

# Agile Softwareentwicklung

Dozent: Florian Glufke

# Agile Softwareentwicklung Inhalt

- Scrum
- Planning Poker
- Task Board
- Azure DevOps
- User Stories

# Scrum – Einführung

- Rahmenwerk zur Entwicklung und Pflege komplexer (Software-)Produkte
- Ken Schwaber und Jeff Sutherland haben Scrum entwickelt
- Anfang der 1990er Jahre entstanden
- Definition: „Ein Rahmenwerk, innerhalb dessen Menschen komplexe adaptive Aufgabenstellungen angehen können, und durch das sie in die Lage versetzt werden, produktiv und kreativ Produkte mit höchstmöglichem Wert auszuliefern.“

# Scrum – Einführung

- Scrum ist:
  - Leichtgewichtig
  - Einfach zu verstehen
  - Schwierig zu meistern
- Rahmenwerk besteht aus:
  - Scrum Teams und ihre Rollen
  - Ereignissen
  - Artefakten
  - Regeln
- Besteht auf der Theorie empirischer Prozesssteuerung
- Hat einen iterativen und inkrementellen Ansatz

# Scrum – Einführung

- Drei Säulen der empirischen Prozesssteuerung
  - Transparenz
    - Aspekte nach einem gemeinsamen Standard definiert
    - gemeinsame Prozesssprache
    - gemeinsames Verständnis der „Definition of Done“
  - Überprüfung
    - Fortschritt regelmäßig in Bezug auf die Erreichung der Ziele überprüfen
  - Anpassung
    - Bei Abweichungen müssen Prozesse und zu bearbeitende Aufgaben angepasst werden

# Scrum – Einführung

- Ereignisse für Überprüfung und Anpassung:
  - Sprint Planning
  - Daily Scrum
  - Sprint Review
  - Sprint Retrospektive
  - (Refinement)

# Scrum – Einführung

- Scrum Werte:
  - Selbstverpflichtung
    - Scrum Team Mitglieder verpflichten sich die Ziele zu erreichen
  - Mut
    - Haben den Mut, das Richtige zu tun und an schwierigen Problemen zu arbeiten
  - Fokus
    - Fokussiert an den Zielen des Scrum Teams zu arbeiten
  - Offenheit
    - Offener Umgang mit allen Belangen der Arbeit und den damit verbundenen Herausforderungen
  - Respekt
    - Mitglieder von Scrum Teams respektieren sich gegenseitig

# Scrum – Einführung

- Scrum Teams bestehen aus
  - Product Owner
  - Scrum Master
  - Entwickler Team (Developer)



# Scrum – Team

- Product Owner
  - Verantwortung für Wertmaximierung des Produkts
  - Und für die Arbeit des Entwickler Teams
  - Management des Product Backlogs
  - Entscheidungsträger wie das Produkt weiterentwickelt wird

# Scrum – Team

- Entwickler Team
  - Profis die an dem Produkt arbeiten
  - Selbst organisiertes Team
  - Interdisziplinär, so dass alle Fähigkeiten, die benötigt werden vorhanden sind
  - Es gibt keine Unterscheidung zwischen den einzelnen Mitgliedern des Entwickler Teams
  - Teamgröße: 3 - 9

# Scrum – Team

- Scrum Master
  - Ist für die Durchführung von Scrum verantwortlich
  - Sorgt für die Einhaltung von Theorie, Praktiken und Regeln
  - Ist ein „Servant Leader“
  - Optimiert die Zusammenarbeit im Team
  - Unterstützt den Product Owner beim Managen des Product Backlogs

# Scrum – Team

- Scrum Master
  - Coaching des Entwicklerteams
  - Beseitigt Hindernisse, die das Entwicklerteam aufhalten
  - Unterstützt bei der Durchführung von Scrum Ereignissen
  - Führt Scrum in der Organisation ein
  - Zusammenarbeit mit anderen Scrum Mastern

# Scrum – Ereignisse

- Scrum definiert verschiedene Ereignisse mit definierter Regelmäßigkeit und zeitlicher Beschränkung (Time Box)
- Ereignisse dienen der Überprüfung und Anpassung
- Ereignisse in Scrum
  - Sprint
  - Sprint Planning
  - Daily Scrum
  - Sprint Review
  - Sprint Retrospektive
  - (Refinement)

