



Dr. Ralf Lauterbach  
Unternehmensberatung

# PRODUKTENTWICKLUNG

---

IN DER SOFTWARE-INDUSTRIE

Dr. Ralf Lauterbach

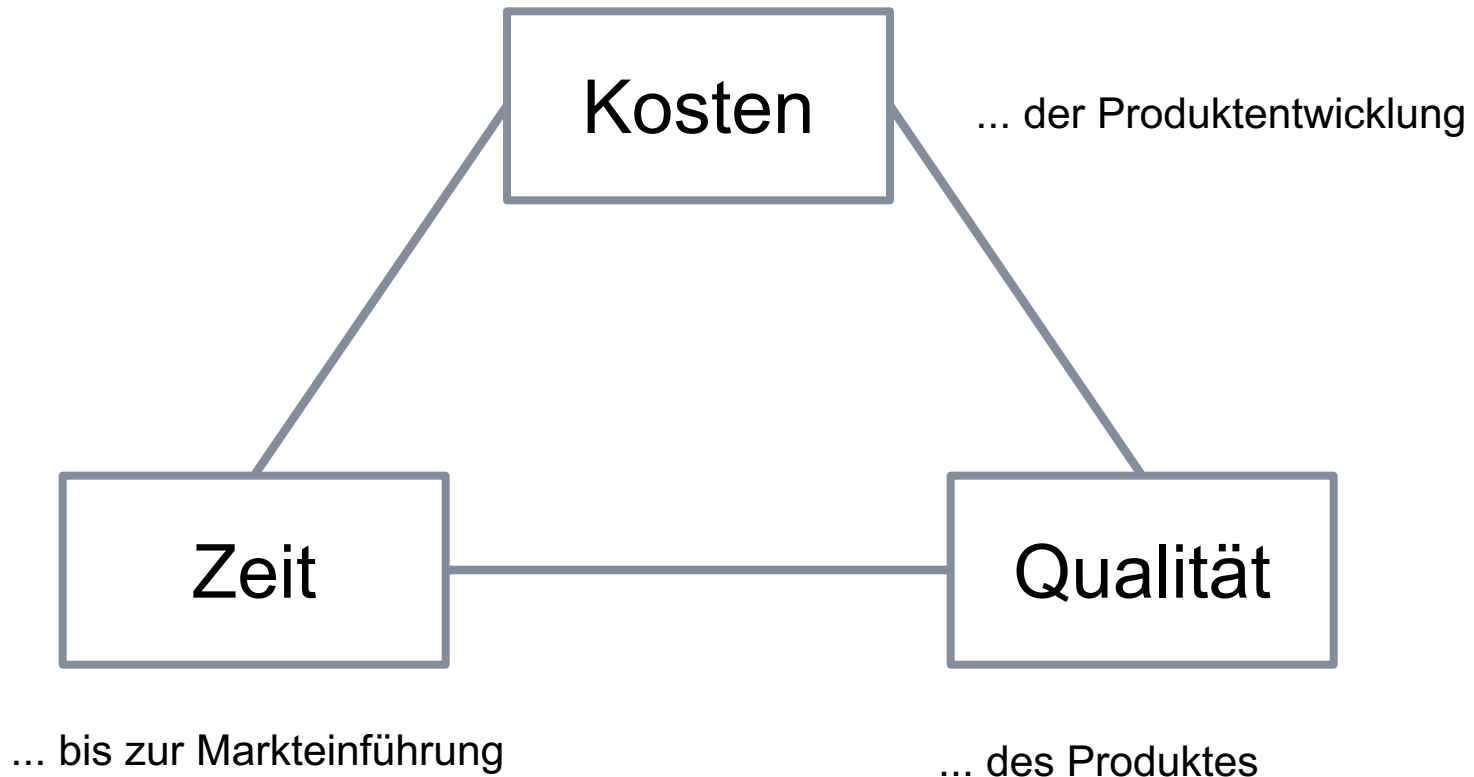
# Produktentwicklung digitaler Produkte - was ist zu tun? -

