Softwaretechnik C - Softwaremanagement

LE 04: Vorgehens- und Prozessmodelle

Fachhochschule Dortmund
University of Applied Sciences

Organisatorisches

Reflektion

- Motivation

Reflektion

- Von der Produktvision bis zum erfolgreichen Produkt:
  Was ist zu tun? Was ist zu beachten? Rahmen für die Umsetzung?
  - Anforderungen – Woher? Validität?
  - Budget?
  - Zeit / Termine / Fristen?
  - Mitarbeitende, Funktionsbereiche & Teams?
  - Qualität und Umfang?
  - Bereitstellung / Betrieb der Lösung?
  - Andere Projekte / Produkte, die parallel bearbeitet werden?

Reflektion

- Inhalte

  - Vorgehens- und Prozessmodelle der Softwaretechnik
    - Wann eignet sich der Einsatz?
    - Wann eignet sich welches Modell?
    - Wesentliche Merkmale und Funktionsweisen?

  - Anforderungsmanagement
  - Qualitätsmanagement
  - Konfigurationsmanagement
  - Risikomanagement
    - Risikomanagementprozess und Planung von Gegenmaßnahmen
  - Produktmanagement
    - Anforderungen, Änderungen, Lieferantenvereinbarungen
  - Prozessverbesserung

Agenda

- Klassische Vorgehensmodelle
  - Wasserfallmodell
  - Nebenläufiges Modell
  - Spiralmodell
  - Prototyping
- Monumentale Vorgehensmodelle
  - V-Modell XT
  - Rational Unified Process (RUP)
- Agile Modelle
  - **Scrum**
  - Kanban (IT-Kanban)
  - Extreme Programming (XP)
  - Feature Driven Development
  - Adaptive Software Development
  - Crystal
  - Lean Software Development
  - SAFe

Scrum

- Inhalte und Ablauf
- Rollen
- User Stories und Anforderungsspezifikation
- Projektfortschrittskontrolle
- Klassifikation

Scrum

- Inhalte und Ablauf
- Rollen
- User Stories und Anforderungsspezifikation
- Projektfortschrittskontrolle
- Klassifikation

Scrum
Inhalte und Ablauf

- Schwaber, Sutherland und Beedle haben Scrum im Bereich der Softwareentwicklung bekanntgemacht
- Kein Akronym, sondern ein Begriff aus dem Rugby
  - Bezeichnet das enge Gedränge bei erneutem Spielstart in Folge eines Fouls
- Basiert auf den Grundideen einer schlanken Produktion (**Lean Production**) und besteht aus wenigen klaren Regeln
- Berücksichtigt
  - dass in jedem Projekt **neue Erfahrungen und Erkenntnisse** gewonnen werden
  - die Erkenntnis, dass Projektentwicklung empirische Anteile hat und somit **nur teilweise vorhersehbar** ist

Scrum
Inhalte und Ablauf

- Projekt wird durch explizite **Überwachungskriterien** und **Rückkopplungsmechanismen** kontrolliert
- Die Regeln von Scrum müssen **konsequent und diszipliniert** eingehalten werden, damit Scrum funktioniert
- Scrum schafft ein notwendiges Maß an **Transparenz**:
  - Alle Aktivitäten, die zur Planung und Erstellung eines Softwareprodukts notwendig sind, werden innerhalb weniger Wochen ausgeführt
  - Tatsächlicher Projektfortschritt wird schnell und unmittelbar sichtbar

Scrum
Inhalte und Ablauf

Fachhochschule Dortmund
University of Applied Sciences

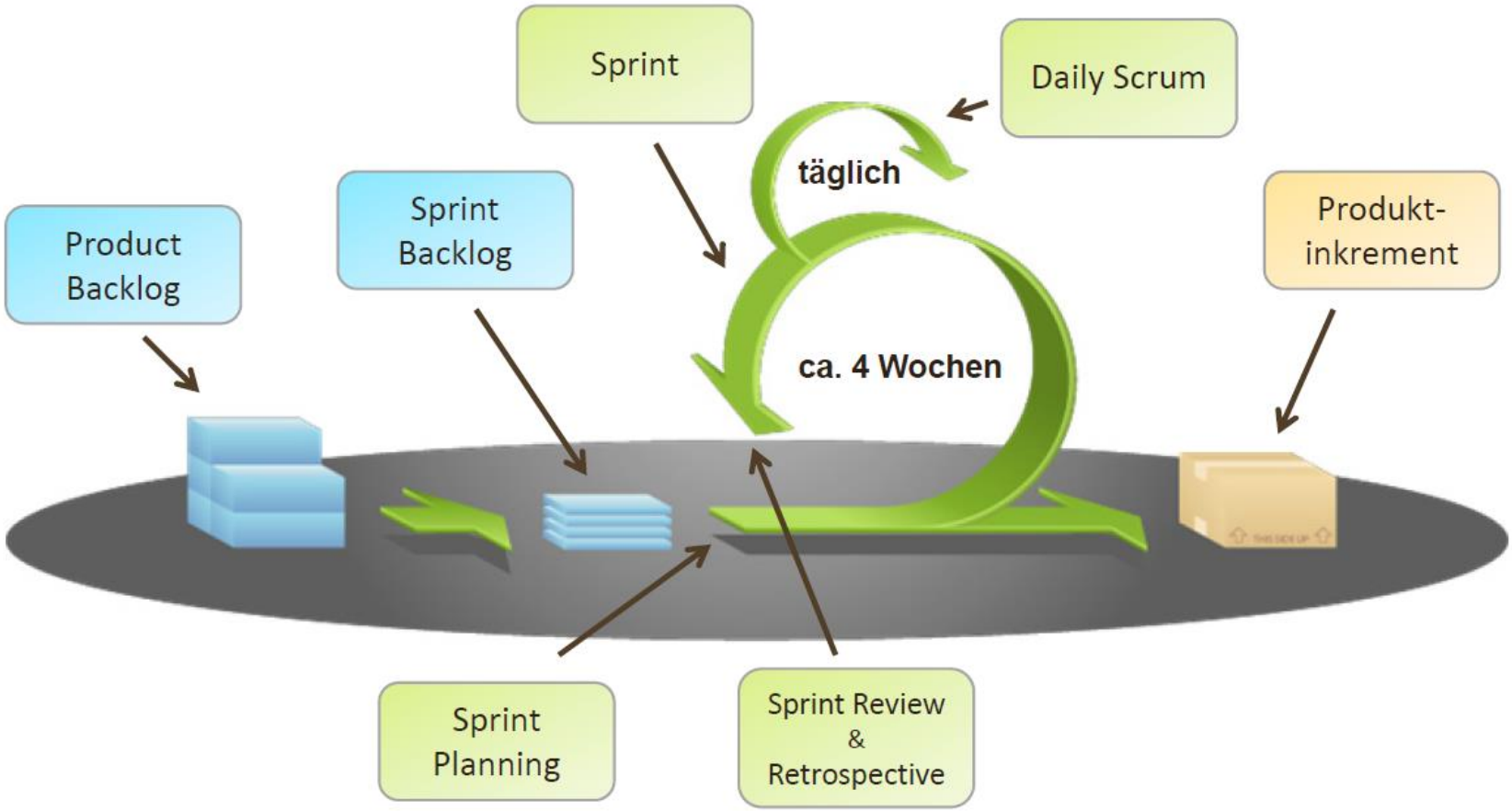
- Scrum-Projekt besteht aus einer Reihe kurzer Arbeitszyklen (**Sprints**)
- Jeder Zyklus wandelt Anforderungen aus dem Product Backlog in ein auslieferbares **Produktinkrement** um
  - Produktinkrement ist lauffähig, getestet und dokumentiert
- **Dauer** eines Sprints
  - Wird bspw. mit den Teamdefinitionen festgelegt
  - beträgt zwischen 1 Woche und 1 Monat (max. 30 Tage)
- Jeder Sprint endet zum vereinbarten **Termin** (Timeboxing)
- Bevor ein Scrum-Projekt starten kann, müssen die Rollen **Product Owner, Team und Scrum-Master** einsatzbereit sein
- Zudem muss das **Product Backlog** initial mit Anforderungen gefüllt sein

Scrum
Inhalte und Ablauf

- Bei Scrum werden sämtliche fachlichen und technischen Anforderungen in einem sog. **Product Backlog** (de facto das Anforderungsdokument) gesammelt
- Der Auftraggeber (**Product Owner**) legt die Prioritäten der Umsetzung der Anforderungen fest
- Die eigentliche Softwareentwicklung erfolgt in sogenannten **Sprints**
- Vor jedem Sprint wählt der Product Owner Anforderungen aus dem Product Backlog aus und überträgt sie – mit Prioritäten versehen – in das **Sprint Backlog**
- Diese Anforderungen soll das Team im Rahmen eines Sprints realisieren, sodass ein lauffähiges **Software-Inkrement** entsteht

Scrum
Inhalte und Ablauf

Fachhochschule Dortmund
University of Applied Sciences



Scrum
Inhalte und Ablauf

- Der Endtermin eines Sprints darf nicht überschritten werden
- Um den Termin einzuhalten, kann der **Sprintumfang** in Abstimmung zwischen dem Team und dem Product Owner **reduziert** werden
- Ein **Scrum Master** sorgt dafür, dass das Team ungestört arbeiten kann
- Scrum Master ist entweder ein Berater, Coach oder Projektmanager
- Während eines Sprints sind **keine Änderungen** an den Anforderungen oder Zielen des aktuellen Sprint Backlogs erlaubt

Scrum
Scrum Board und Teamdefinitionen

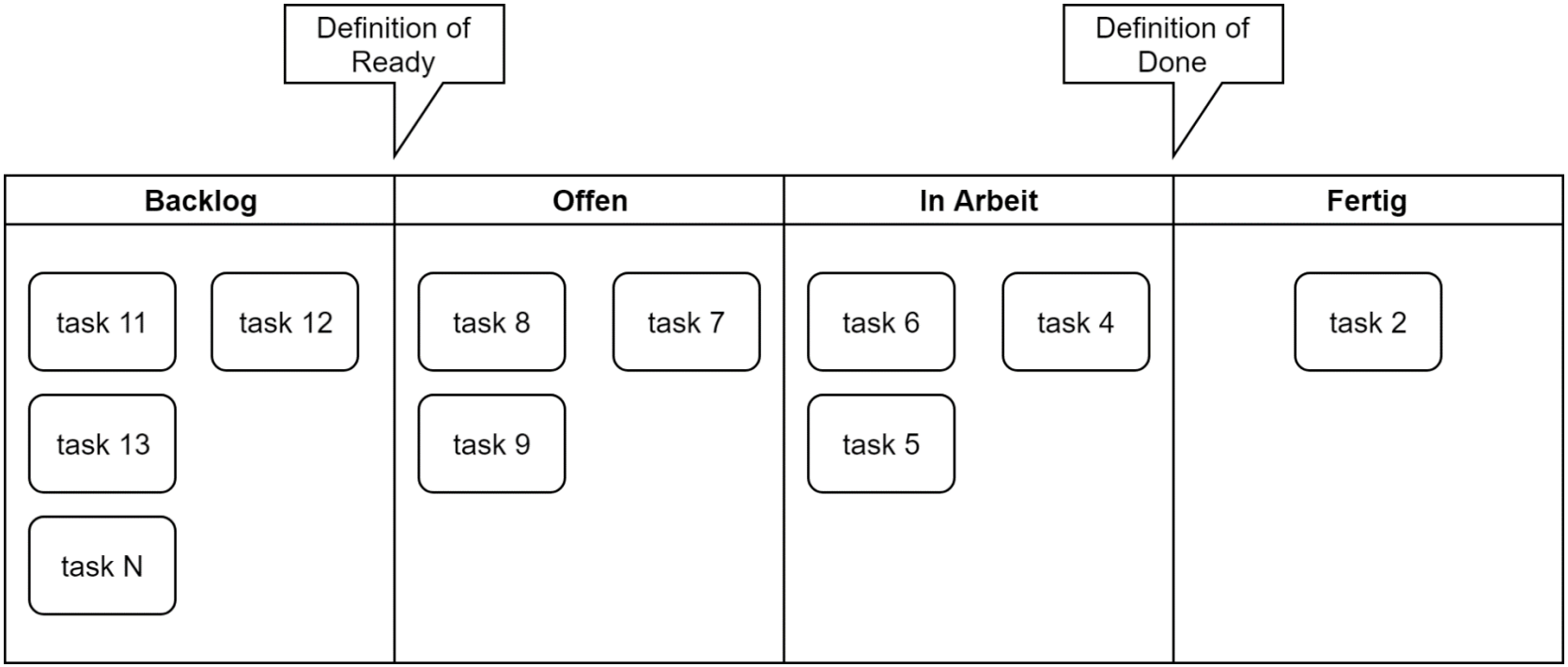
- Ähnlich wie bei Kanban kann das **Sprint Backlog** mittels **Scrum Board** visualisiert werden
- Die Struktur und die Übergangsvoraussetzungen in eine folgende Phase werden üblicherweise mit den **Teamdefinitionen** festgelegt
- **Teamdefinitionen** sind die **individuell vereinbarten Prinzipien der Zusammenarbeit im Team**

Scrum
Scrum Board und Teamdefinitionen

- Wesentliche und häufige Teamdefinitionen sind:
  - Definition of Done (DoD)
    - Allgemeingültige Definition der für die Fertigstellung einer User Story zu erbringenden Kriterien, bspw.:
      - Automatisierte Tests sind existieren und laufen
      - Notwendige Dokumentation wurde ergänzt
      - Der Product Owner hat die User Story abgenommen
      - Mindestens 2 Entwickler haben an der User Story gearbeitet.
  - Definition of Ready (DoR)
    - Allgemeingültige Definition der für den Entwicklungsbeginn einer User Story zu erbringenden Kriterien, bspw.:
      - Die User Story ist geschätzt
      - Die User Story wird von allen verstanden
      - Die User Story hat einen klaren Geschäftswert
      - Die User Story ist nicht größer als 20 Story Points
      - Die User Story beschreibt nur eine Anfoderung
      - Die User Story enthält klare Akzeptanzkriterien

Scrum
Scrum Board und Teamdefinitionen

Fachhochschule Dortmund
University of Applied Sciences



Scrum Ceremonials

- Daily Scrum
  - Wann: täglich
  - Dauer: 15 min
  - Teilnehmer: Scrum Team, Scrum Master
  - Ziel: Statusabgleich der Tasks im Team, Identifikation von Problemen
- Backlog Refinement / Story Time
  - Wann: Während eines Sprints
  - Dauer: ~ 10% des Sprints
  - Teilnehmer: Scrum Team, Scrum Master, Product Owner
  - Ziel: Verfeinerung von User Stories / Backlog Items zur Planungsoptimierung (Sprint Planning, Release Planning)
  - Durchführung: wie Teil 1 des Sprint Plannings, jedoch ohne Sprint Backlog. Auf diese Weise kann der Product Owner User Stories verfeinern und besser priorisieren oder Sprints / Releases auf Basis der Velocity vorausschauend planen.

Scrum Ceremonials

Fachhochschule Dortmund
University of Applied Sciences

- Sprint Planning
  - Wann: Zu Beginn eines neuen Sprints
  - Dauer: ~ 5% des Sprints
  - Teilnehmer: Scrum Team, Scrum Master, Product Owner
  - Ziel: Planung des nächsten Sprints bzw. Sprint-Umfangs und Bestimmung des Sprint Backlogs
  - Durchführung:
    - Teil 1: Der Product Owner erläutert die User Stories in absteigender Reihenfolge gemäß Priorität im Backlog. Das Team diskutiert die Anforderungen und gibt eine Aufwandsschätzung ab. Ist die Velocity erreicht, steht das Sprint-Backlog fest.
    - Teil 2: Das Scrum-Team diskutiert jede User Story im aktuellen Sprint Backlog und legt fest, wie die Stories realisiert werden sollen. Dabei können Stories auch zerlegt werden. Die Teilnahme des Product Owners in Teil 2 ist nicht erforderlich. Sollten sich Fragen oder Änderungen bzgl. Aufwandsschätzung ergeben, werden diese jedoch umgehend mit dem Product Owner geklärt. Am Ende kann ein neuer Sprint auf Basis des so verfeinerten Sprint Backlogs beginnen.

Scrum Ceremonials

- Sprint Review
  - Wann: Am Ende eines Sprints
  - Dauer: ~ 2,5% des Sprints
  - Teilnehmer: Scrum Team, Scrum Master, Product Owner, Stakeholder
  - Ziel: Präsentation des Produktinkrements und Feedback
  - Durchführung:
    - Der Product Owner stellt das Sprint-Ziel vor.
    - Das Scrum Team präsentiert das Produktinkrement und die erreichten Ziele des zurückliegenden Sprints.
    - Die Zuhörer / Stakeholder geben Feedback.