# Scrum – Ereignisse

- Wichtigstes Ereignis in Scrum ist der Sprint
- Dauer: 3-4 Wochen
- Hat immer die gleiche Dauer
- Am Ende steht ein auslieferbares Produkt-Inkrement
- Zu einem Sprint gehören:
  - Sprint Planning
  - Daily Scrums
  - Entwicklungsarbeit
  - Sprint Review
  - Sprint Retrospektive

# Scrum – Ereignisse

- Anforderungsumfang darf sich während des Sprints ändern
- Sprint ist wie ein Projekt mit einem definierten Zeitrahmen
- Sprint sollte in der Regel nicht abgebrochen werden
- Sprint Planning
  - Werden die Arbeiten für den kommenden Sprint geplant
  - Plan entsteht durch die Zusammenarbeit des gesamten Scrum Teams
  - Max. 8 Stunden für einen einmonatigen Sprint
  - Scrum Master ist für die Durchführung verantwortlich
  - Product Owner beschreibt das Sprint-Ziel

# Scrum – Ereignisse

- Sprint Planning
  - Festlegen welche Backlog-Einträge das Sprint-Ziel erfüllen
  - Erarbeitung gemeinsames Verständnis der Arbeitsinhalte
  - Eingangsgrößen
    - Produkt-Backlog
    - Neuestes Produkt-Inkrement
    - Kapazität des Entwicklerteams
  - Entwicklerteam bestimmt die Anzahl der ausgewählten Produkt-Backlog-Einträge
  - Daraus resultiert das Sprint-Backlog
  - Entwicklerteam organisiert selber, wie es das Sprint-Ziel erreicht



# Scrum – Ereignisse

- Daily-Scrum findet täglich am gleichen Ort zur gleichen Zeit statt
- Zeitlicher Rahmen: 15 Minuten
- Dient zur Synchronisation der Aktivitäten des Entwicklerteams
- Und der Planung des kommenden Arbeitstages
- Das Entwicklerteam schildert hierzu:
  - Was habe ich gestern erreicht
  - Was werde ich heute erledigen
  - Gibt es Hindernisse oder Probleme

# Scrum – Ereignisse

- Daily Scrum dient zur:
  - Überprüfung des Fortschrittes Richtung Sprint-Ziel
  - Trend bei der Abarbeitung des Sprint-Backlogs
  - Sprint-Ziel täglich im Blick zu haben
- Scrum Master sorgt dafür, dass Daily Scrum stattfindet und die Regeln dazu eingehalten werden

# Scrum – Ereignisse

- Nutzen von Daily Scrum
  - Verbesserte Kommunikation
  - Macht andere Meetings überflüssig
  - Identifiziert und beseitigt Hindernisse
  - Fördert schnelle Entscheidungsfindung
  - Erhöht den Wissensstand des Entwicklungsteams
- Entscheidendes Meeting zur Überprüfung und Anpassung

# Scrum – Ereignisse

- Am Ende eines Sprints findet das Sprint Review statt
- Dient der Überprüfung des Produktinkrements
- Und der Anpassung des Produkt-Backlogs
- Scrum Team und Stakeholder betrachten das Ergebnis des Sprints
- Zeitlicher Rahmen: 4 Stunden
- Scrum Master organisiert das Meeting

# Scrum – Ereignisse

- Sprint Review beinhaltet:
  - Produkt Owner stellt vor, welche Product-Backlog-Einträge erledigt sind und welche nicht
  - Wie der aktuelle Stand des Produkt-Backlogs ist
  - Entwicklerteam zeigt:
    - Was lief gut
    - Welche Probleme gab es und wie wurden diese gelöst
    - Die Ergebnisse der erledigten Arbeiten
  - Alle Teilnehmer erarbeiten gemeinsam was als nächstes zu tun ist, als Input für das kommende Sprint Planning

# Scrum – Ereignisse

- Sprint Review beinhaltet:
  - Überprüfung der Marktsituation und Möglichkeiten des Produkteinsatzes
- Ergebnis Sprint Review:
  - Überarbeitetes Produkt-Backlog für die nächsten Sprints

# Scrum – Ereignisse

- Sprint Retrospektive dient zur Selbstüberprüfung des Scrum Team
- Daraus ergibt sich ein Verbesserungsplan für den kommenden Sprint
- Findet zwischen Sprint Review und Sprint Planning statt
- Zeitlicher Rahmen: 3 Stunden
- Scrum Master organisiert das Meeting