## Generelle Aufgaben bei jeder modernen digitalen Produktentwicklung

- **Produktmanagement**
  - Marktanalysen
  - Markteingangsstrategie
  - Wettbewerberanalysen
  - Business Case
  - Business Canvas
  - Personas
  - User Journeys
  - User Stories
  - Wireframes
  - Click Dummies
  - Budgetplanung
  - Vermarktung
- **Projektmanagement**
  - Planning & Tracking
  - Risk + Issues
  - Cost Controlling
- **Development**
  - Architektur
  - Technology Review
  - Coding (agile mode)
  - Automated Build Process
  - Releasemanagement
- **QA**
  - Functional Testing
  - System Integration Test
  - NFRs (e.g. Load Testing)
  - Security Testing / Pentests
  - Friendly User Tests
- **Betrieb / Operations**
  - Infrastruktur
  - Integration
  - Certificate Mgmt
  - Device Mgmt
  - Monitoring
  - Reporting
  - 24/7
  - SLA
- **Customer Care**
- **Customer Feedback**
- **Vertriebsbriefing**
- **Kommunikation**
- **Social Media**
- **Pressearbeit**
- **Financial Review**

# Produktentwicklung - High-Level-Sicht -

Trade-offs



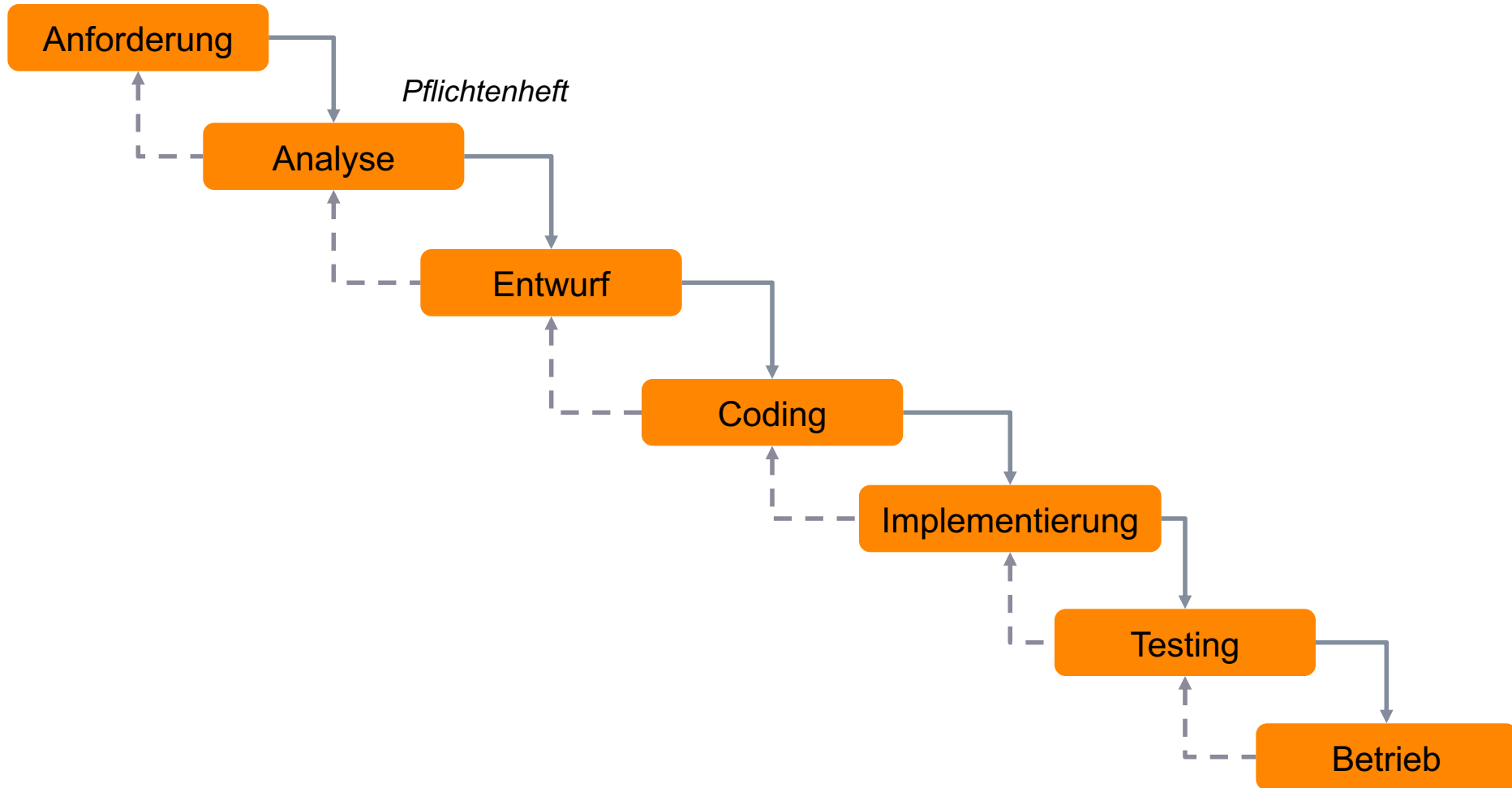