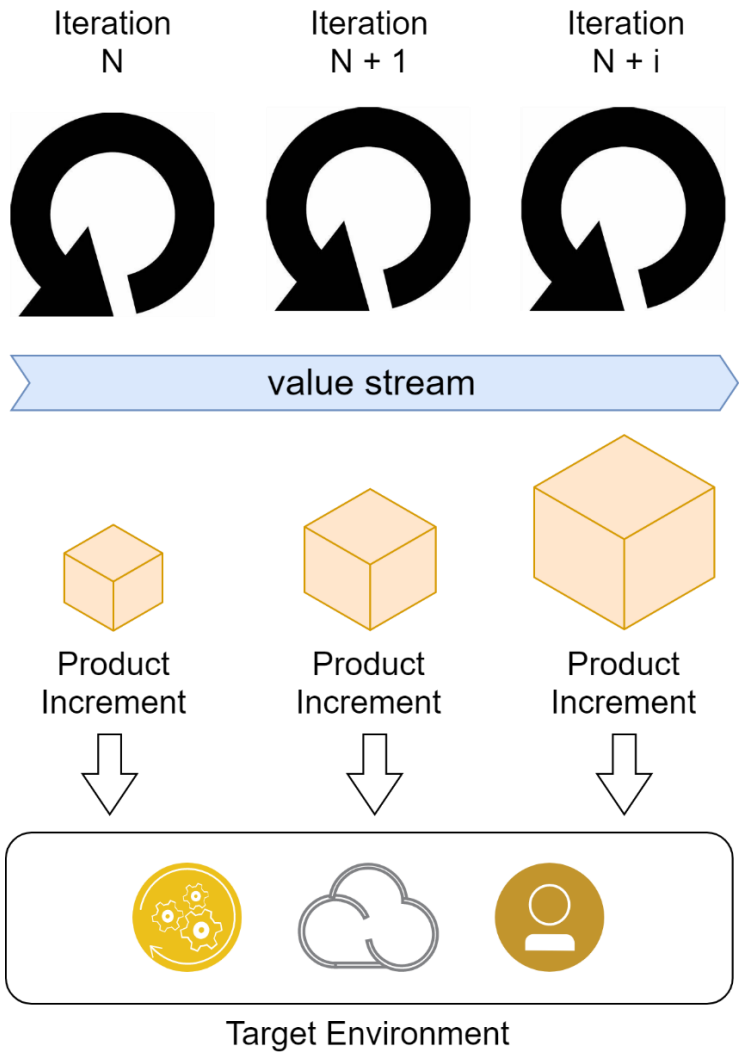
Scrum Ceremonials

- Sprint Retrospective
  - Wann: Am Ende eines Sprints
  - Dauer: ~ 2% des Sprints
  - Teilnehmer: Scrum Team, Scrum Master
  - Ziel: Identifikation von Verbesserungspotential im Prozess und Austausch zur Verbesserung der Zusammenarbeit
  - Durchführung:
    - Der Scrum Master nimmt eine moderierende Rolle ein. Das Event soll dabei zielführend und ergebnisorientiert durchgeführt werden. Üblicherweise erfolgt die Retrospektive in folgenden Phasen:
      - Startphase
      - Informationen zusammentragen
      - Einsichten gewinnen
      - Entscheidungen herbeiführen
      - Abschluss

Scrum
Agile Werte

Fachhochschule Dortmund
University of Applied Sciences

Scrum
Mehrwert und Produktinkrement



Scrum

- Inhalte und Ablauf
- Rollen
- User Stories und Anforderungsspezifikation
- Projektfortschrittskontrolle
- Klassifikation

Scrum
Rollen

- Scrum kennt drei **Rollen**
  - 1. Product Owner, 2. Team und 3. Scrum-Master
- Alle drei Rollen müssen adäquat besetzt sein (Staffing-Prozess)
- Alle drei Rollen müssen **eng zusammenarbeiten**, um ein Scrum-Projekt zum erfolgreichen Abschluss zu führen
  - Schließlich stellt das Agile Manifest klar, dass Individuen und Interaktionen wichtiger als Prozesse und Werkzeuge sind [Beck et al. 2001]
- Typischerweise arbeitet ein **Product Owner** mit einem Team zusammen
- Jedes Team hat einen eigenen **Scrum-Master**
- Jeder Scrum-Master betreut ein **Team**

Scrum
Rollen

1. Product Owner entscheidet,
   - Welche Anforderungen für eine Version umgesetzt werden sollen
   - Wann die Software ausgeliefert wird
2. Team
   - Führt die Arbeit aus
   - Entscheidet, wie viele Anforderungen es in einem Sprint verlässlich in einem Produktinkrement umsetzen kann
   - Organisiert seine Arbeit selbstständig
3. Scrum-Master
   - Hilft allen Beteiligten, Scrum richtig anzuwenden
   - Unterstützt das Team, seine Produktivität kontinuierlich leisten und verbessern zu können

Scrum
Rolle: Product Owner

Fachhochschule Dortmund
University of Applied Sciences

- **Repräsentiert die Endkundenbedürfnisse**, im Idealfall zu 100%
- Steuert die Softwareentwicklung durch **Priorisierung der Anforderungen** aus dem Product Backlog
- Arbeitet mit dem Team über die gesamte Projektlaufzeit eng zusammen
- Rolle des Product Owner ist „mehr“ als ein „gewöhnlicher“ Produkt- bzw. Projektmanager bzw. Projektleiter
- Rolle vereint klassische Produkt- und Projektmanagement-Aufgaben
- Product Owner hat die **zentrale** Stellung in Scrum
  - Anforderungen werden nicht einmal zu Beginn fest definiert, eingefroren und dann an die Entwicklung übergeben
  - Umsetzung des Produkts wird nicht einfach an einen Projektleiter delegiert
  - Bei Scrum ist der Product Owner direkt und fest in die SWE integriert

Scrum
Rolle: Product Owner

Fachhochschule Dortmund
University of Applied Sciences

- Anforderungsbeschreibung und Anforderungsmanagement
  - Erfassung der Kundenwünsche und -anforderungen
  - Erstellung Produktkonzept und Product Backlog
  - Kontinuierliches Anforderungsmanagement (Verfeinern, Eliminieren etc.)
  - Enge und regelmäßige Abstimmungsprozesse mit dem Kunden
- Release-Management und Return on Investment
  - Verantwortung für das Erreichen der Projektziele
  - Entscheidet über Auslieferungszeitpunkt, Funktionalität und Kosten der Softwareversion
  - Erstellt und aktualisiert den Releaseplan
  - Führt wichtige Projektmanagementaufgaben selbst durch

Scrum
Rolle: Product Owner

Fachhochschule Dortmund
University of Applied Sciences

- Enge Zusammenarbeit mit dem Team
  - Hilft dem Team die Kundenwünsche und –anforderungen zu verstehen
  - Hilft dem Team eine Transformation in Produktinkremente vorzunehmen
  - Schnittstelle zwischen dem Kunden und der eigentlichen Softwareentwicklung
  - Detaillierung und Präzisierung der im nächsten Sprint umzusetzenden Anforderungen
  - Teilnahme an allen Scrum-Besprechungen
  - Überprüfung und Abnahme der in einem Sprint erzielten Arbeitsergebnisse
  - Tipp: Regelmäßig mit Team zusammenarbeiten (1h nach dem Daily Scrum)
  - Fragen und Feedback zu Arbeitsergebnissen IM laufenden Sprint

Scrum
Rolle: Product Owner

- Stakeholder-Management
  - Product Owner bindet die verschiedenen Interessengruppen inklusive den Endkunden ein
  - Product Owner bündelt und filtert die verschiedenen Anforderungen
  - Stakeholder kommen z.B. aus Marketing, Vertrieb, Service und IT
  - Meistens wichtigste PO-Aufgabe: Anforderungen des Kunden verstehen UND an das Team kommunizieren können

Scrum
Rolle: Product Owner

- Product Owner als Chief Engineer
  - Hybrid aus Produktmanager, Chefarchitekt und Projektleiter
  - Verantwortlich für den Projektplan und die Projektsteuerung
  - Verfügt über fundiertes technisches Wissen und trifft ggf. grundsätzliche Architekturentscheidungen
  - Umfassender Überblick über sämtliche Entwicklungsaktivitäten
  - Optimierung des Entwicklungsprozesses: Eliminierung überflüssiger Tätigkeiten, Verhinderung der Überlastung einzelner Mitarbeiter, ausgeglichener Arbeitsfluss

Scrum
Rolle: Product Owner

Fachhochschule Dortmund
University of Applied Sciences

- Product Owner vertritt die Auftraggeberseite in fachlicher Hinsicht
- Ermittelt und aktualisiert kontinuierlich die Kunden-Anforderungen
- Product Owner muss vom Management (Auftragnehmer) bevollmächtigt sein, eigenständige Entscheidungen hinsichtlich Funktionsumfang und/oder Priorisierung der Anforderungen treffen zu können
- Oberstes Ziel ist für den Product Owner die Rentabilität (Wirtschaftlichkeit) des Projekts und Produkts
  - Und das kann er nicht erfüllen, wenn er von Seiten des Kunden bzw. Auftraggebers kommt

Scrum
Rolle: Scrum-Team

Fachhochschule Dortmund
University of Applied Sciences

- Führt alle Arbeiten aus, die zur **Umsetzung der Anforderungen** in auslieferbare Produktinkremente notwendig sind
- Softwareentwickler, Architekten, Qualitätssicherung
  - arbeiten in Scrum somit eng zusammen und
  - Erstellen die Arbeitsergebnisse gemeinschaftlich
- Wichtiger Staffing-Prozess, sodass die richtigen Mitarbeiter(innen) im Team mitarbeiten
  - Wissen, Erfahrungen, Fähig- und Fertigkeiten
  - Mitarbeiter bestimmen mit, an welchen Projekten sie mitarbeiten

Scrum
Rolle: Scrum-Team

- Scrum-Team
  - ist bevollmächtigt
  - entscheidet allein, wie viele Anforderungen es innerhalb des nächsten Sprints in ein Produktinkrement umwandeln kann
  - Autonom
  - Interdisziplinär besetzen
  - Selbstorganisiert
  - Allgemein anerkannte Teamgröße: 7 +- 2
  - Vollzeitteammitglieder
  - Arbeitsplätze in unmittelbarer Nähe
  - Einer für alle, alle für einen
  - Visueller Arbeitsplatz

Scrum
Rolle: Scrum-Team

Fachhochschule Dortmund
University of Applied Sciences

- In Scrum wird **nicht** festgelegt
  - Wie die Arbeit unter den Softwareentwicklern aufgeteilt wird und
  - wie das Team sich organisiert
- Es wird großer Wert auf die **Eigenverantwortlichkeit und Kreativität** der Softwareentwickler gelegt
- Erwartung: Die Softwareentwickler haben das entsprechende Know-how und die nötige Erfahrung und sind somit für den Sprint und das Sprintergebnis verantwortlich

Scrum
Rolle: Scrum Master

Fachhochschule Dortmund
University of Applied Sciences

- Agiert als **Coach und Change-Agent**
- Hilft dem Team und dem Unternehmen **Scrum richtig einzusetzen**
- Soll den Prozess, die neuen Denk- und Verhaltensweisen etablieren
- **Jedes Team** hat **einen Scrum-Master**
- Scrum-Master hat seinen Schreibtisch im Teambereich und arbeitet eng beim Team

Scrum
Rolle: Scrum Master

- Scrum etablieren
- Das Team unterstützen
- Direkte Zusammenarbeit von Product Owner und Team sicherstellen
- Hindernisse beseitigen
- Entwicklungspraktiken verbessern helfen
- „Führen durch Dienen“ (servant leadership)
  - Vorbildfunktion
  - Keine Autorität bezüglich der Arbeitsorganisation im Team
  - Keine Zuweisung von Aufgaben an Teammitglieder

Scrum
Rolle: Scrum Master

- Eigenschaften eines Scrum Masters
  - Offen, kommunikativ, reflektiert, eloquent
  - Guter Zuhörer, Rückgrat
  - Moderation
  - Coaching
  - Konfliktmanagement
  - Erfahrung in der Softwareentwicklung

Scrum
Rolle: Scrum Master

Fachhochschule Dortmund
University of Applied Sciences

- Scrum-Master sollte idealerweise vom Team bestimmt werden
  - Das ist häufig unpraktikabel
  - Daher sollte der Entwicklungsleiter bei der Besetzung der Scrum-Master-Rolle mithelfen
- Wird der Scrum-Master in ihrem Unternehmen vom Management besetzt, sollten Sie Formen möglicher Mitbestimmung finden
  - Selbstnominierung der Scrum-Master-Kandidaten
  - Teamabstimmung
  - Mehrheitsentscheid

Scrum
Aufgabenbereiche

Aufgabenbereich	Scrum-Rollen
Scope-Management	Product Owner für die Produktversion Team für die Sprints
Zeitmanagement	Product Owner für den Releaseplan Team für das Sprint Backlog
Kostenmanagement	Product Owner
Kommunikationsmanagement	Product Owner für das Release-Reporting Team für die Berichterstattung im Sprint
Risikomanagement	Product Owner mit Input vom Team
Qualitätsmanagement	Product Owner für die Produktleistungsmerkmale Team für qualitätssichernde Maßnahmen ScrumMaster für die richtige Anwendung des Prozesses
Lieferantenmanagement	Product Owner und Team
Personalmanagement	Management für die Bereitstellung der Mitarbeiter Team für Produktivität und kontinuierliche Prozess-verbesserung

Scrum

- Inhalte und Ablauf
- Rollen
- User Stories und Anforderungsspezifikation
- Projektfortschrittskontrolle
- Klassifikation

Scrum
Anforderungsspezifikation

- In Scrum werden die **Anforderungen** mithilfe von sogenannten **User Stories** erhoben
- User Stories werden im Rahmen von Agilen Modellen/Scrum eingesetzt, um die Anforderungen zu bestimmen
- Die User Stories werden hierbei in der Begriffswelt des Anwenders mithilfe von **natürlicher Sprache** dokumentiert
- Eine User Story ist kurz gehalten und umfasst oftmals nur ein bis zwei Sätze

Scrum
Anforderungsspezifikation

- Ein wichtiger Grundsatz bei User Stories ist, dass sie **aus der Sicht von Anwender:innen** geschrieben werden
- User Stories werden **frei von technischen Begrifflichkeiten** („Entwickler-Jargon“) gehalten
- Jeder, der am Projekt beteiligt ist, also auch jeder Anwender, sollte User Stories **verstehen** können
- Hingegen sollte das Entwicklungsteam tolerant gegenüber der Fachsprache der Anwender sein

Scrum
Abgrenzung Use Case

Fachhochschule Dortmund
University of Applied Sciences

- Anwendungsfälle (Use Cases) aus der klassischen, nicht-agilen, objektorientierten Softwareentwicklung sind durchaus ähnlich zu User Stories
  - sie stellen ebenfalls Anforderungen in der Sprache des Anwenders und im Kontext des Endprodukts der Softwareentwicklung dar
- Bei einem Use Case werden jedoch alle Erfolgs- und Misserfolgsszenarien bei der möglichen Erreichung eines fachlich relevanten Ziels gebündelt dargestellt
- Eine User Story stellt hingegen eine fachlich motivierte Anforderung dar, die von einem Anwender zwar als erfolgreich bzw. nicht erfolgreich umgesetzt beurteilt werden kann
- Anwender wird nur über eine User Story allein keine fachlichen Prozesse vollständig bearbeiten bzw. entsprechende Ziele erreichen können

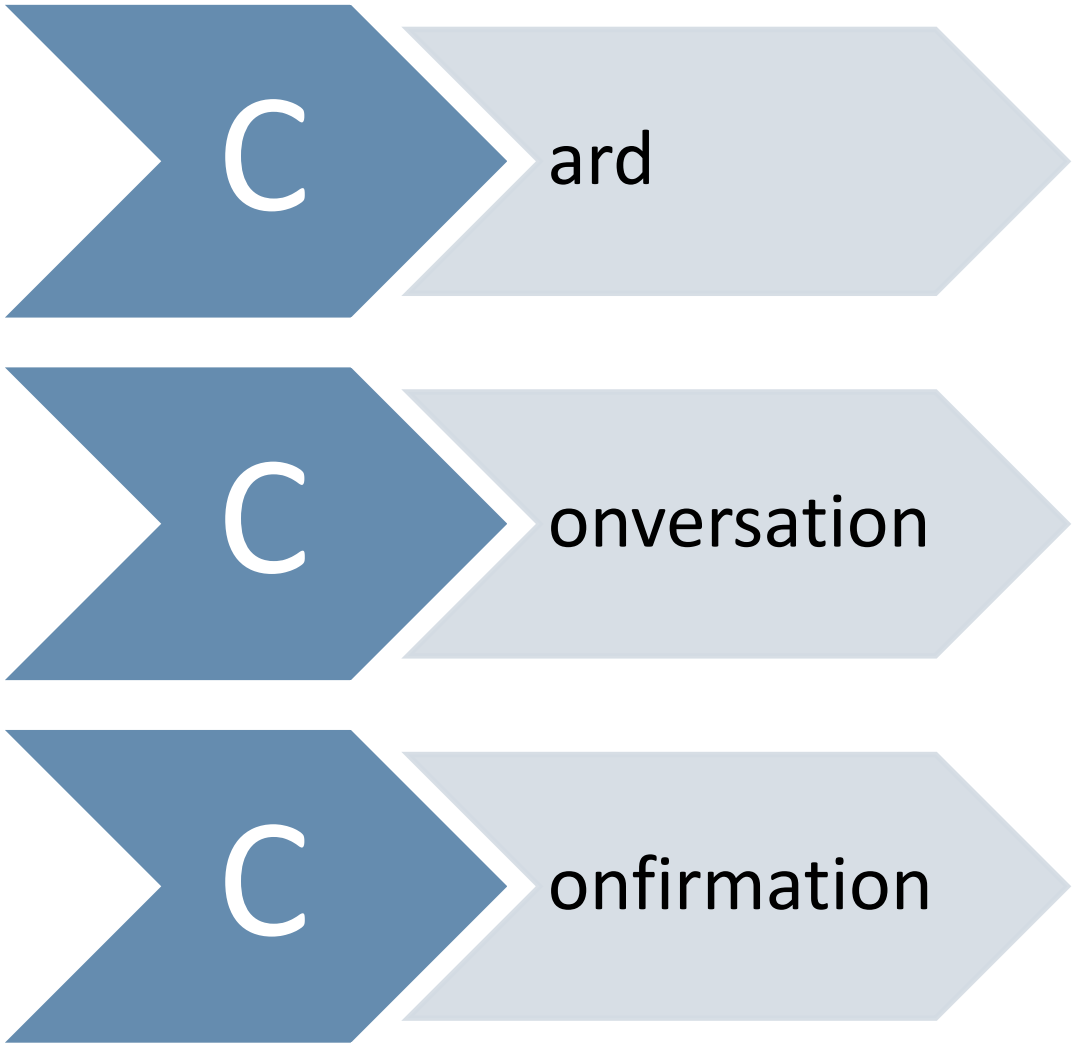
Scrum
Abgrenzung Use Case

Fachhochschule Dortmund
University of Applied Sciences

- Somit kann **EIN Use Case** den Kontext bzw. Zusammenhang für **VIELE User Stories** bilden
  - Eine User Story ist die Überschrift eines konkreten Szenarios
  - Ein Use Case beinhaltet mehrere dieser Szenarien

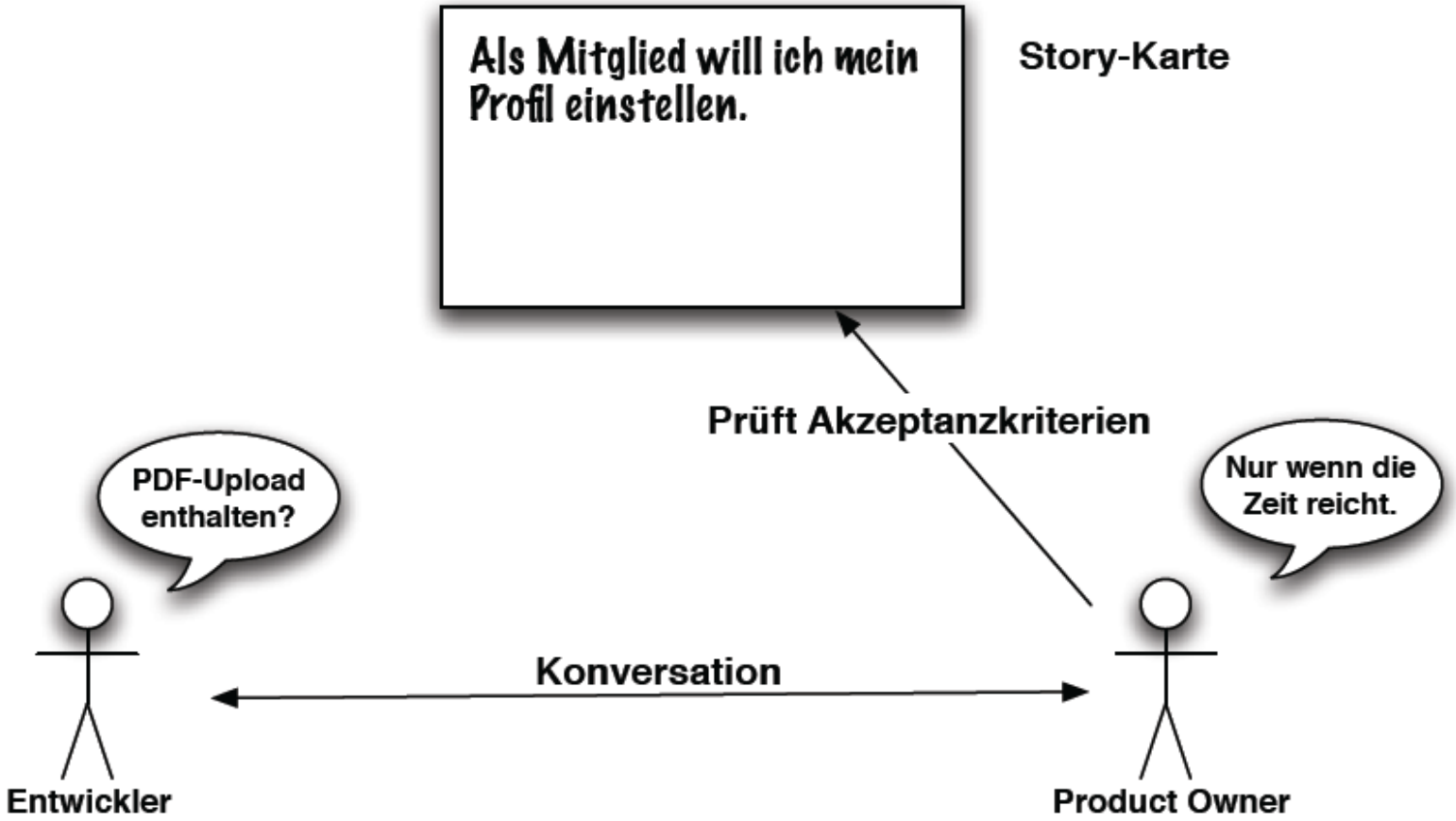
Scrum
Bestandteile einer User Story (CCC)