# Scrum – Ereignisse

- Sprint Retrospektive dient dazu:
  - Wie der vergangene Sprint im Bezug auf die beteiligten Personen, Beziehungen, Prozesse und Werkzeuge verlief
  - die wichtigsten gut gelaufenen Elemente und mögliche Verbesserungen zu identifizieren und in eine Reihenfolge zu bringen
  - einen Plan für die Umsetzung von Verbesserungen der Arbeitsweise des Scrum Teams zu erstellen
  - Anpassung der „Definition of Done“



# Scrum – Ereignisse

- Refinement dient dazu:
  - Einträge im Produkt-Backlog genauer zu definieren
  - Und in kleinere, präzisere Elemente zu zerlegen
  - Findet kontinuierlich statt
  - Kein definierter Zeitpunkt und Zeitrahmen
  - Erreichen der „Definition of Ready“
  - Product Owner mit Mitgliedern des Entwickler-Teams

# Scrum – Artefakte

- Scrum Artefakte repräsentieren die Arbeiten oder die Werte
- Bieten Transparenz und Möglichkeiten zur Überprüfung und Anpassung
- Zu den Artefakten gehören:
  - Produkt Backlog
  - Sprint Backlog
  - Inkrement
  - Definition of Done
  - (Definition of Ready)

# Scrum – Artefakte

- Product Backlog
  - Geordnete Liste mit allem was im Produkt enthalten sein kann
  - Dient als einzige Anforderungsquelle
  - Product Owner ist dafür verantwortlich
  - Niemals vollständig
  - Wird ständig gepflegt und weiterentwickelt
  - Enthält alle Features, Funktionalitäten, Verbesserungen und Fehlerbehebungen

# Scrum – Artefakte

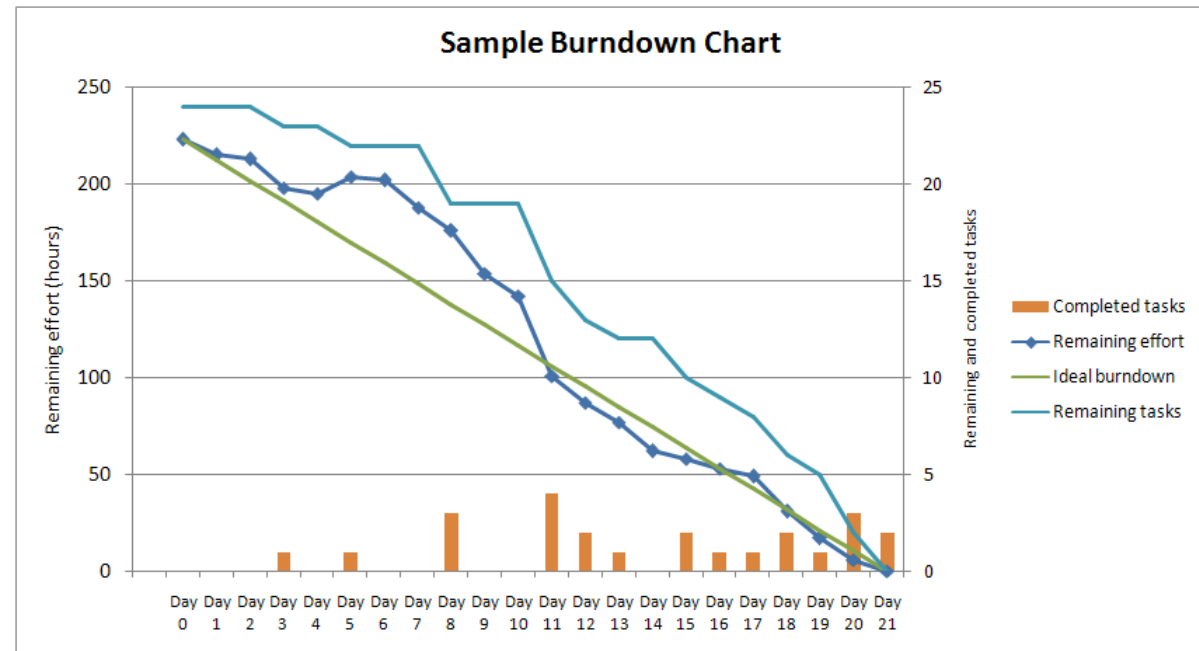
- Product Backlog-Eintrag enthält
  - Beschreibung
  - Position im Product-Backlog (Priorität)
  - Schätzung
  - Wert
- Es können auch mehrere Scrum Teams an einem Product-Backlog arbeiten
- Product Backlog wird ständig verfeinert: Einträge detaillierter beschreiben, Schätzungen hinzufügen, Reihenfolge ändern

