backlog: Beispiel

Feature	Geschätzer Aufwand in Tagen
Speichern des aktuellen Spielstands	3
Anwenderdokumentation in deutscher Sprache	2
Graphische Oberfläche	20
Zaubertrunk versetzt Held in Obelix-Modus	4
…	

Formulierung von Anforderungen

- Anforderungen werden vom Product Owner vorgegeben.
- Dieser muss dazu in verschiedene Rollen schlüpfen.

- Häufig genutztes Format für Formulierung:

- "Als … möchte ich … damit …"

- User-Stories

Das Sprintziel

- Kurze Angabe, worauf der Schwerpunkt der Arbeiten während des Sprints liegt

Ziel des nächsten Sprints

Die Spielsteuerung soll mittels graphischer Benutzeroberfläche möglich sein.

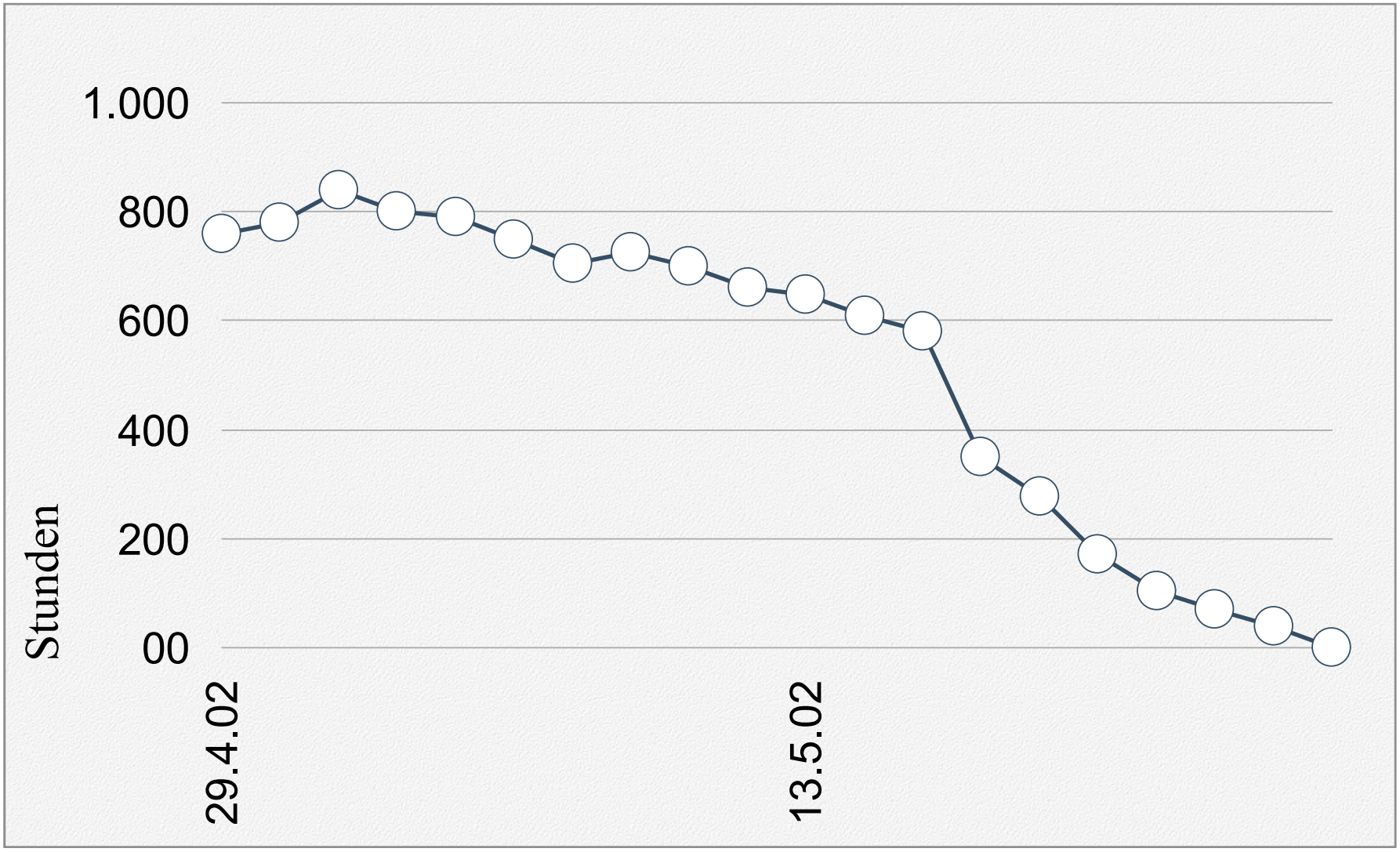
Management des Sprint Backlogs

- Team-Mitglieder wählen Tasks aus (Arbeit wird nicht 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
- Aktualisieren Sie die Zeitschätzung, sobald Sie mehr wissen

Sprint Backlog: Beispiel

Tasks	Mo	Di	Mi	Do	Fr
JUnit-Tests für GUI	8	4	8		
GUI-Klassen schreiben	16	12	10	4	
Online-Hilfe	12				8
Icon-Grafiken für GUI	8				
Fotos für Anzeige der Räume bearbeiten	8	8	8	8	8
Bugfix "Fackel leuchtet lila"			8	4	