# Produktentwicklungskonzept

Ist das Wasserfallmodell mit seinen Prozessen out?

**Nein.** Im Fall großer Sicherheitsanforderung und bei sehr gut planbaren Projekten haben stabile erprobte Prozesse ihre Berechtigung. Sie schützen ggfs. das Unternehmen!!!

# Das klassische Wasserfallmodell der Softwareentwicklung – was ist das eigentlich?

*Lastenheft*



# Das Wasserfallmodell hat Vorteile bei klaren und sehr stabilen Anforderungen

## •VORTEILE:

- klare Abgrenzung der Phasen
- einfache Möglichkeiten der Planung und Kontrolle
- bei stabilen Anforderungen und klarer Abschätzung von Kosten und Umfang sehr effektives Modell

## •NACHTEILE:

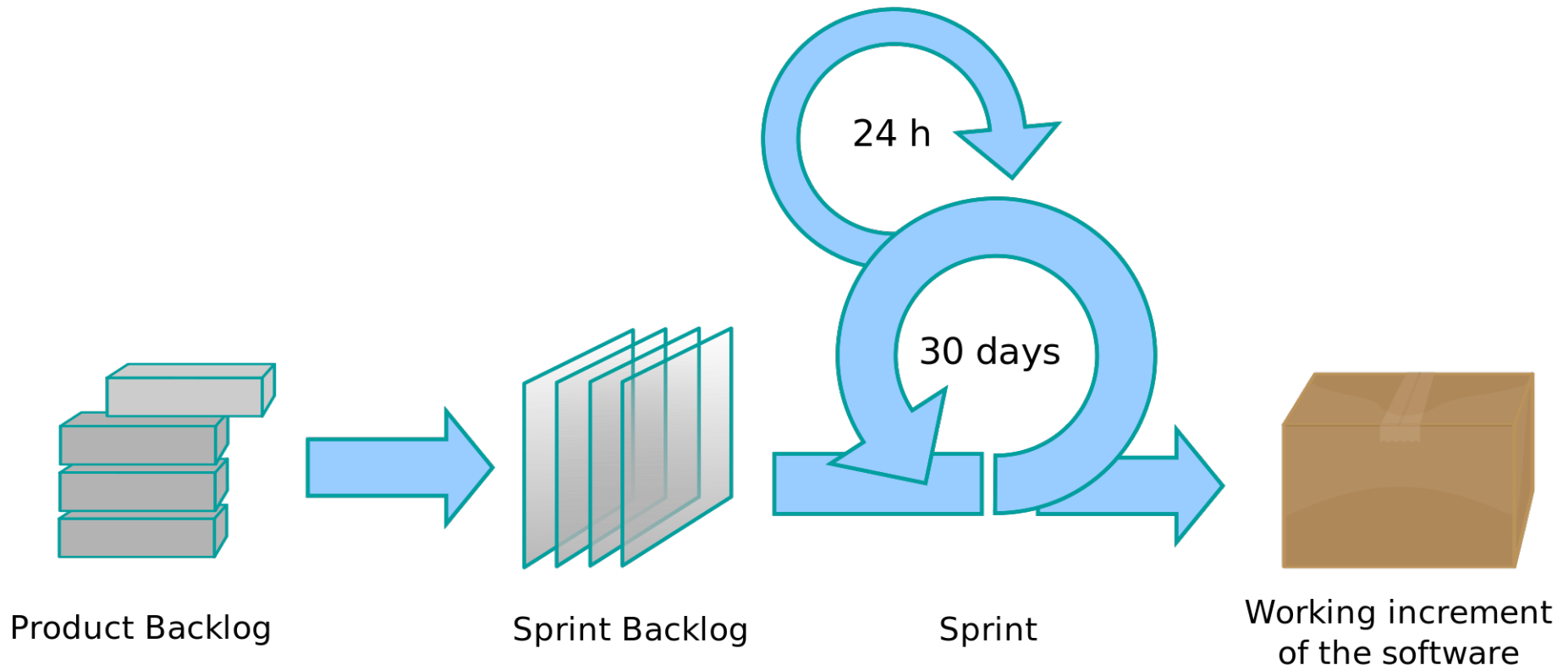
- sehr langer Zeitraum bis man das erste Ergebnis sieht
- kurzfristige / schnelle Änderungen sind nicht möglich
- frühes Festschreiben der Anforderungen führt zu teuren Änderungen
- Fehler werden unter Umständen erst spät erkannt und müssen mit erheblichem Aufwand entfernt werden
- gerne werden die Overhead-Kosten großer Organisationen vergessen!!!