# Scrum – Artefakte

- Product Backlog Einträge werden als „Ready“ gekennzeichnet wenn die Verfeinerung abgeschlossen ist
- Solche Einträge können im nächsten Sprint umgesetzt werden
- Für die Schätzung eines Eintrags ist das Entwicklerteam verantwortlich
- Product Owner hilft beim Verständnis der Einträge

# Scrum – Artefakte

- Verbleibende Arbeit zur Erreichung eines Ziels kann aufsummiert werden
- Zur Fortschrittsprognose werden Burndown- oder Burnup-Diagramme eingesetzt



# Scrum – Artefakte

- Sprint Backlog
  - Enthält alle Einträge vom Product-Backlog für den Sprint
  - Prognose des Entwicklerteams welche Funktionalität im Produkt-Inkrement enthalten sein wird
  - Enthält erforderliche Arbeiten für das Sprint-Ziel
  - Wird während des Sprints angepasst und erweitert
  - Zeigt den Fortschritt
  - Die Verantwortung liegt beim Entwicklerteam

# Scrum – Artefakte

- Sprint Backlog
  - Im Daily Scrum wird der Fortschritt und restlichen Arbeiten begutachtet
- Inkrement
  - Ergebnis am Ende eines Sprints
  - Enthält alle im Sprint Backlog umgesetzten Einträge
  - Es muss in einem verwendbaren Zustand sein
  - Muss nicht zwingend ausgeliefert werden