Fachhochschule Dortmund
University of Applied Sciences



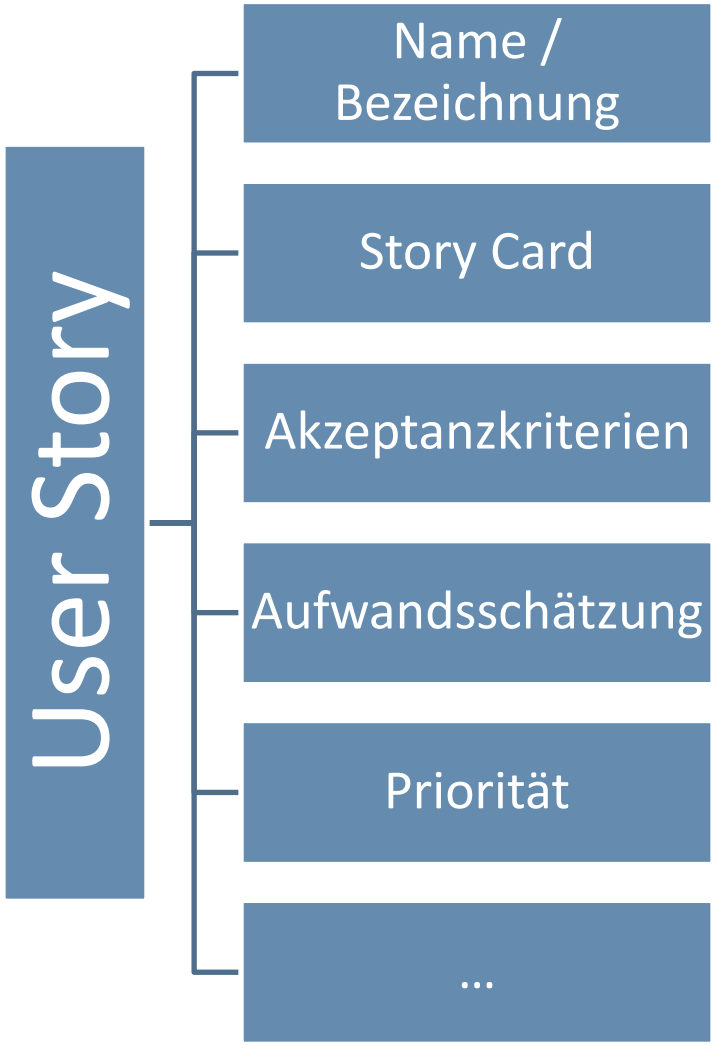
Scrum
Bestandteile einer User Story (CCC)

Fachhochschule Dortmund
University of Applied Sciences



Scrum
Bestandteile einer User Story im Backlog

Fachhochschule Dortmund
University of Applied Sciences



Scrum
Name einer User Story

- Eine **kurze, prägnante Benennung**, die sich die Teammitglieder und die anderen Beteiligten gut merken können.
- Am besten ist sie so, dass jeder etwas damit anfangen kann, wenn jemand sagt “Wir sollten noch xy umsetzen”.
- Soll die Anmeldung an einem Softwaresystem beschrieben werden, kann die Bezeichnung “Anmeldung” ausreichend sein, wenn möglichst alle im Team verstehen, was damit gemeint ist.
- Üblicherweise sichtbarer Teil im Backlog

Scrum
Story Card

- Template
  - Als [**Benutzer**] möchte ich [**Ziel**], sodass [**Grund für das Ziel**]
- Beispiele
  - Als Kunde möchte ich mich an der Oberfläche anmelden, sodass ich Zugriff auf meinen Warenkorb erhalte.
- Tipps
  - Der Grund für ein Ziel ist wichtig, da er den Wert einer User Story darstellt. Sollte der Grund fehlen oder keinen Informationsgehalt besitzen, versuchen Sie, die User Story nach dem umgekehrten Muster zu erstellen: Um [Wert/Grund], möchte ich als [Benutzer], dass [Ziel].

Scrum
Story Card

Fachhochschule Dortmund
University of Applied Sciences

- Es ist möglichst klar darzustellen, von **welchem Autor** mit **welcher Rolle** bzw. Anwendertyp die Anforderung überhaupt kommt
- Am häufigsten anzutreffen sind User Stories mit dem Anwendertyp “Benutzer” oder „Sachbearbeiter in der Abteilung xy“
- Es sollte jedoch deutlich spezifischer zum Ausdruck gebracht werden, um welchen Benutzer es sich handelt
- „Als Benutzer mit einem Facebook-Account” oder „als SAP R/3-Nutzer“
- Das ist schon ein ganzes Stück spezifischer und hilft allen Beteiligten einzuschätzen, wie groß die Nutzergruppe ist und woher ihr Wunsch stammt

Scrum
Story Card

- **Nutzen**: An dieser Stelle wird der intendierte Nutzen, der erreicht werden soll, beschrieben
- Die Idee ist dabei, dass die anschließend beschriebene Funktionalität aus dem Nutzenstreben abgeleitet sein muss
- Ein Benutzer will sich z.B. nicht als Selbstzweck bei einer Softwareanwendung anmelden, sondern macht dies nur, um die Kernfunktionen der Software anschließend nutzen zu können
- Zudem sollte beschrieben werden, welche Funktionalität zur Realisierung des Nutzens implementiert werden soll

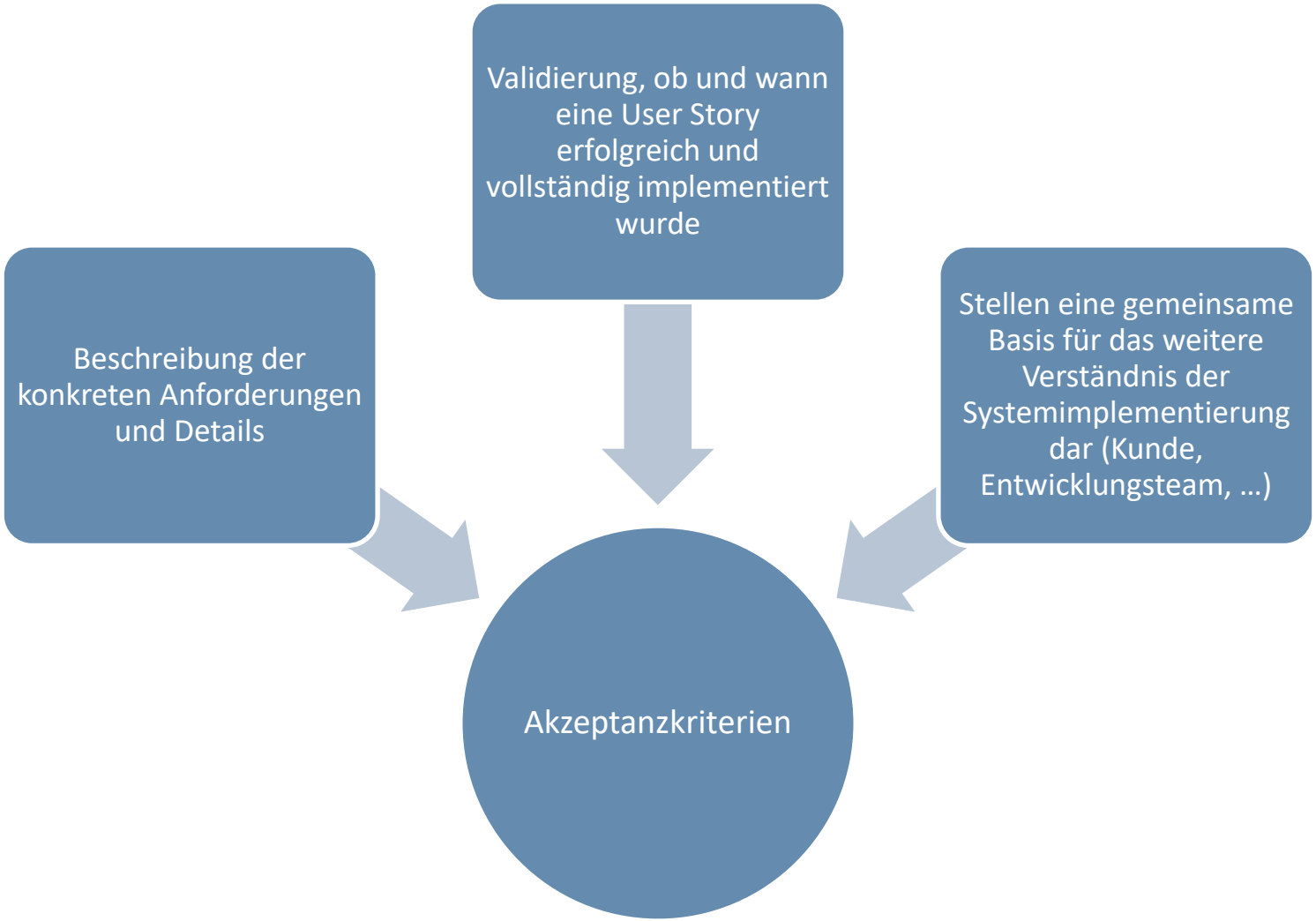
Scrum
Story Card

Fachhochschule Dortmund
University of Applied Sciences

- User Stories können einen sehr unterschiedlichen Grad an **Genauigkeit** annehmen
- Dies hängt auch damit zusammen, wie nah oder weit die **Umsetzung** bzw. **Reifegrad der User Story** der Anforderung ist
- Es kann sein, dass:
  - die Lösung und Umsetzung der User Story noch weitgehend offen gehalten werden soll, oder
  - es ist schon eine Entscheidung für eine konkretere Umsetzungsvariante (Entwurfsentscheidung) gefallen.

Scrum
Akzeptanzkriterien - Ziele

Fachhochschule Dortmund
University of Applied Sciences



Scrum
Akzeptanzkriterien - Charakteristika

Fachhochschule Dortmund
University of Applied Sciences

- **Bindeglied** zwischen User Story und Testfällen
- Mögliches Verfahren zur Identifikation von Akzeptanzkriterien: **W-Fragen**
  - Wer, was, wann, wo, welche, wie, wie viele, …
  - Bezug zu Substantiven, Verben, Adjektiven der Story Card

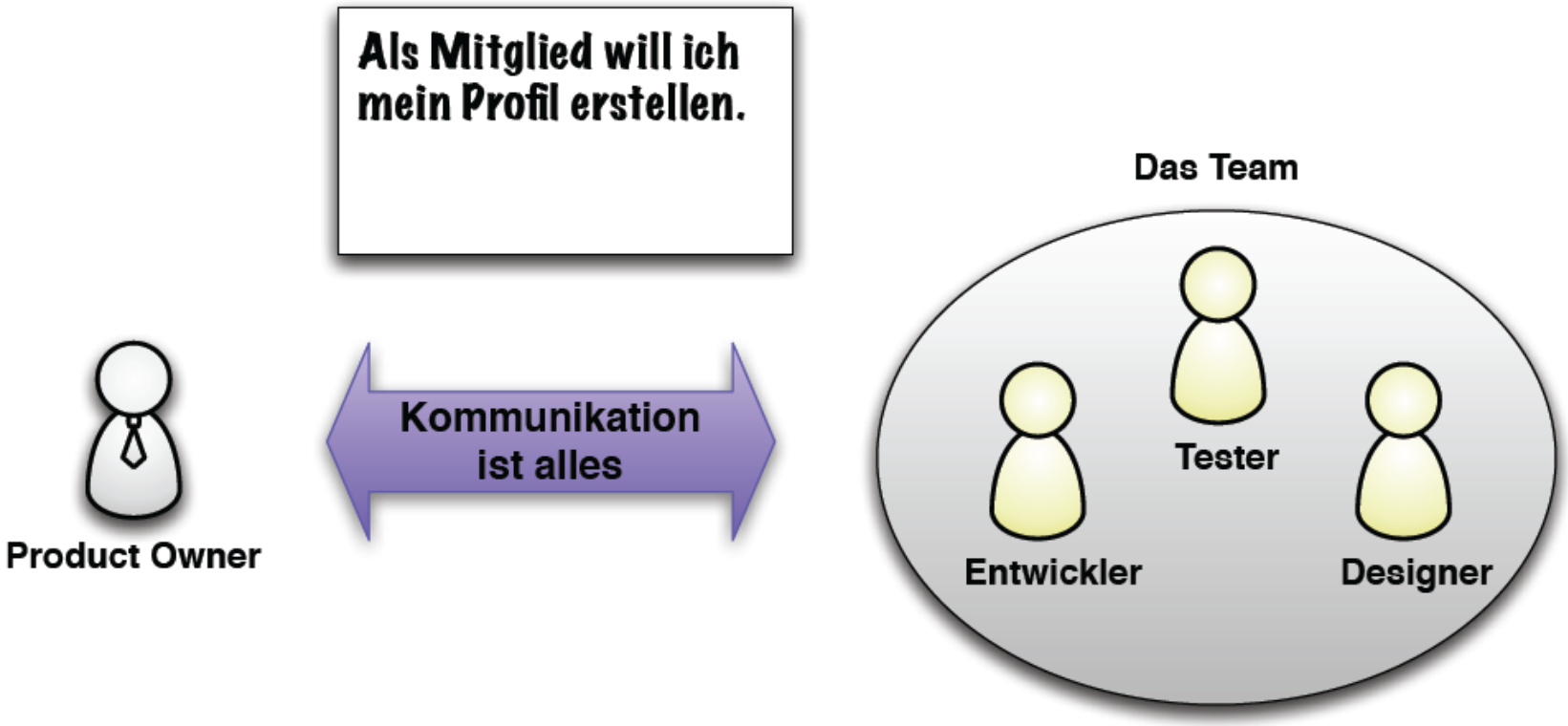
Scrum
Akzeptanzkriterien

- Wie kann festgestellt werden, ob die User Story entsprechend den Anforderungen umgesetzt wurde?
- Die Akzeptanzkriterien spielen dabei eine besonders wichtige Rolle
- Sie beantworten die Frage, was getestet werden soll, und ob die User Story erfolgreich umgesetzt wurde.

Scrum
Akzeptanzkriterien - Beipsiele

- Beispiel
  - Story Card
    - Als Benutzer möchte ich mich an der Webseite anmelden, damit ich Zugriff auf meinen persönlichen Warenkorb erhalte.
  - Akzeptanzkriterien
    - Die Anmeldung erfolgt auf Basis des User-Credentials der Benutzerverwaltung.
    - Der Benutzer erhält zwei Textfelder auf der Login-Seite zur Eingabe eines Benutzernamens und eines Passworts.
    - Die Eingabe auf der Login-Seite kann mittels Tastatureingabe (Enter) oder Button abgesendet werden.
    - Die Zeichen des Passworts werden im Textfeld nicht im Klartext angezeigt.
    - Die Eingabe wird mit den verwalteten Benutzeridentitäten validiert.
    - Bei erfolgreicher Validierung wird mit aktivem Benutzerkontext auf die Startseite weitergeleitet.
    - Bei nicht erfolgreicher Validierung wird eine Fehlermeldung angezeigt.

Scrum
Kommunikation

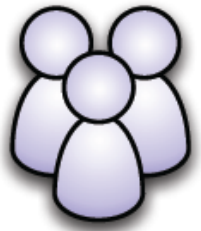


Scrum
Kommunikation

- Zunächst kann eine gewisse Unsicherheit vorhanden sein, ob das Entwicklungsteam die User Stories in der Form akzeptiert
- Hier ist es aber wie mit vielen anderen Dingen in der agilen Softwareentwicklung:
  - Es gehört Mut dazu, Dinge auszuprobieren und sich einzugestehen, dass am Anfang nicht alles rund läuft.
  - Nach ein paar Wochen der Arbeit mit User Stories haben dann alle Beteiligten eine gemeinsame Vorstellung davon entwickelt, wie umfangreich, detailliert und sauber die User Stories ausgearbeitet werden müssen.
  - Das funktioniert am besten, wenn darüber hinaus regelmäßige Sitzungen durchgeführt werden, in denen mit dem Entwicklungsteam und Anwendern zusammen an den Stories gearbeitet wird.