Das Sprint Burndown-Diagramm

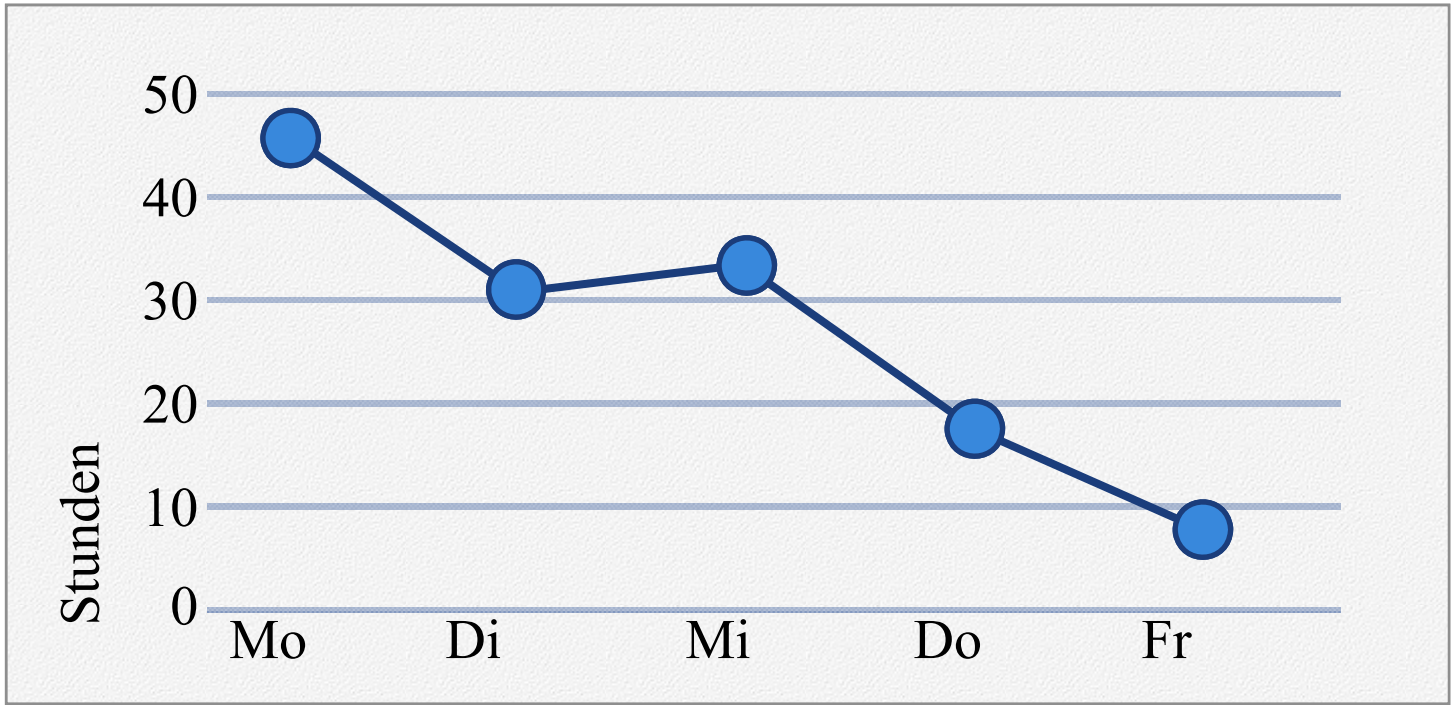


Das Sprint Burndown-Diagramm

- Das Burndown-Diagramm (engl. burndown chart) zeigt die im Sprint geleistete / noch zu erledigende Arbeit

- x-Achse: Zeit
- y-Achse: verbleibende Arbeit im aktuellen Sprint (Tasks oder geschätzter Aufwand)

Tasks	Mo	Di	Mi	Do	Fr
Code the user interface	8	4	8		
Code the middle tier	16	12	10	7	
Test the middle tier	8	16	16	11	8
Write online help	12				



Vorlagen für Backlogs

- http://agilesoftwaredevelopment.com/scrum/simple-product-backlog
- http://agilesoftwaredevelopment.com/scrum/simple-sprint-backlog

oder entsprechende Plugins in Redmine

Diskussion

- In welchen Projekten funktioniert Scrum?
- In welchen eher nicht?

- We have heard about new ways of developing software by paying consultants and reading Gartner reports. Through this we have been told to value:
- **Individuals and interactions over processes and tools**

  and we have mandatory processes and tools to control how those individuals (we prefer the term ‘resources’) interact
- **Working software over comprehensive documentation**

  as long as that software is comprehensively documented
- **Customer collaboration over contract negotiation**

  within the boundaries of strict contracts, of course, and subject to rigorous change control
- **Responding to change over following a plan**

  provided a detailed plan is in place to respond to the change, and it is followed precisely
- That is, while the items on the left sound nice in theory, we’re an enterprise company, and there’s no way we’re letting go of the items on the right.

Quelle: halfarsedagilemanifesto.org

Quellenangabe

- Teile dieser Präsentation wurden entnommen aus der deutschen Version von “An Introduction to Scrum” von Mike Cohn, übersetzt von Simon Roberts und Birgit Panzram.

- http://www.mountaingoatsoftware.com/presentations/30--an-overview-of-scrum

Urheberrechte

- Sie dürfen:
  - das Werk vervielfältigen, verbreiten und öffentlich zugänglich machen
  - Bearbeitungen des Werkes anfertigen
- zu den folgenden Bedingungen:
  - Namensnennung - Sie müssen den Namen des Autors/Rechteinhabers in der von ihm festgelegten Weise nennen (wodurch aber nicht der Eindruck entstehen darf, Sie oder die Nutzung des Werkes durch Sie würden entlohnt).
- Diese Lizenz lässt die Urheberpersönlichkeitsrechte unberührt.

- Weiteres hierzu unter http://creativecommons.org/licenses/by/3.0/