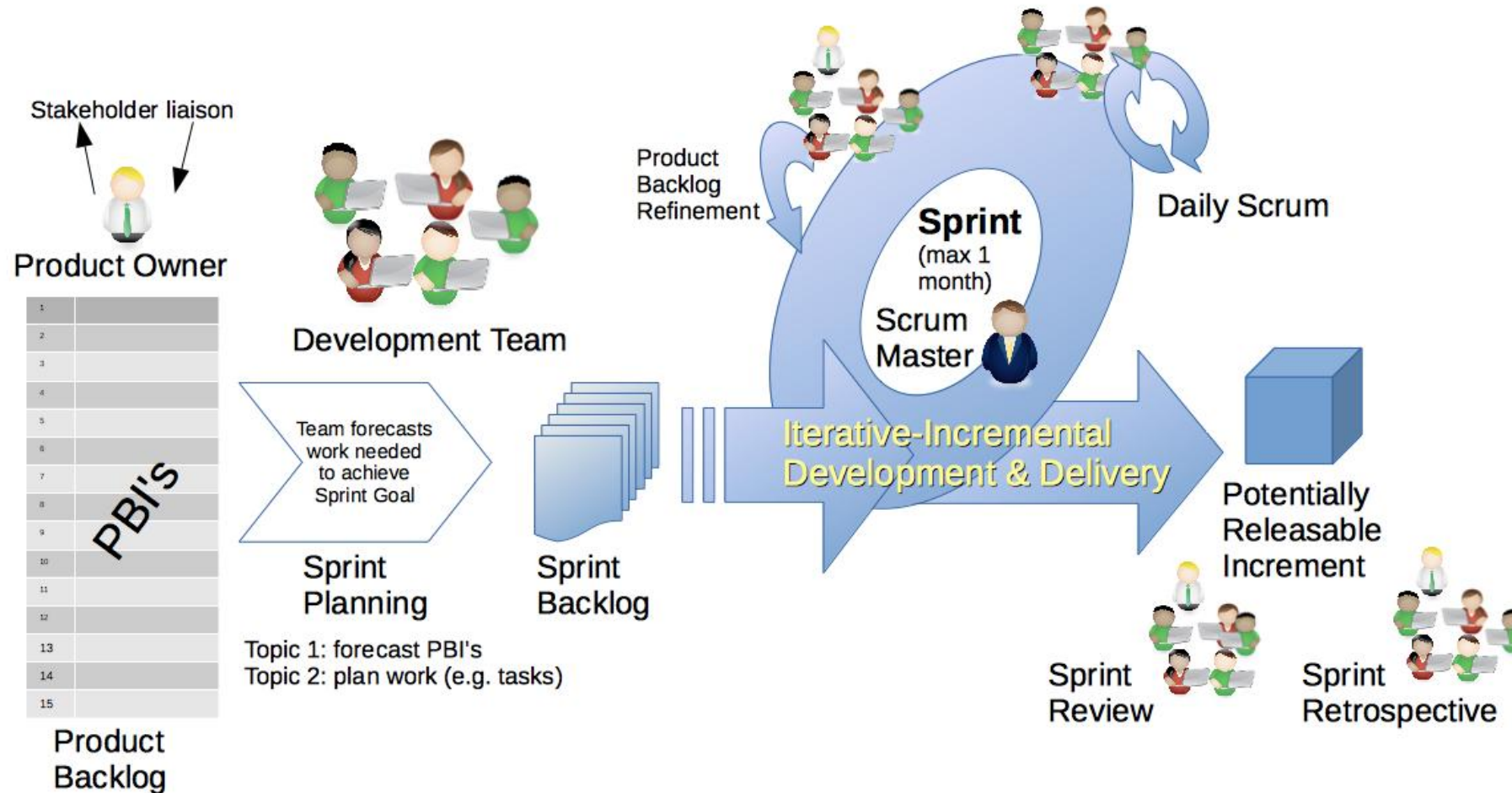
# Scrum – Definition of Done

- Alle müssen verstehen was „Done“ bedeutete
- Kann unterschiedlich sein von Scrum Team zu Scrum Team
- Scrum Team legt die „Definition of Done“ fest
- Arbeiten mehrere Scrum Teams an einem Produkt muss es eine gemeinsam „Definition of Done“ geben
- Legt fest wann ein Product Backlog-Eintrag als abgeschlossen gilt
- Kann erweitert werden, um eine höhere Produktqualität zu erreichen

# Scrum – Transparenz

- Transparenz ist ein wichtiger Aspekt von Scrum
- Ermöglicht Überprüfung
- Schwierigkeiten und Probleme werden so früh erkannt
- Sämtliche Artefakte müssen transparent für alle sein
- Ermöglicht fundierte Entscheidung zu treffen