Scrum
Schreiben von User Stories

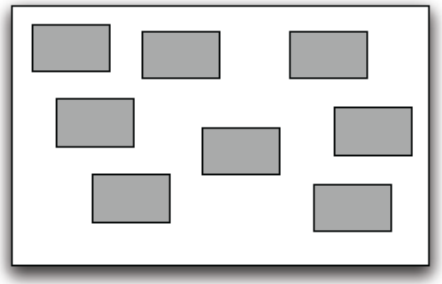
Anforderungs-Workshop



Kunde

Productmanagement

Product Owner



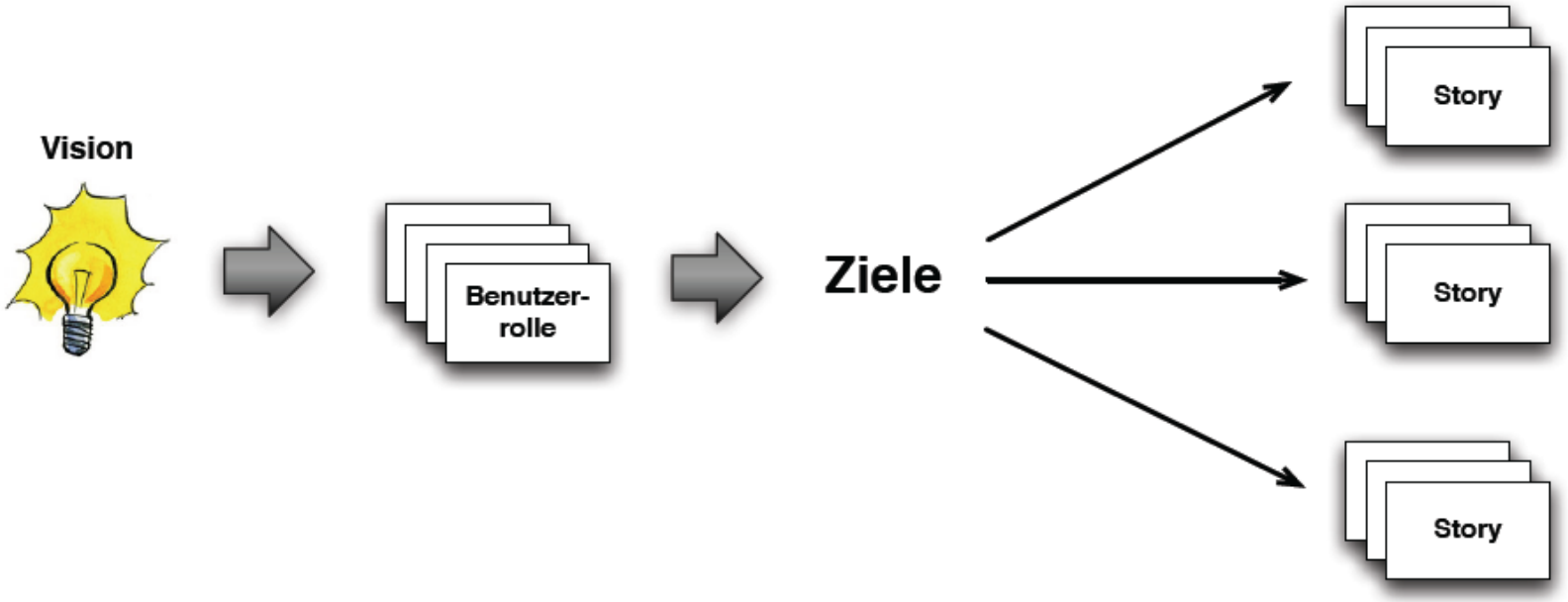
Marketing

Team

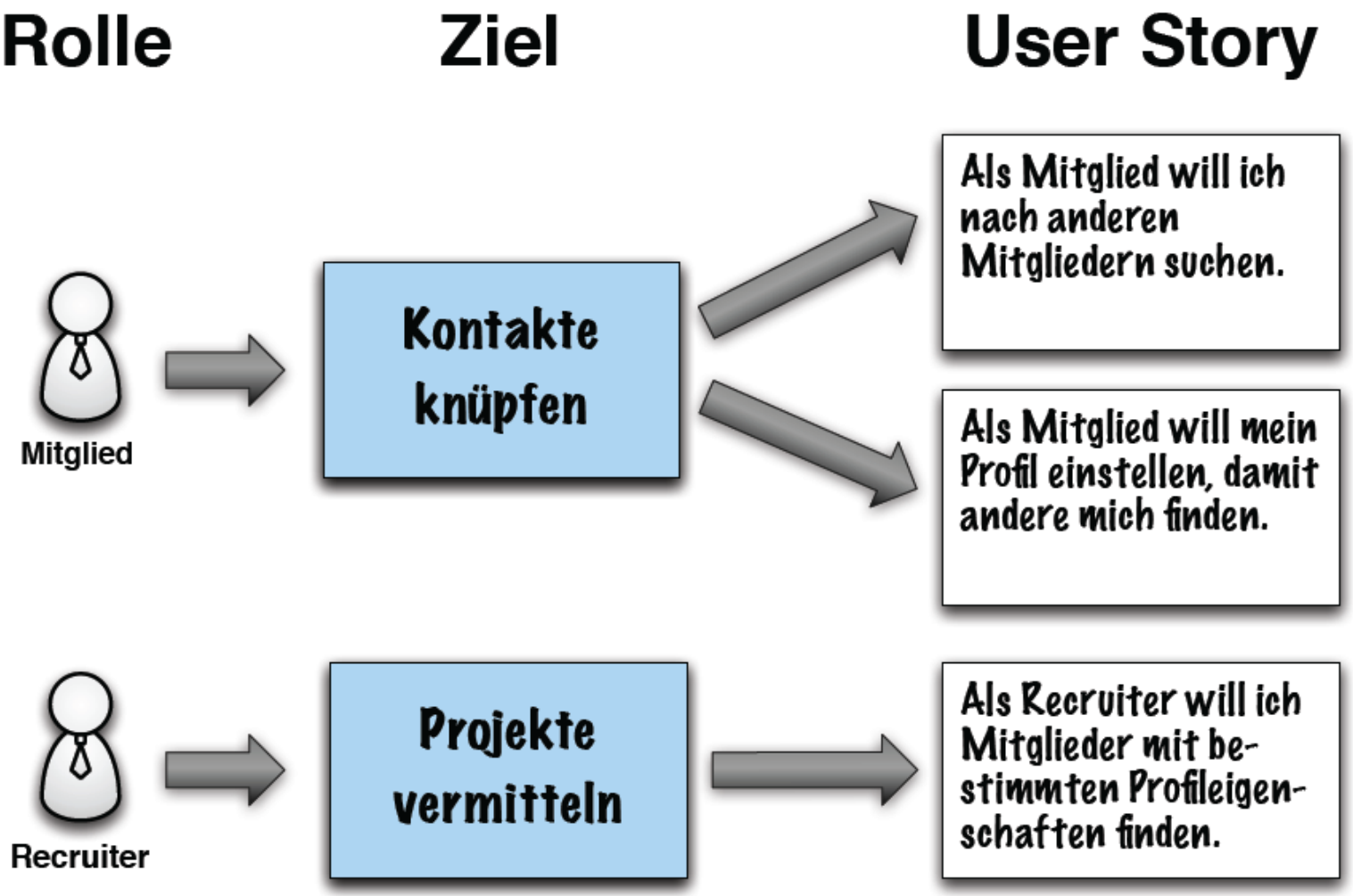
Geschäftsführer

Scrum
Schreiben von User Stories

Fachhochschule Dortmund
University of Applied Sciences

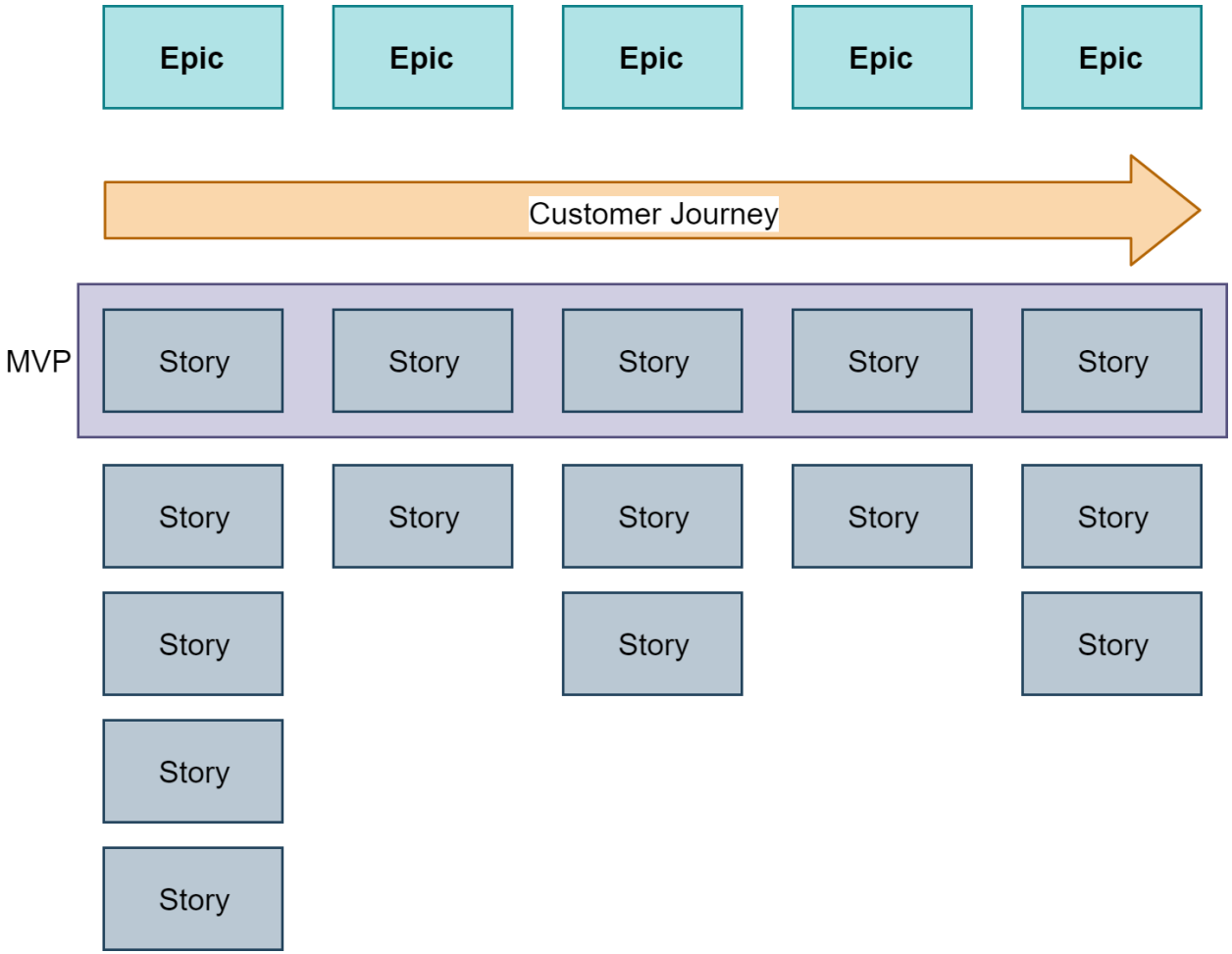


Scrum
Schreiben von User Stories



Scrum
User Stories - Story Map

Fachhochschule Dortmund
University of Applied Sciences



Scrum
User Stories - INVEST

Fachhochschule Dortmund
University of Applied Sciences

- Gute User Stories berücksichtigen das INVEST Prinzip
  - **I** ndependent (unabhängig)
  - **N** egotiable (verhandelbar
  - **V** aluable (werthaltig)
  - **E** stimatable (schätzbar)
  - **S** mall (klein)
  - **T** estable (überprüfbar/testbar)

Scrum
User Stories - Checkliste

1. Um wen geht es? Wessen Problem wird gelöst? Welche Rolle hat der Autor inne?
2. Haben wir das Problem hinterfragt und verstanden oder die Anforderung einfach vom Kunden übernommen?
3. Wird bereits eine Lösung vorgegeben oder wirklich nur das Problem beschrieben?
4. Was ist der Nutzen dieser Anforderung? Passt er auf die genannte Rolle?

Scrum
Aufwandsschätzung und Metriken

- Zeitbasierte Schätzung
  - Aufwandsschätzung in Zeiteinheiten, üblicherweise Personentage (PT) oder Stunden
  - Absolute Schätzung
  - Naheliegendes klassisches Verfahren aus dem Projektmanagement

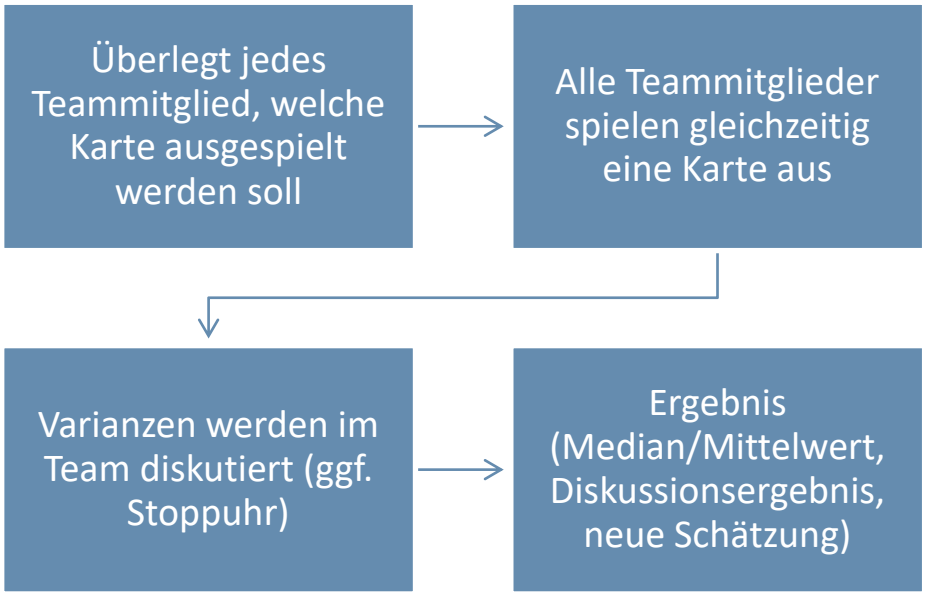
- T-Shirt-Größen
  - Wertebereich: Kleidergrößen
    - XS / S / M / L / XL / XXL
  - Losgelöst von Zahlenwerten
  - Verhältnis zwischen den Werten oft unklar bzw. bedarf häufig einer impliziten oder expliziten Umrechnung (Abstände, Velocity)

Scrum
Aufwandsschätzung und Metriken

- Story Points
  - Relativer Aufwand einer Story bezogen auf andere Stories
  - Berücksichtigt geschätzte Komplexität und Schwierigkeit einer Story
  - Bsp. Für Wertebereich: Fibonacci
    - 0 / 1 / 2 / 3 / 5 / 8 / 13 / 20 / 40 / 100
  - Ausrichtung an Basis-Stories als „Kammerton“ und Richtwert
  - Vorteile
    - Keine emotionale Bindung an Zeitpunkte / Datum
    - Fokus auf Komplexität statt auf Bearbeitungsdauer
    - Bestimmung der Velocity auf Basis von Punkten
    - Relativer Aufwand lässt sich leichter schätzen als Dauer
    - Schätzung als Bearbeitungsdauer hängt stark von der Person ab, die daran arbeitet

Scrum
Aufwandsschätzung und Metriken

- Planning Poker
  - Spielkarten mit Schätzwerten gemäß Wertebereich
    - Bsp.: Je Karte ist eine Zahl der Fibonacci-Reihe abgebildet
  - Jedes Teammitglied erhält einen vollständigen Kartensatz
  - Nachdem die User Story vorgestellt wurde:



Scrum

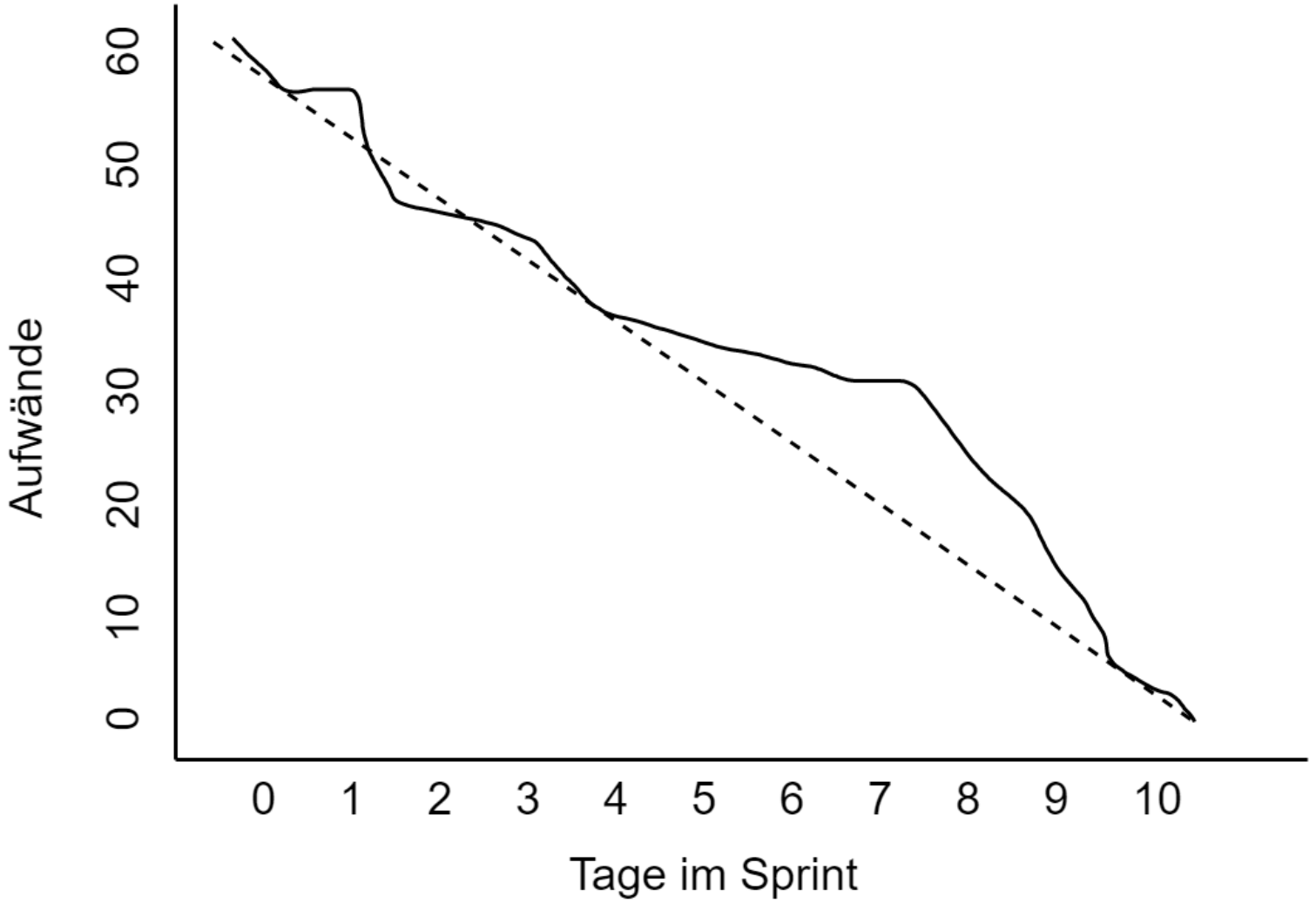
- Inhalte und Ablauf
- Rollen
- User Stories und Anforderungsspezifikation
- Projektfortschrittskontrolle
- Klassifikation

Scrum
Projektfortschrittskontrolle

- Teammitglieder protokollieren jeden Tag ihre geschätzten (Rest-) Aufwände, die bis zur Erledigung einer User Story noch benötigt werden
- Restaufwände werden im sogenannten Sprint Burndown Chart hinterlegt:
  - Horizontale Achse: Tage im Sprint
  - Vertikale Achse: Kumulierte Restaufwände aller User Stories des Sprint Backlogs
- Restaufwand muss gegen Null gehen, wenn Termin gehalten werden soll
- Wird Produkt nach der Prognose früher fertig, dann wählt der Product Owner weitere Anforderungen aus dem Product Backlog aus