# Scrum – Übersicht

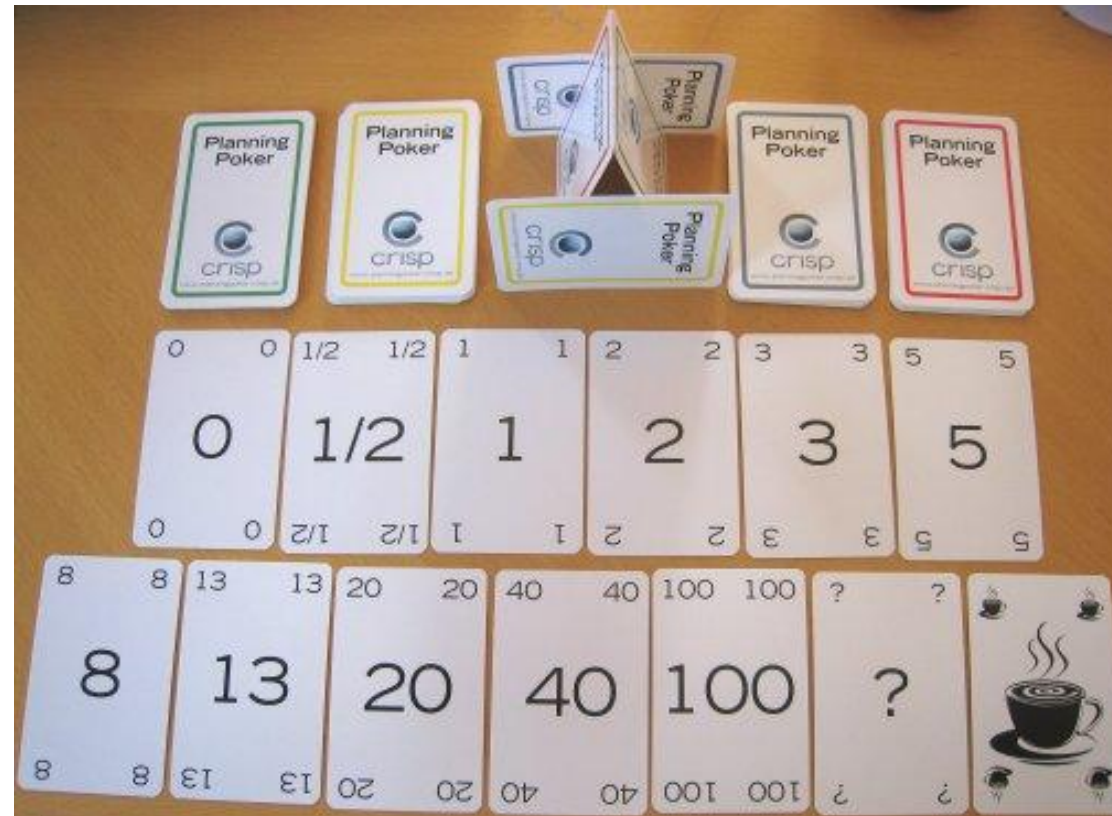


# Scrum – Nachteile/Schwierigkeiten

- Einführung sorgt für viel Veränderung in bestehenden Teams
- Bestehende Hierarchien werden in Frage gestellt
- Transparenz mag nicht jeder
- Steakholder müssen Scrum mittragen
- Schwierig einzusetzen bei Werkverträgen

# Planning Poker

- Methodik zur Aufwandsabschätzung



# Planning Poker

- Product Owner stellt die zu schätzende User Story (Product-Backlog Eintrag) vor
- Inhalt und Fragen werden geklärt
- Jeder Teilnehmer schätzt zunächst den Aufwand für sich alleine ab
- Hierfür werden die Karten verwendet

# Planning Poker

- Vermeidung von Beeinflussung
- Alle legen gleichzeitig ihre Schätzung offen
- Bei großer Abweichung erklären die mit niedrigster und höchster Schätzung
- So lange wiederholen bis Konsens gefunden
- Bei kleiner Abweichung Durchschnitt verwenden

# Planning Poker

- Bedeutung der Zahlen kann gewählt werden:
  - Arbeitsaufwand in Stunden
  - Story Points
- Kann auch zum Bewerten von Risiko/Schwierigkeitsgrad verwendet werden



# Task Board

- Zur Visualisierung des Sprint Backlogs wird Task Board eingesetzt
- Task Board ist eine Kanban-Tafel



# Task Board

- Anzahl der Spalten kann frei gewählt werden
- Typischerweise vier Spalten:
  - User Storys die umgesetzt werden
  - Unbearbeitete Tasks
  - Tasks in Bearbeitung
  - Erledigte Tasks (Definition of Done!)

# Task Board

- Weitere Spalten können z.B. sein
  - Test
  - Dokumentation
  - Review
  - Blockiert
- Task Board ist Grundlage des Daily Scrum
- Sorgt für Transparenz