Scrum
Sprint Burndown Chart

Fachhochschule Dortmund
University of Applied Sciences



Scrum
Velocity

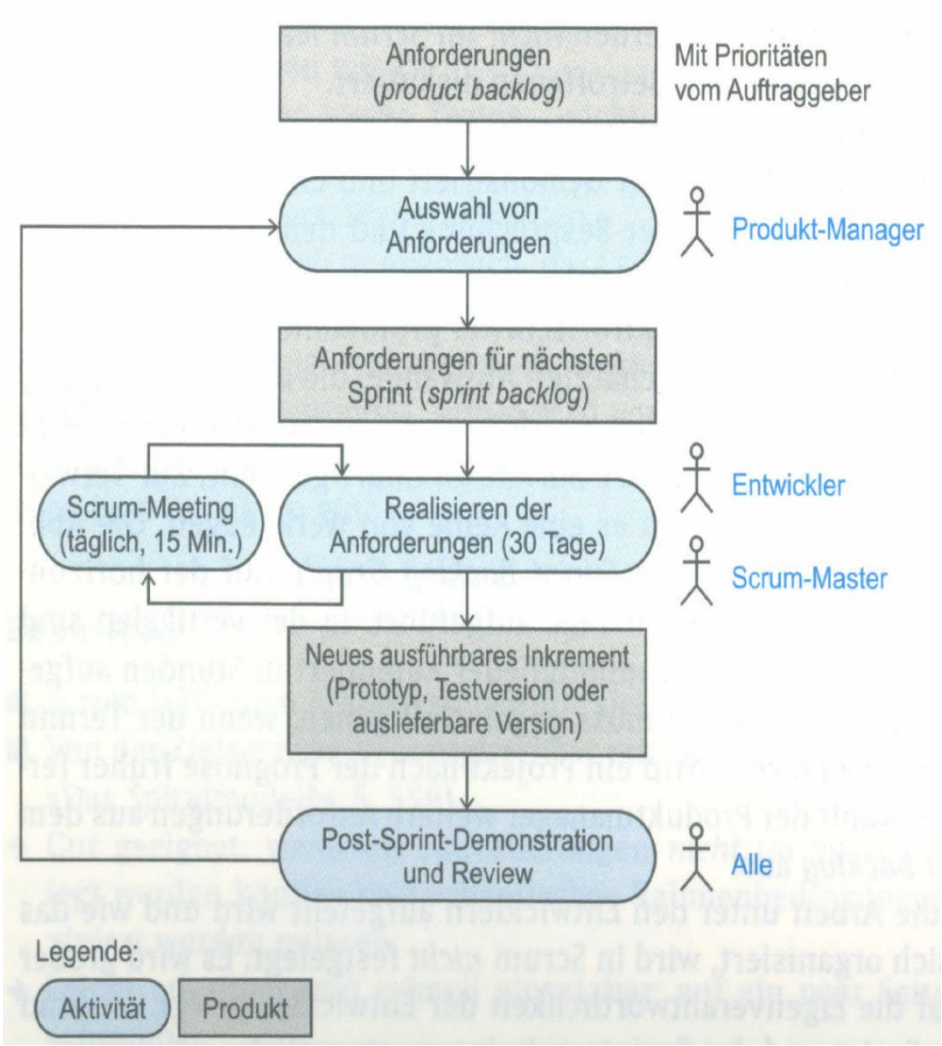
Fachhochschule Dortmund
University of Applied Sciences

- Die Velocity eines Teams gibt an, welchen **Arbeitswert** das Scrum Team in einem Entwicklungszyklus / Sprint bewältigen kann
- Die Velocity ist dabei ein nivellierter Wert auf Basis vergangener Sprints (bspw. durchschnittliche Velocity der letzten 3-4 Sprints)
- Zur Berechnung der Velocity werden nur vollständig erledigte User Stories berücksichtigt, nur teilweise oder gar nicht erledigte User Stories werden nicht bei der Berechnung der Velocity berücksichtigt. Beispiel:
  - Im Sprint sind Story A mit 5, Story B mit 8 und Story C mit 13 Story Points eingeplant
  - Am Ende des Sprints sind Story A und C vollständig bearbeitet, Story B ist jedoch nur zu ca. 80% fertiggestellt
  - Die Velocity wird daher auf Basis der Stories A und C berechnet und beträgt 18 Story Points
  - Die aktuelle Velocity kann als Durchschnittswert mehrerer vergangener Sprints berechnet werden

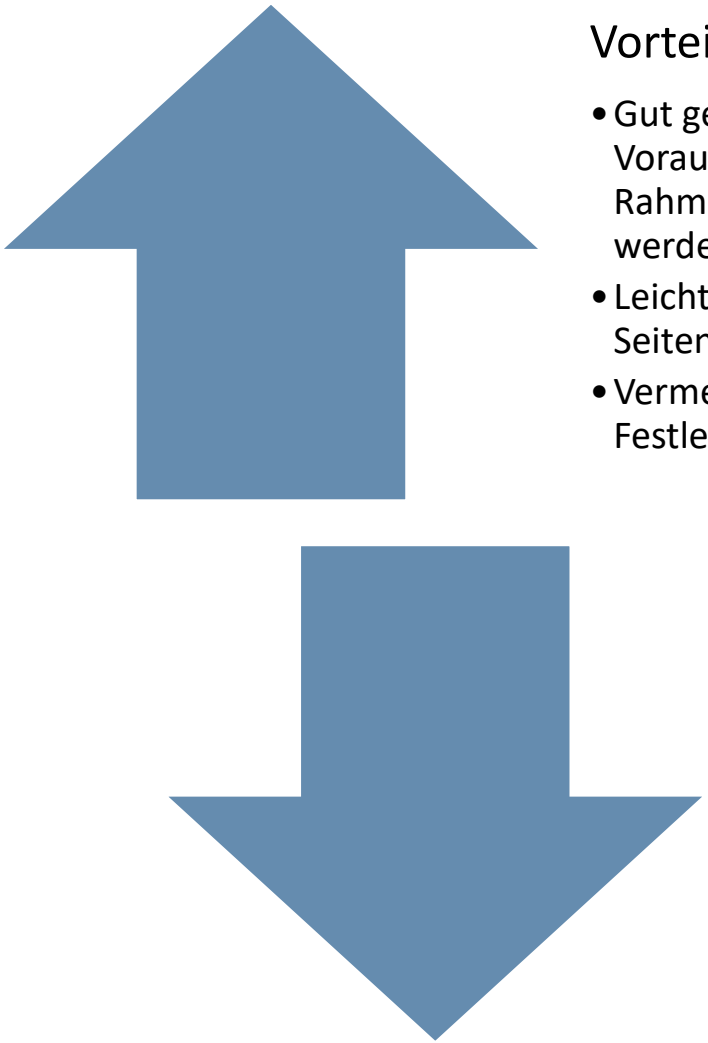
Scrum

- Inhalte und Ablauf
- Rollen
- User Stories und Anforderungsspezifikation
- Projektfortschrittskontrolle
- Klassifikation

Scrum
Normierte Darstellung



Scrum
Vor- und Nachteile



Vorteile

- Gut geeignet, wenn die Anforderungen nicht im Voraus festgelegt werden können und „chaotische“ Rahmenbedingungen antizipiert werden müssen
- Leicht erlernbar und schnell einsetzbar, auf ein Paar Seiten beschreibbar
- Vermeidet für den Auftragnehmer die frühzeitige Festlegung eines festen Preises

Nachteile

- Erfolg hängt stark von den Teammitgliedern, ihren Kenntnissen und ihrer Persönlichkeit ab
- Keinerlei Hilfestellungen für die Entwicklung von Software
- Problematisch bei Festpreisprojekten

Scrum
Klassifikation

- Entwicklungsphilosophie und Prozessmodell
- Art und Weise der Qualitätskontrolle legt das Team fest
- Grobgranularer Rahmen mit feingranularen Festlegungen in einigen Details:
  - X-tägiger Sprint
  - 15-minütige tägliche Besprechung
- Relativ allgemeines Modell für die Entwicklung eines komplexen Produkts in einem kleinen Team

Agenda

- Klassische Vorgehensmodelle
  - Wasserfallmodell
  - Nebenläufiges Modell
  - Spiralmodell
  - Prototyping
- Monumentale Vorgehensmodelle
  - V-Modell XT
  - Rational Unified Process (RUP)
- Agile Modelle
  - Scrum
  - Kanban (IT-Kanban)
  - Extreme Programming (XP)
  - Feature Driven Development
  - Adaptive Software Development
  - Crystal
  - Lean Software Development
  - **SAFe**

SAFe
Agile Skalierung: Warum?

SAFe
Agile Skalierung: Warum?

Fachhochschule Dortmund
University of Applied Sciences

Scaled Agile

Mehrere agile Teams arbeiten an **gemeinsamer Lösung** (value stream, solution)

Abhängigkeiten & Notwendige **Abstimmungsprozesse** auf verschiedenen Ebenen (andere Teams, Abteilungen, …)

Ziele

Optimierte **Synchronisierung** verschiedener Teams & Akteure für „**Alignment**“ von **Value Streams und Solutions**

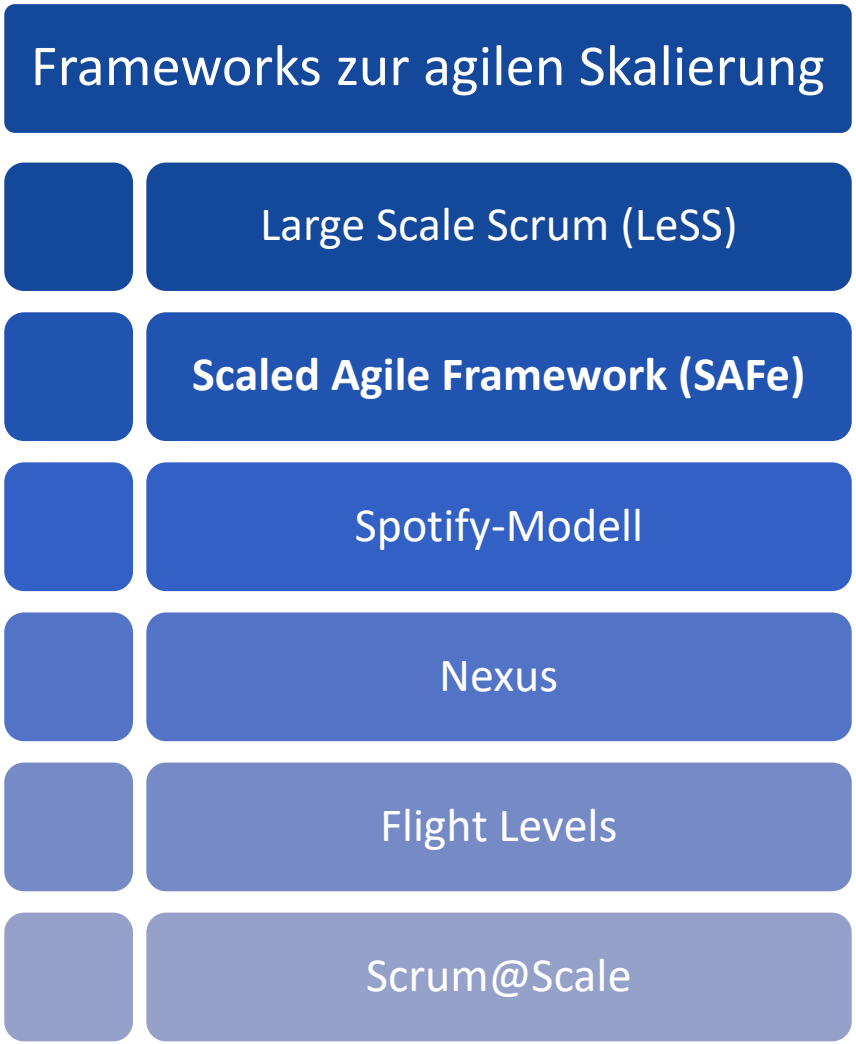
Autonome Teams & **Minimierung von Abhängigkeiten**

Methoden

Agile Werte und Prinzipien auf **allen Organisationsebenen**

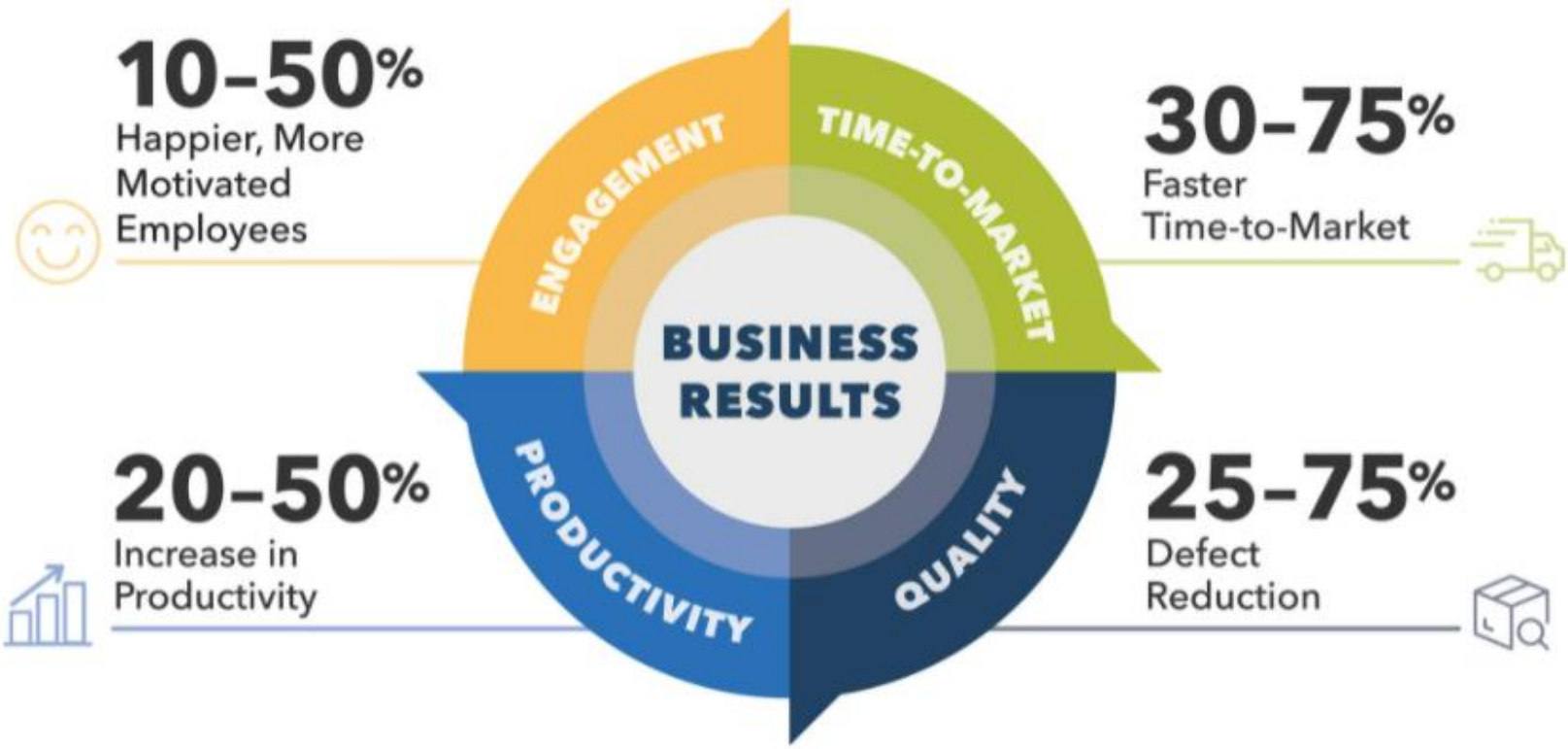
Methoden und Werkzeuge zur **Koordination** und **Synchronisierung mehrerer** agiler Teams, die gemeinsam an **Value Streams und Solutions** arbeiten

SAFe
Agile Skalierung: Frameworks



SAFe
Ziele und Benefits

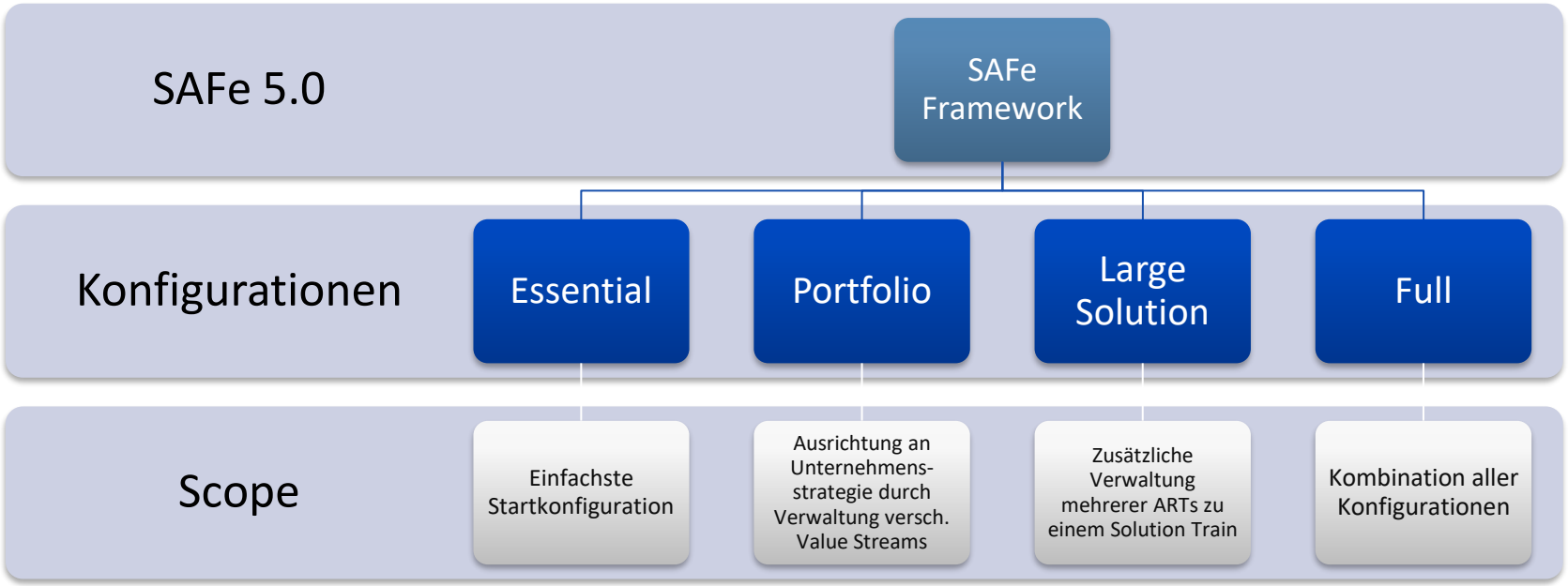
Fachhochschule Dortmund
University of Applied Sciences



Quelle: Richard Kastner, Dean Leffingwell: Safe 5.0 Distilled: Achieving Business Agility With the Scaled Agile Framework. Addison Wesley, 2020.

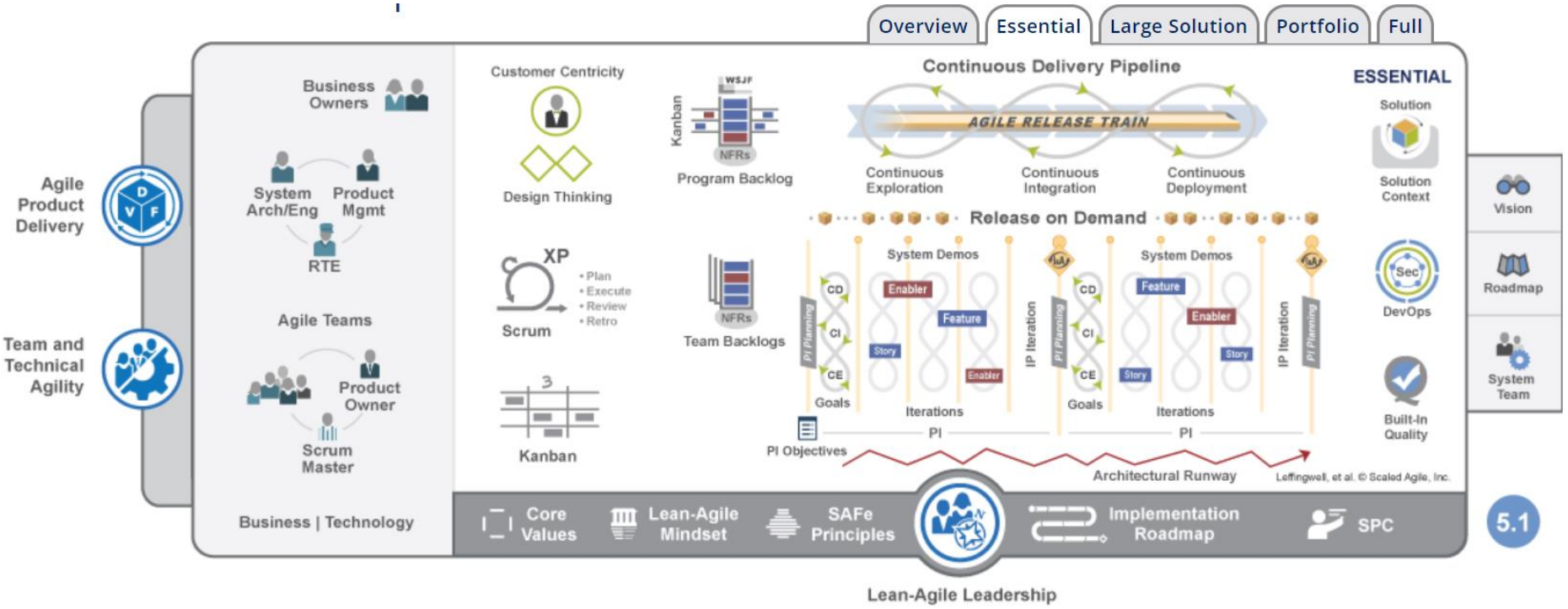
Agile Skalierung Frameworks

Fachhochschule Dortmund
University of Applied Sciences



SAFe Essential

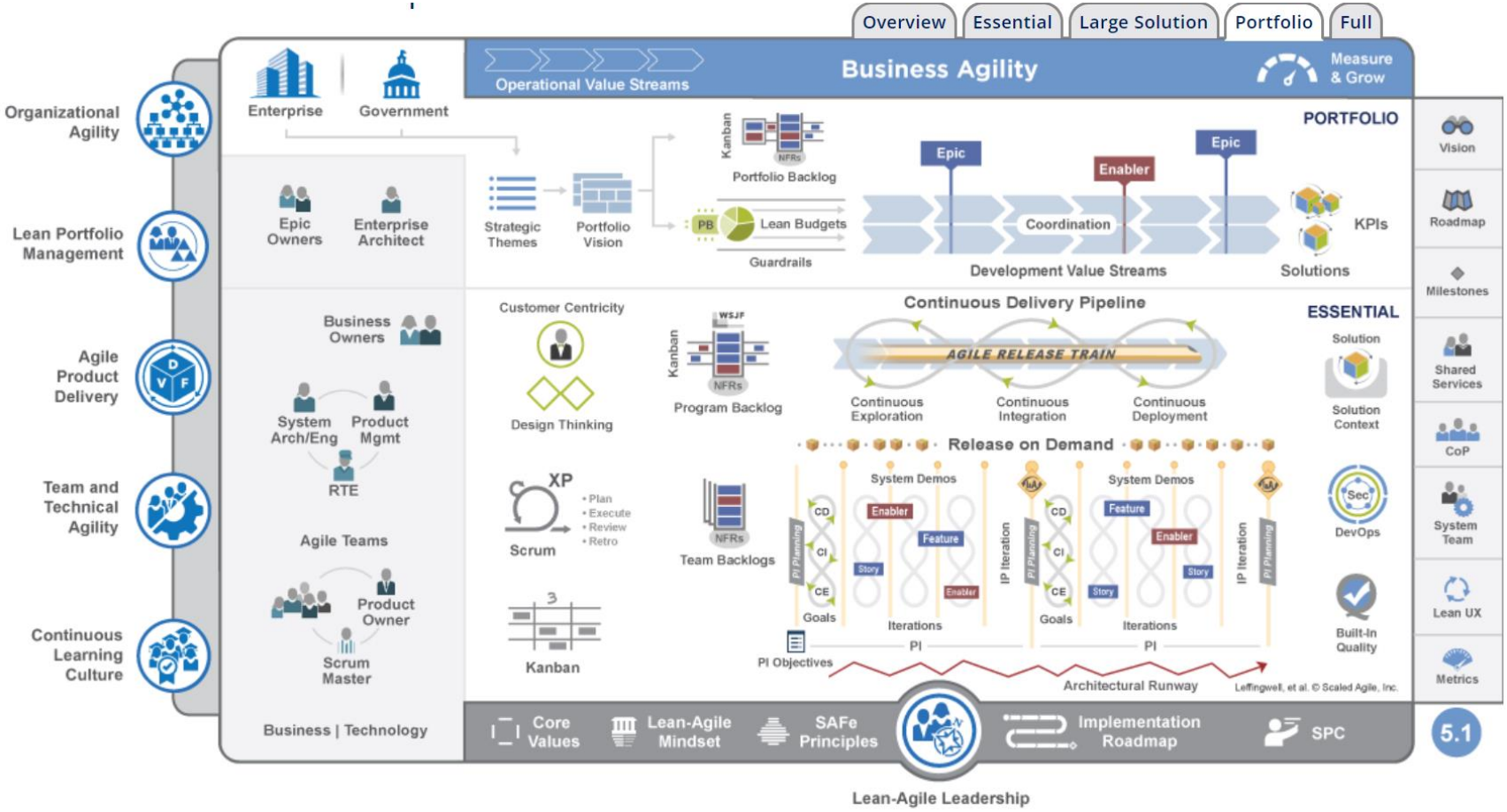
Fachhochschule Dortmund
University of Applied Sciences



Quelle: SAFe 5 for Lean Enterprises (scaledagileframework.com)

SAFe Portfolio

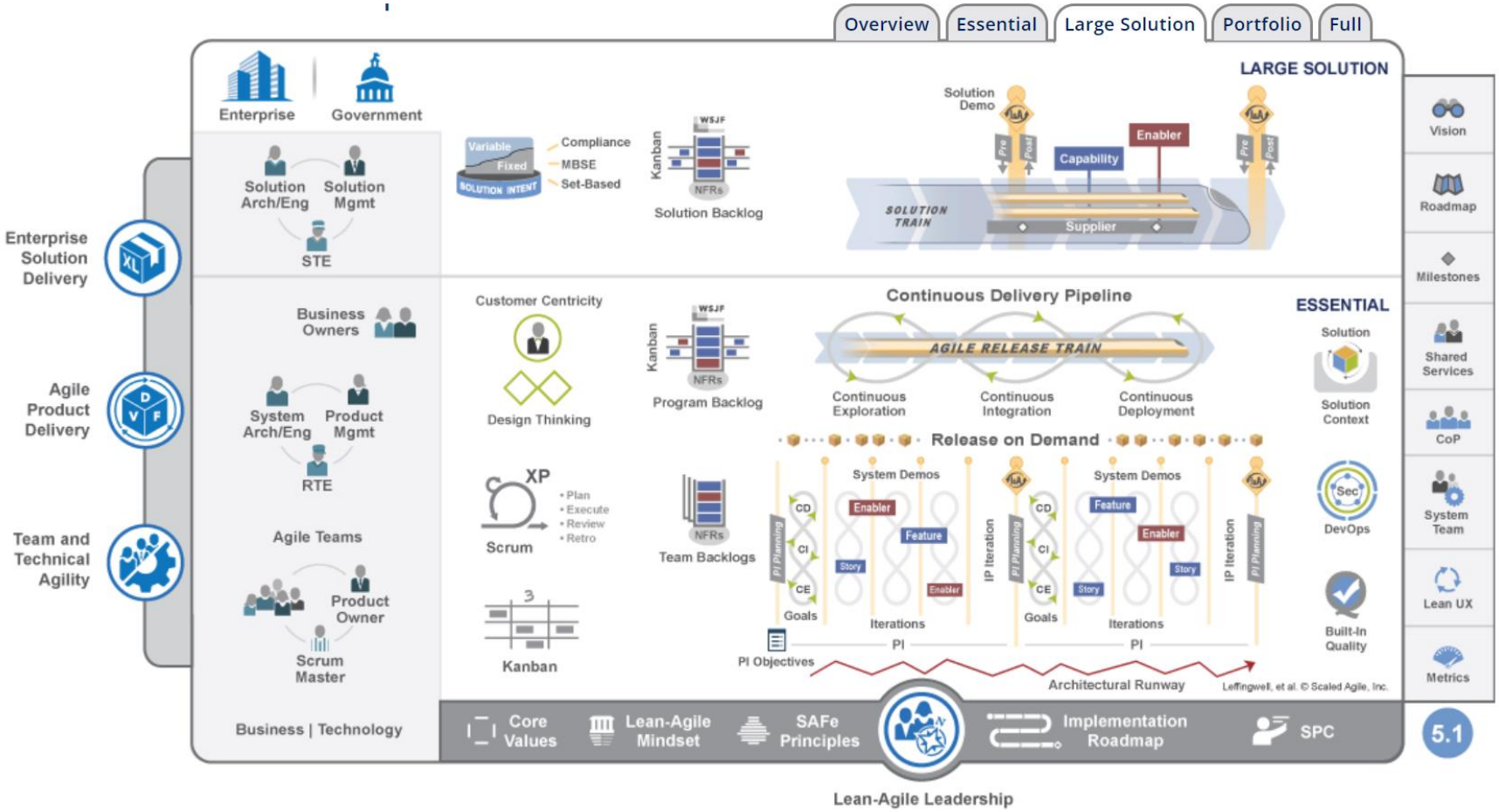
Fachhochschule Dortmund
University of Applied Sciences



Quelle: [SAFe 5 for Lean Enterprises (scaledagileframework.com)]

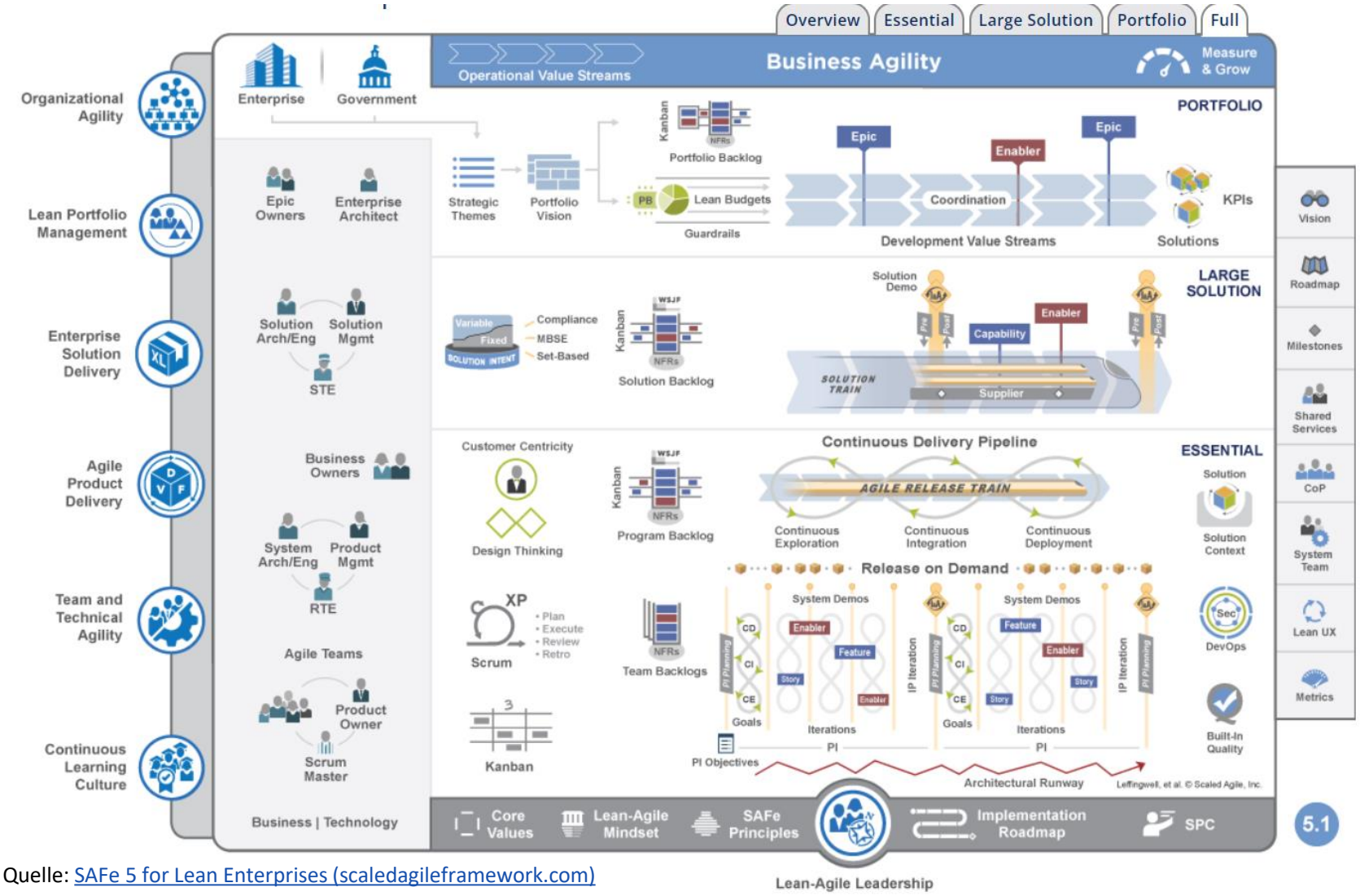
SAFe
Large Solution

Fachhochschule Dortmund
University of Applied Sciences



Quelle: [SAFe 5 for Lean Enterprises (scaledagileframework.com)]

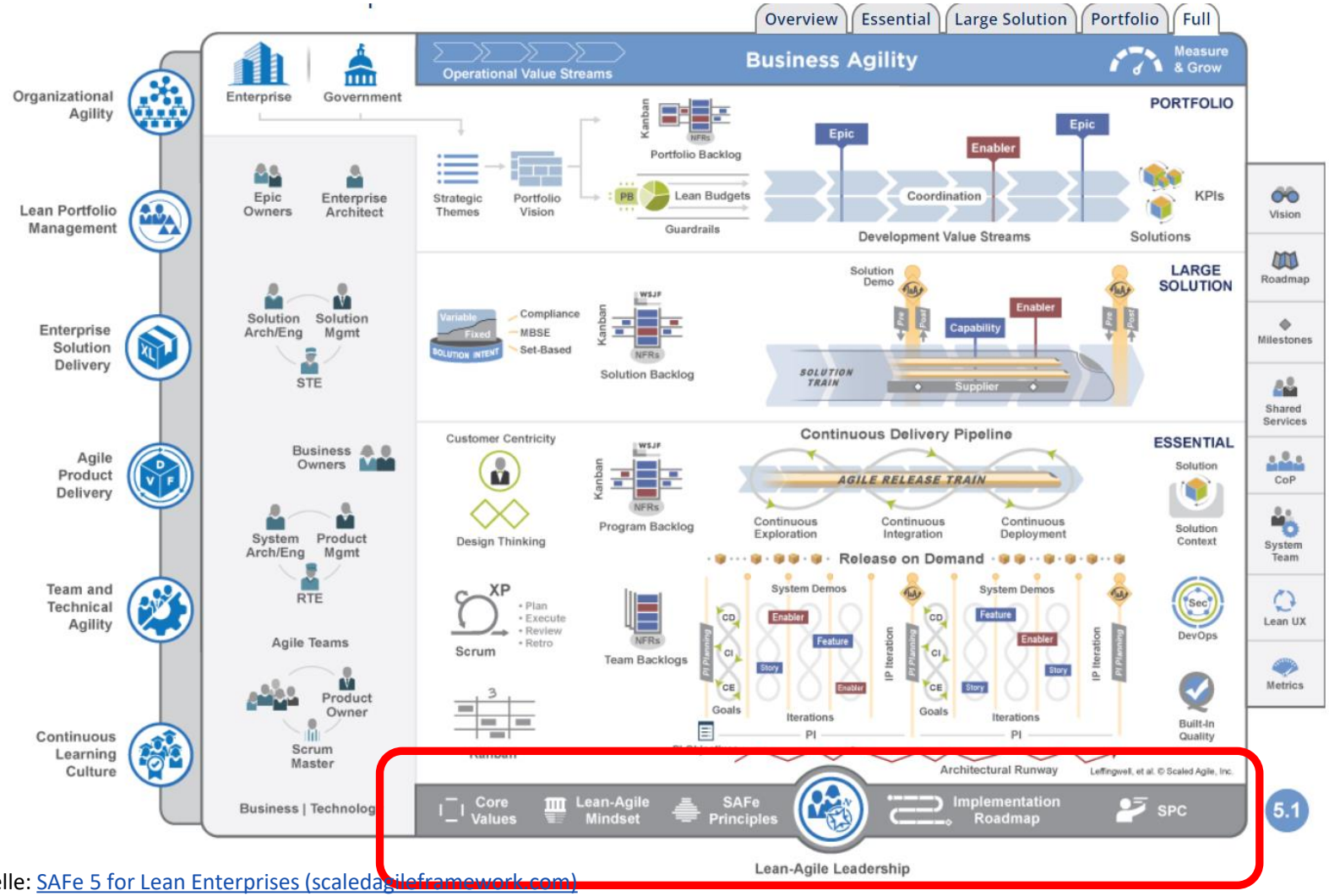
SAFe Full



Quelle: [SAFe 5 for Lean Enterprises (scaledagileframework.com)]

SAFe Foundation

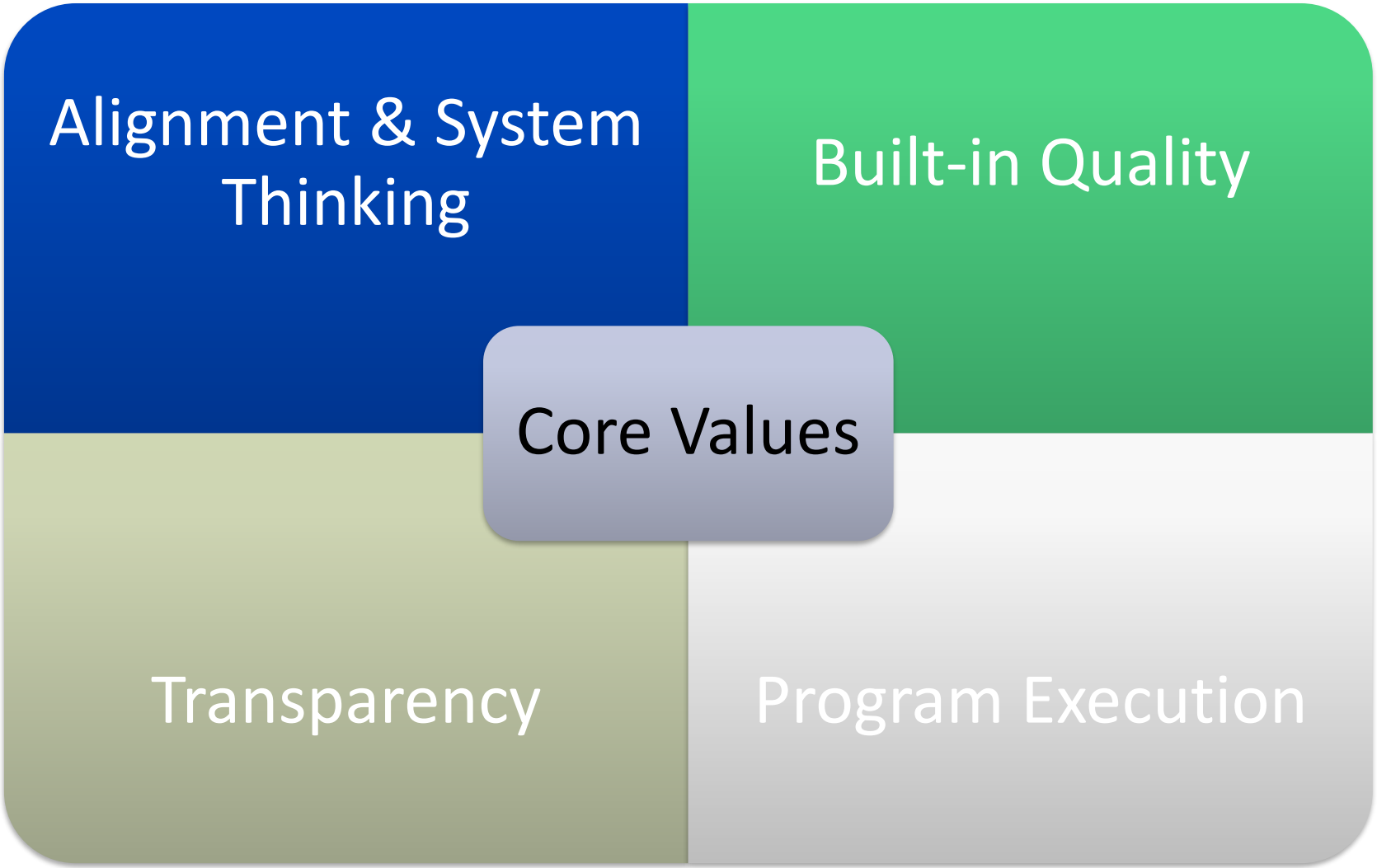
Fachhochschule Dortmund
University of Applied Sciences