# Azure DevOps

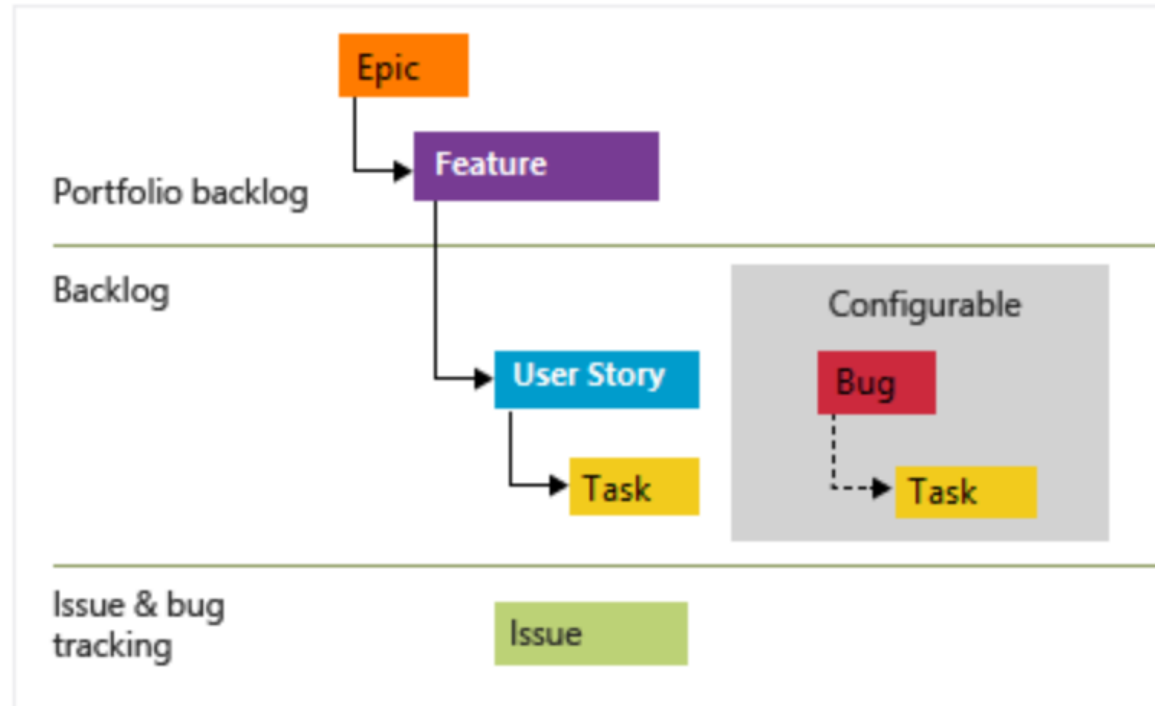
- Lösung von Microsoft (früher „Team Foundation Server“)
- Unterstützt Scrum Teams
- Integriert GIT
- Verwaltung von
  - Backlog
  - Sprints
  - Tasks
  - Bugs
- Task Board

# Azure DevOps

- Vergleichbare Lösungen:
  - Jira (kostenpflichtig)
  - Redmine (Open Source)
  - Agilo for Trac (Open Source)

# Azure DevOps

- Work Items



# Azure DevOps

- Backlog

Azure DevOps florianglufke / Kursverwaltung DHBW / Boards / Backlogs

Kursverwaltung DHBW Team

Backlog Analytics + New Work Item View as Board Column Options ...

Order	Work Item Type	Title	State	Iteration Path
1	Bug	Es werden nicht alle Kurse angezeigt	New	Kursverwaltung DHBW\Sprint 1
+ 2	User Story	Bestehenden Kurs löschen	New	Kursverwaltung DHBW\Sprint 1
3	User Story	Kurse anzeigen	Active	Kursverwaltung DHBW\Sprint 1
4	User Story	Anmelden bei der Kursverwaltung	New	Kursverwaltung DHBW\Sprint 1
5	User Story	Neue Benutzer anlegen	New	Kursverwaltung DHBW\Sprint 2
6	User Story	Kurs anlegen	New	Kursverwaltung DHBW\Sprint 2
7	Bug	Anmeldung der Studenten funktioniert nicht	New	Kursverwaltung DHBW\Sprint 1
8	User Story	Bestehenden Kurs bearbeiten	New	Kursverwaltung DHBW

**Planning**

Drag and drop work items to include them in a sprint.

Kursverwaltung DHBW Team Backlog

Sprint 3 **Current** 25.04.2022 - 20.05.2022  
20 working days

No work scheduled yet

+ New Sprint

# Azure DevOps

- Task Board

**Azure DevOps** florianglufke / Kursverwaltung DHBW / Boards / Sprints

Kursverwaltung DHBW Team

28. März - 22. April  
20 work days

Sprint 2 | Person: All

**Taskboard** | Backlog | Capacity | Analytics | + New Work Item | Column Options

**Taskboard Columns:**

- New 48 h**
  - 2 Neue Benutzer anlegen (40 h, State: New, Assignee: Florian Glufke)
  - 12 Oberfläche zum Anlegen eines neuen Benutzers (16, State: New, Assignee: Unassigned)
  - 14 Unit-Test für das Anlegen eines Benutzer erstellen (8, State: New, Assignee: Florian Glufke)
  - 16 Kurse anzeigen (24 h, State: Active, Assignee: Florian Glufke)
- Active 16 h**
  - 11 Implementierung der Business-Logik einen neuen Benutzer anzulegen (16, State: Active, Assignee: Florian Glufke)
  - 17 Oberfläche zum Anzeigen einer Liste aller belegter Kurse implementieren (24, State: New, Assignee: Unassigned)
- Closed**
  - 7 Tabelle für Benutzer erstellen (State: Closed, Assignee: Unassigned)



# User Stories

- Beschreibt gewünschte Funktionalität einer Software
- Besteht aus:
  - Beschreibung
  - Diskussionen
  - Akzeptanzkriterien
- Stellt einen Wert für einen Benutzer dar

# User Stories

- Wichtige Eigenschaften von User Stories:
  - Unabhängig
  - Verhandelbar
  - Wertvoll für Kunden und Benutzer
  - Schätzbar
  - Klein
  - Testbar

# User Stories

- Beispiel für User Stories:
  - „Ein Benutzer kann eine Bewertung für ein Produkt abgeben“
  - „Ein Benutzer kann nach Produkten suchen“
- Schlechte Beispiele für User Stories:
  - „Die Software wird in Python geschrieben“
  - „Die Software wird Verbindungen zur Datenbank über einen Verbindungspool aufbauen“

# User Stories

- Detail einer User Story kann zusätzliche User Story ergeben
- User Stories sollten aufgeteilt werden wenn sie zu groß sind
- Üblicher Umfang:
  - Halber Tag für ein Entwickler
  - bis zu zwei Wochen für zwei Entwickler
- Details einer User Story sollen diskutiert werden statt alles aufzuschreiben
- User Story kann um Notizen erweitert werden

# User Stories

- Beispiel Aufteilung einer User Story
  - „Ein Benutzer kann nach Produkten suchen“
  - Wird aufgeteilt in:
    - „Ein Benutzer kann nach Produkten suchen durch Angabe von Name des Produktes, Preisbereich, Bewertung und Name des Herstellers“
    - „Ein Benutzer kann sich zum Ergebnis der Suche zu jedem Produkt den Namen, ein Foto und den Preis anzeigen lassen“.
    - „Ein Benutzer kann sich die Ergebnisse einer Suche nach dem Namen und dem Preis sortieren lassen“.

# User Stories

- User Story ist keine feste Vereinbarung
- Kann sich während der Entwicklung verändern
- Wird geschrieben vom Kunden-Team
  - Stellen sicher, dass die Anforderungen der Benutzer erfüllt werden
  - Kunden-Team besteht aus: Tester, Produktmanager, tatsächlichen Benutzern und Designern
- Wird in der Geschäftssprache geschrieben
- Enthält kein technisches Fachjargon
- Wird priorisiert anhand des Wertes für den Benutzer (und Aufwand)

# User Stories

- Akzeptanzkriterien definieren wann die User Story abgeschlossen ist
- Stellen sicher, dass die Erwartungen der Kunden erfüllt werden
- Beispiele für Akzeptanzkriterien:
  - Die Anzahl der Suchergebnisse wird korrekt angezeigt
  - Die Suchergebnisse sind standardmäßig nach dem Namen alphabetisch sortiert
  - Es werden maximal 100 Ergebnisse angezeigt

# User Stories

- Vorteile von User Stories
  - Stärken die verbale Kommunikation anstatt die geschriebene
  - Werden von allen verstanden
  - Haben eine passende Größe für die Planung
  - Passen zu einem iterativen Entwicklungsprozess
  - Verzögern das Befassen mit Details, bis das nötige Verständnis hierfür vorhanden ist oder bis eine User Story realisiert wird



# User Stories

- Können mit Planning Poker geschätzt werden
  - Werden üblicher Weise mit StoryPoints geschätzt
- Einsatz mit Scrum
  - Backlog enthält UserStories
  - Sprints können mit UserStories geplant werden
  - Sprint hat bestimmte Velocity
    - Menge der Arbeit die in einem bestimmten Sprint umgesetzt werden kann
    - Wird in StoryPoints angegeben
  - Entsprechend können UserStories auf verschiedene Sprints verteilt werden

# Referenzen

- Scrum Guide <https://www.scrum.org/resources/scrum-guide>
- Essential Scrum – A practical Guide to the most popular agile Process by Kenneth S. Rubin
- Azure DevOps <https://azure.microsoft.com>
- User Stories Applied – For Agile Software Development by Mike Cohn