Quelle: SAFe 5 for Lean Enterprises (scaledagileframework.com)

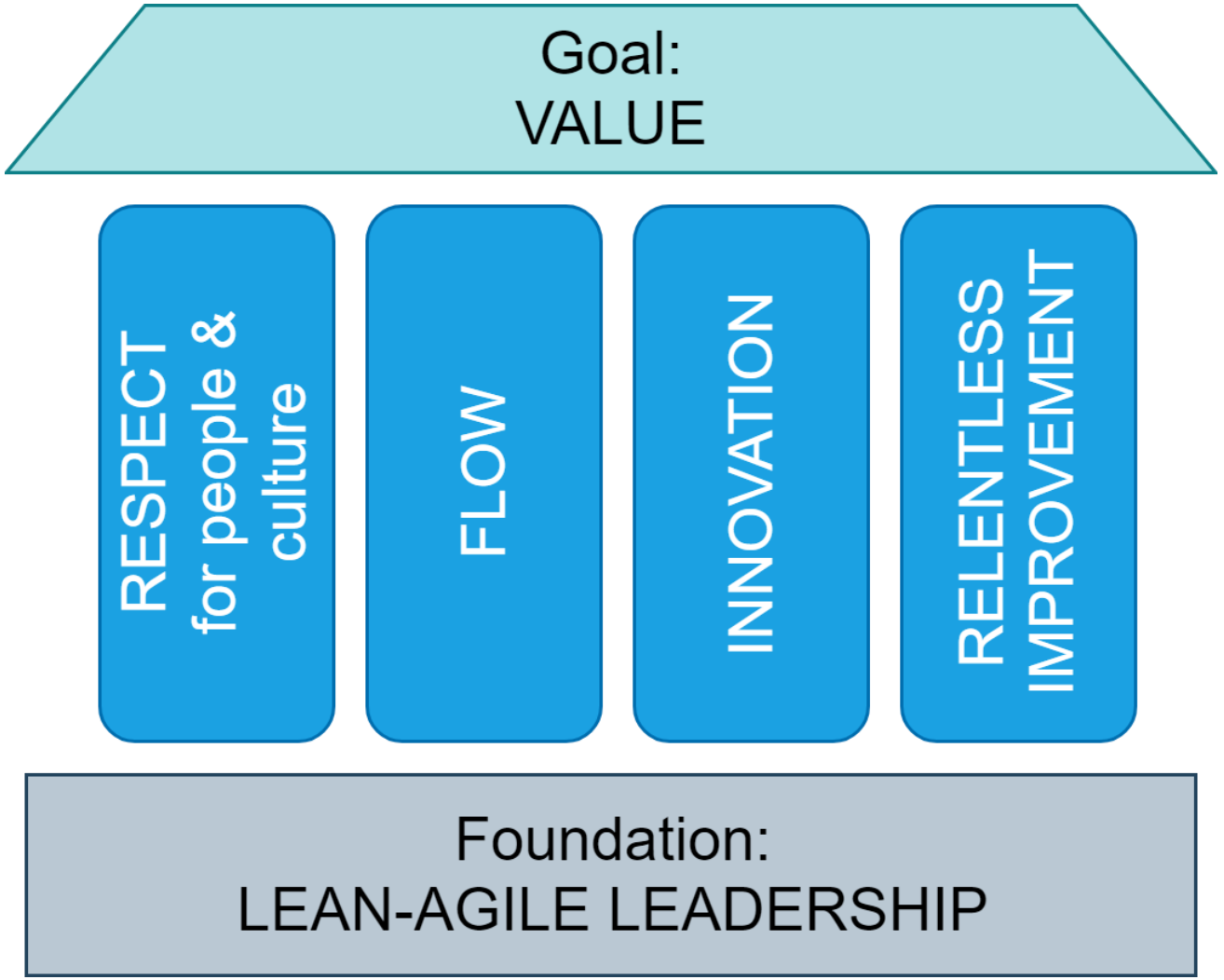
SAFe Foundation
Core Values

Fachhochschule Dortmund
University of Applied Sciences



SAFe Foundation
Lean-Agile Mindset: SAFe House of Lean

Fachhochschule Dortmund
University of Applied Sciences



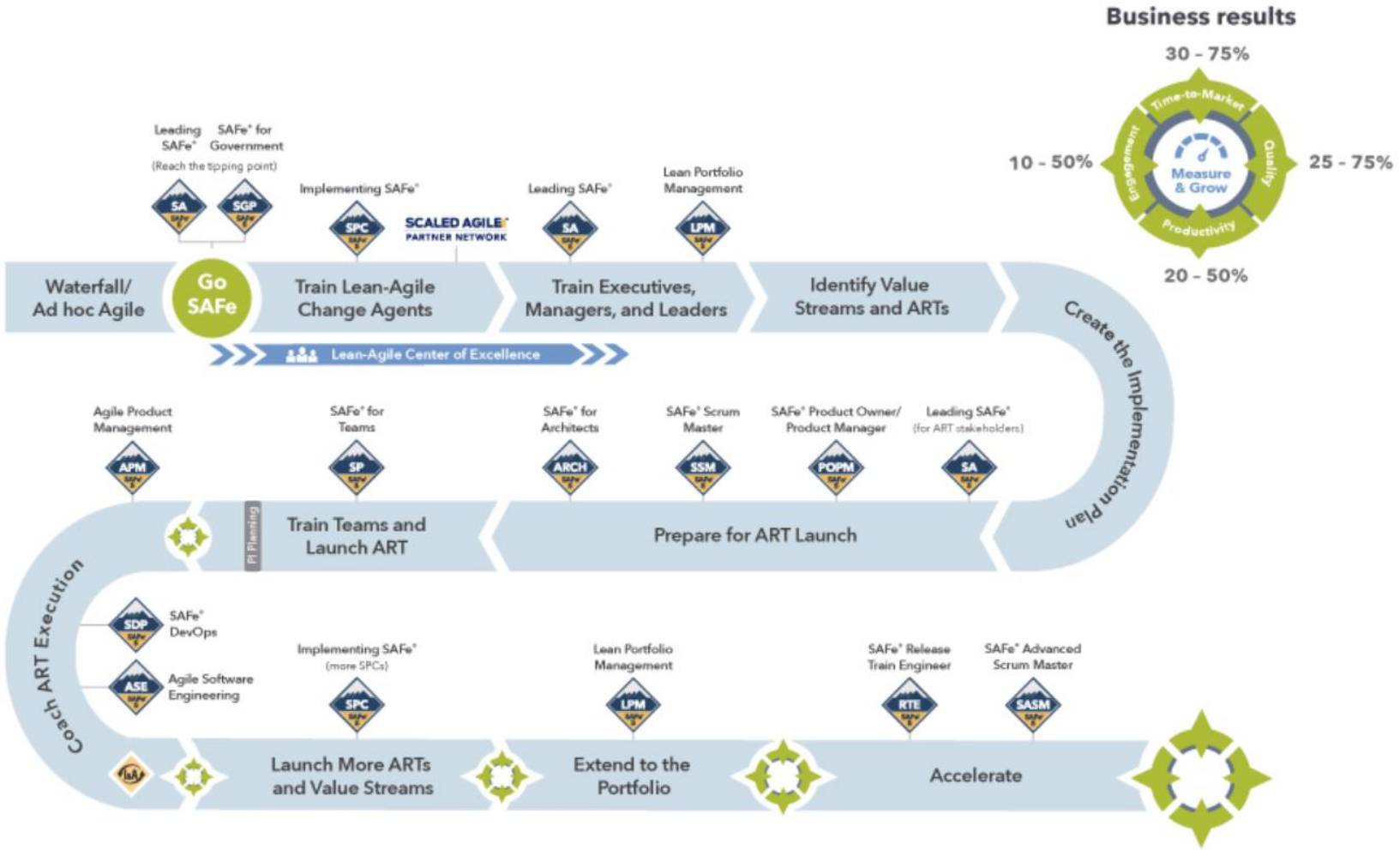
SAFe Foundation
Rollen

- Safe Program Consultant (SPC)
  - Change Agents
  - Technisches Wissen über SAFe
  - Unterstützung bei der kontinuierlichen Verbesserung der Entwicklungsprozesse
  - Wichtige Rolle bei der erfolgreichen Implementierung von SAFe
  - Unternehmensintern oder -extern

- Lean-Agile Leaders
  - Verantwortlich für erfolgreiche Anwendung von SAFe und der Ergebnisse
  - Unterstützen Teams durch Ausüben und Coaching der SAFe Lean-Agile Prinzipien und Praktiken
  - Management

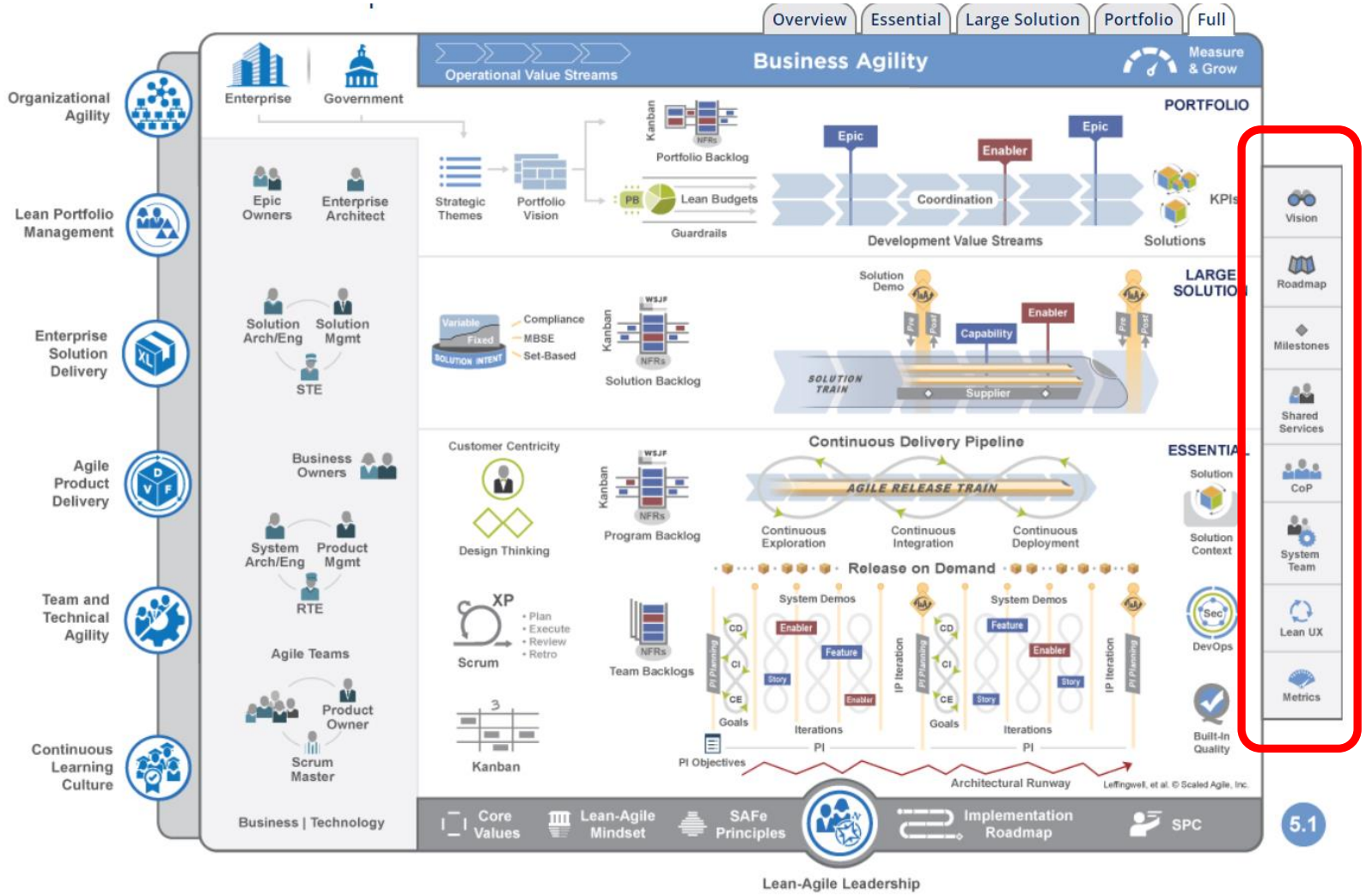
SAFe Foundation
Implementation Roadmap

Fachhochschule Dortmund
University of Applied Sciences



SAFe
Spanning Palette

Fachhochschule Dortmund
University of Applied Sciences



SAFe Spanning Palette
Wesentliche Elemente

Fachhochschule Dortmund
University of Applied Sciences

Metrics

- Objektive Metrik: working solutions
- Weitere mittel- und langfristige Metriken für Fortschrittsmessungen

Community of Practice

- Informelle Gruppe von Teammitgliedern und Experten, die Wissen über Domäne austauschen

Milestones

- Geplante und spezifische Ziele oder Ereignisse

Roadmap

- Geplante Deliverables und Meilensteine auf einer Timeline

Vision

- Zukünftige Sicht der Lösung (Solution), die zu entwickeln ist und kunden- und stakeholderspezifische Anforderungen umsetzt

System Team

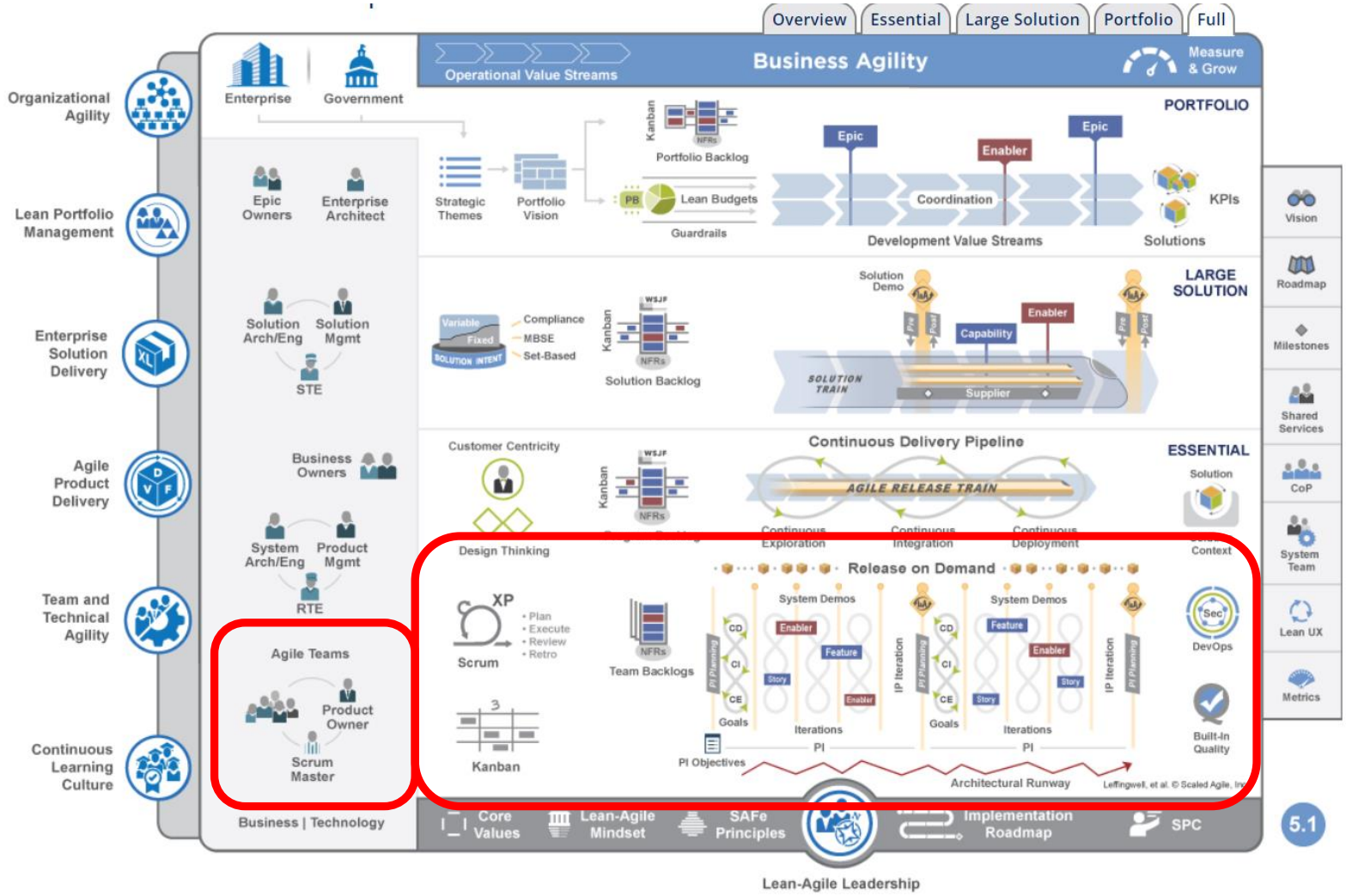
- Agiles Team, welches Umsetzung der agilen Delivery Pipeline (CI/CD, Testautomatisierung, …) unterstützt

Lean UX

- Anwendung der Lean Principles auf User Experience Design

SAFe
Team Level

Fachhochschule Dortmund
University of Applied Sciences



SAFe Team Level
Agile Teams

Fachhochschule Dortmund
University of Applied Sciences

Cross-funktionale Besetzung, bestehend aus

- **Development Team**
- **Product Owner**
- **Scrum Master**

Verwenden **agile Praktiken** der Wahl, typischerweise basierend auf **Scrum, Kanban & XP**

- Häufig: ScrumXP

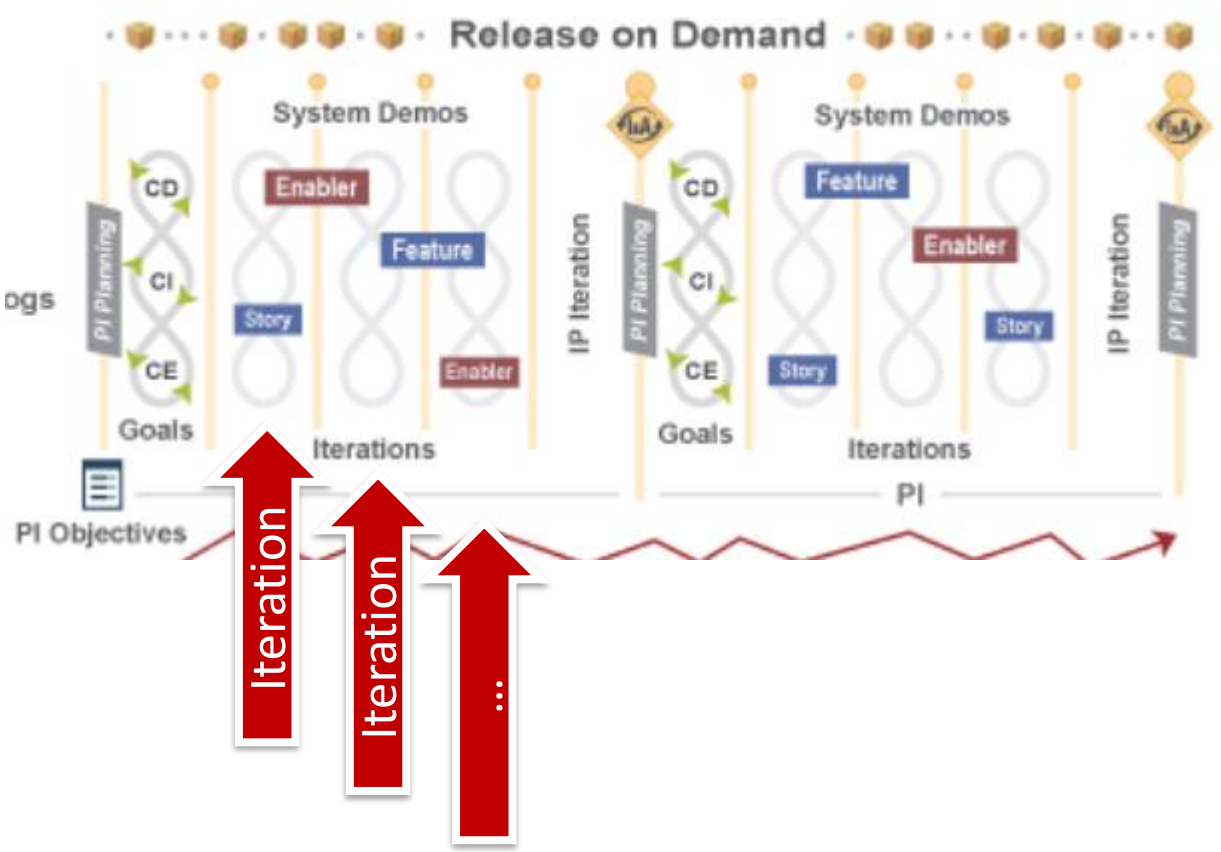
SAFe Team Level
Agile Teams

Fachhochschule Dortmund
University of Applied Sciences

- Aufgaben
  - **Entwickeln werthaltige Inkremente in kurzen Zeitintervallen**
  - **Define, Build, Test, Deploy**
  - Bestimmung von Umfang und Komplexität von Aufgaben
  - **Commitment zu Iterationen oder PIs**
  - Unterstützung oder Erstellung der Automatisierung von CI/CD pipelines
  - Kontinuierliche Verbesserung des Teamprozesses

- Agile Teams & ART
  - „**Agile Teams are on the Train**“: SAFe Agile Teams befähigen den ART durch Zusammenarbeit und Erstellung werthaltiger Inkremente
  - Agieren innerhalb des **gemeinsamen Rahmens**, welcher den ART reguliert und führt
    - Plan together
    - Integrate together
    - Deploy & Release together
    - Learn together

SAFe Team Level Events



- Iteration
  - Planning
  - Review
  - Execution
  - Retrospective
- Backlog Refinement
- Innovation and Planning (IP) Iteration

SAFe Team Level
Teamübergreifende Synchronisation

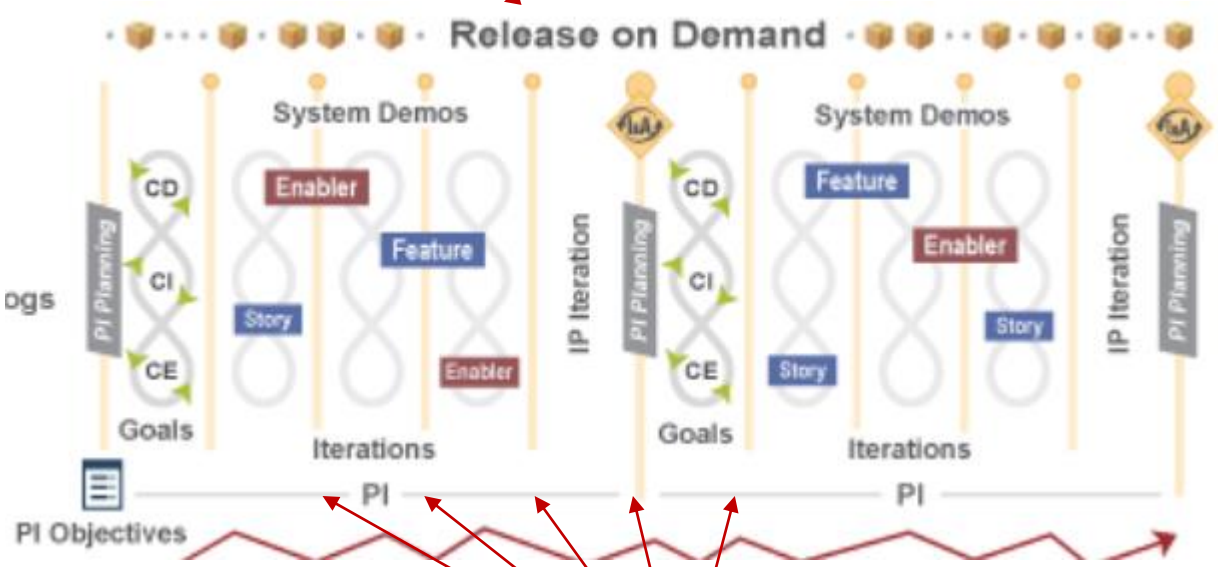
Deploy & Release together

Learn together

Team 1

Team 2

Team 3



Plan together

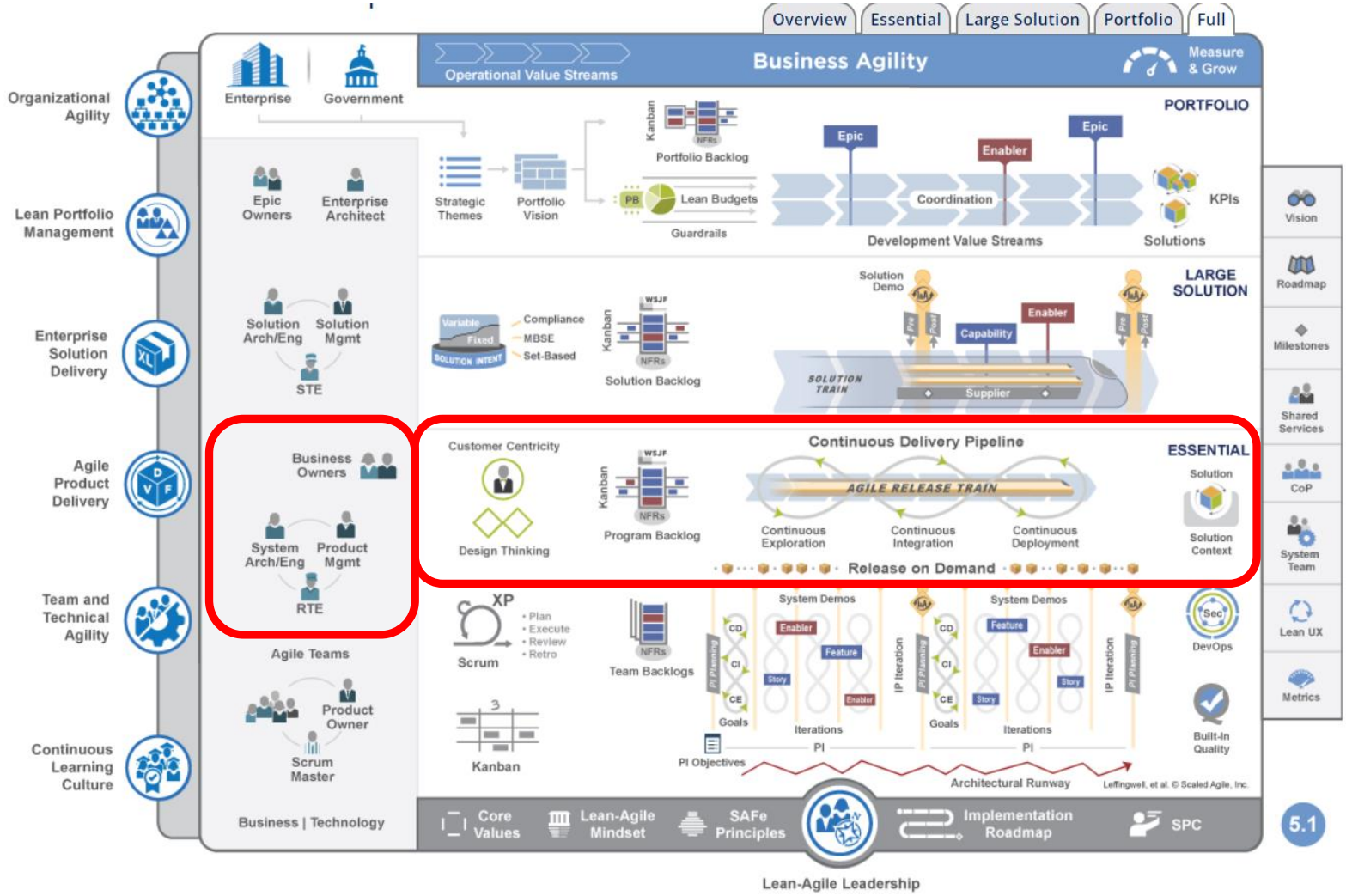
Integrate & demo together

Agile Teams
Built-in Quality

- Software-Praktiken
  - Continuous Integration
  - Test-First
  - Refactoring
  - Pair Work
  - Collective Ownership
  - Agile Architecture
- Hardware- & Firmware-Praktiken
  - Model-Based Systems Engineering
  - Set-Based Design
  - Frequent system integration
  - Exploratory early iterations
  - Design verification

SAFe
Program Level

Fachhochschule Dortmund
University of Applied Sciences



SAFe Program Level
Agile Release Train (ART)

- Langlebiges **Team aus Agile Teams**
- Alignment & Synchronisation mehrerer Teams zu einer **gemeinsamen Business- und Technologie-Mission sowie gemeinsamen Lösung und System**
  - „The teams are sprinting, but the system isn‘t“ → CI über alle Komponenten nach jeder Iteration!
- Entwickelt und liefert Lösungen **inkrementell** aus
  - Basis: **Serie von Iterationen** mit fester **Dauer innerhalb einer PI Timebox**
- Virtuelle Organisation (~ 50 – 125 Personen)
  - Gemeinsame Planung, Kommittent, Ausführung
- Um die wesentlichen **Value Streams** (Wertströme) **der Organisation** organisiert

SAFe Program Level Rollen

Fachhochschule Dortmund
University of Applied Sciences

System Architect / Engineer

- Individuum oder Team
- Definiert die Gesamt-Architektur eines Systems
- Abstraktionsebene oberhalb einzelner Agile Teams und Komponenten
- Definiert Nichtfunktionale Anforderungen, Major System Elements, Subsysteme und Schnittstellen

Product Management

- Verantwortet, was entwickelt wird, gemäß Vision, Road Map und Features im Program Backlog
- Bindeglied zwischen Kundinnen / Kunden und Product Owners
- Begleitet Solution Validation

Release Train Engineer (RTE)

- Ähnlich Scrum Master: „servant leader“, welche/r Ausführung, Impediment-Removal, Risiko- und Abhängigkeitsmanagement und Continuous Improvement auf Programm-Ebene ermöglicht

Business Owners

- Key Stakeholders eines ART
- Verantworten die Business Outcomes eines ART

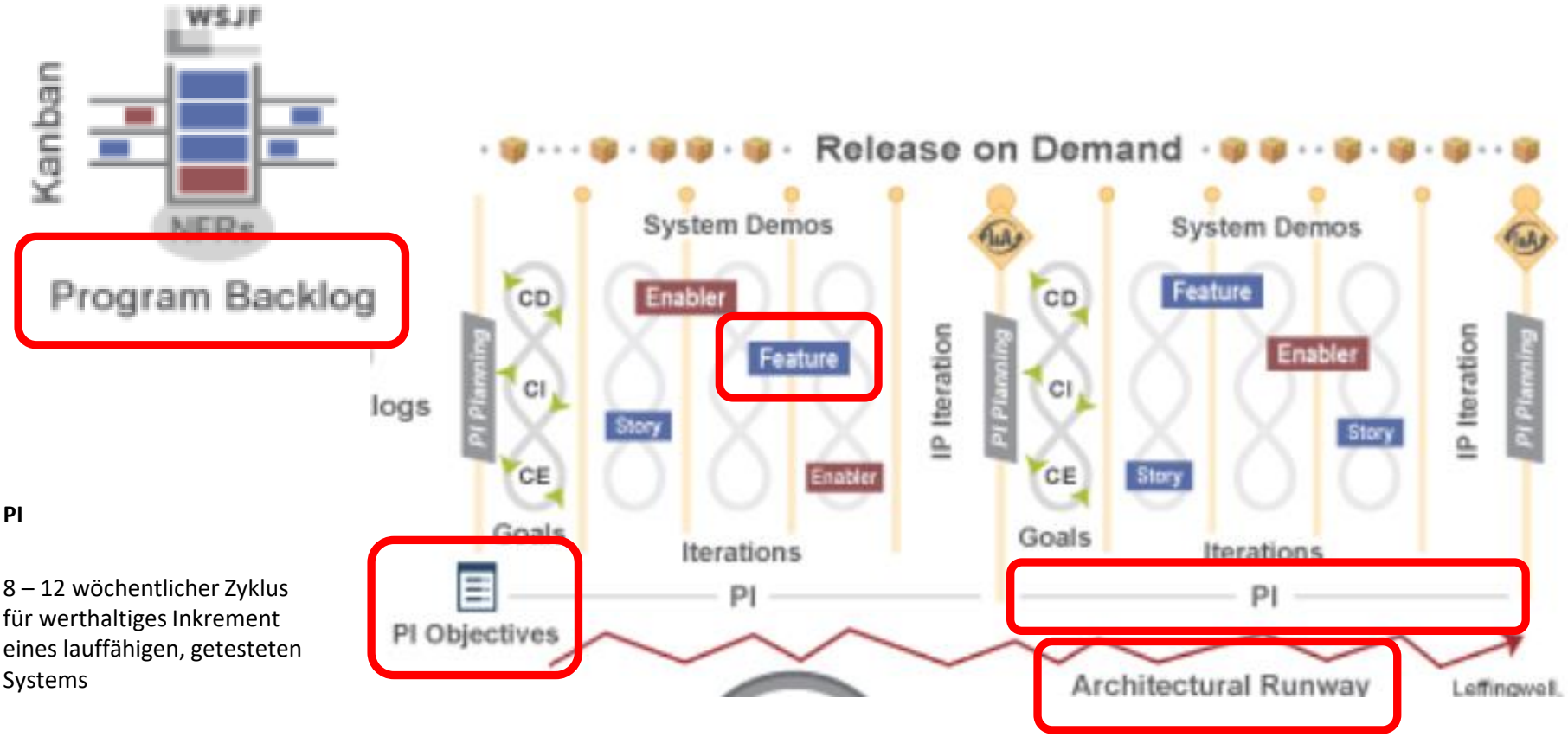
Customers

- Ultimative Käufer:innen einer Solution

Schnittstellen zu Spanning Palette:

- System Team
- Shared Services
- Release Management

SAFe Program Level Artefakte

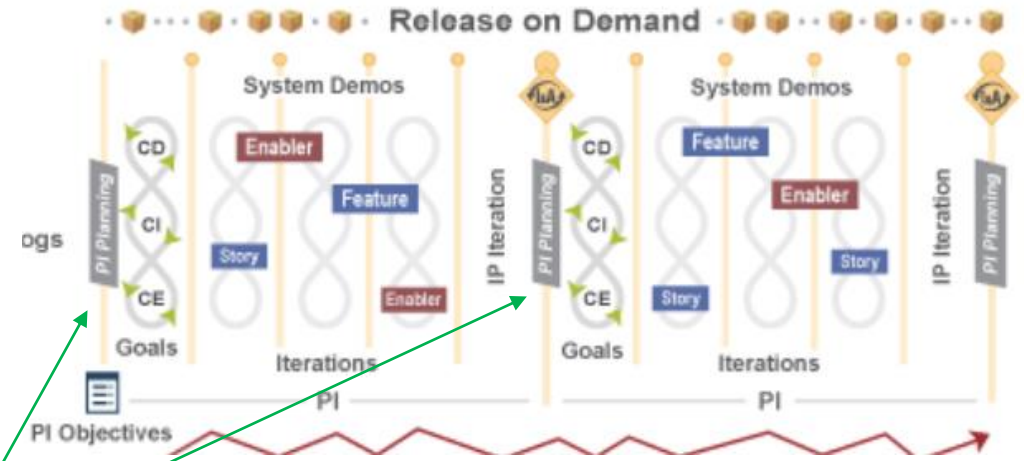


PI

8 – 12 wöchentlicher Zyklus für werthaltiges Inkrement eines lauffähigen, getesteten Systems

Typischerweise 4 Development iterations + 1 Innovation & Planning (IP) iteration

SAFe Program Level
Wesentliche Events



- PI Planning
  - Agile Teams schätzen, was geliefert wird
  - Stellen Abhängigkeiten zu anderen Agile Teams oder ARTs dar
- System Demo
  - 2-wöchentliches Event (nach jeder Iteration) sowie **finale System Demo am Ende des PI**
  - Feedback der Stakeholder
  - Regelmäßige Integration der Agile Teams eines ART
- Inspect & Adapt
  - Regelmäßige Timebox für Reflektion, Problemlösung und Verbesserungsmaßnahmen
  - Ziel: Optimierung der Velocity, Qualität und Zuverlässigkeit der PIs
  - Ergebnis: Menge von Improvement Backlog Features

Literatur

- Balzert, H. (2008): Lehrbuch der Softwaretechnik, Softwaremanagement, 2. Auflage, Heidelberg: Spektrum Akademischer Verlag.
- Pichler, R. (2008): Scrum, Heidelberg: dpunkt-Verlag.
- Sommerville, I. (2012): Software Engineering, 9. aktualisierte Auflage, München: Pearson Studium.
- Wirdemann, R. (2009): Scrum mit User Stories. Hanser Verlag.
- Röpstorff, S., Wiechmann, R. (2016): Scrum in der Praxis, 2. aktualisierte Auflage. Heidelberg: dpunkt.verlag.
- Richard Kastner, Dean Leffingwell: Safe 5.0 Distilled: Achieving Business Agility With the Scaled Agile Framework. Addison Wesley, 2020.
- SAFe 5 for Lean Enterprises (scaledagileframework.com)

Herzlichen Dank für Ihre Aufmerksamkeit !