**Typische Stimmen aus der Produktentwicklung hinsichtlich der Vorgaben der IT:**  
„Oft müssen wir alleine für die betrieblichen Abnahmen 6 Wochen und mehr einplanen.“

# Die agile Welt

... und wie sieht jetzt eine agile Lösung aus? Was ist dieses Scrum?

# Scrum – Scope und Anforderungen sind zunächst grob und generisch



Quelle: Wikipedia



# Scrum ist vorteilhaft, wenn die Vision des Produktes noch unscharf ist und mit dem Kunden entwickelt wird

## •VORTEILE:

- Flexibles Modell mit kleinen Iterationen
- Völlige Transparenz
- Freiraum für das Entwicklungsteam
- Anpassungen können während der Entwicklung durchgeführt werden
- Kundenfeedback kann jederzeit schnell umgesetzt werden
- Juristische Anforderungen sind schnell umsetzbar
- Sehr schnelle Reaktion auf Marktänderungen
- Wenig Bürokratie
- „On-Site-Customer“, für tägliches Feedback Auftraggeber-Auftragnehmer

## •NACHTEILE:

- Zu Beginn keine juristisch verbindlichen Vorgaben bezüglich bzgl. des Produktes möglich, da die Ausprägung erst über die Zeit entsteht
- Hoher Kommunikations- und Abstimmungsaufwand – jedoch beschränkt auf ein überschaubares Team
- Keine Garantie, dass alle ursprünglichen Funktionen umgesetzt werden

# Wesentliche Konzeptbestandteile

1. **Time Boxing** - eine Time-Box ist ein Zeitabschnitt, der nicht überschritten werden darf und in dessen Grenzen Meetings oder Entwicklungs-Inkrementen ablaufen.
2. **Sprint** - Scrum-Projekte werden inkrementell entwickelt. Jedes Inkrement ist eine Time-Box von normalerweise 20-30 Kalendertagen und heißt Sprint.
3. **Sprint-Planning-Meeting** - zu Beginn jedes Sprints findet als Kick-Off das Sprint Planning Meeting statt. Es dauert maximal einen Tag (Time-Box 8 Std.) und dient dazu, das Arbeitspaket des Scrum-Teams für den kommenden Sprint (Sprint Backlog) zu schnüren.
  1. Der Product Owner präsentiert dem Team die Product Backlog Items mit der höchsten Priorität und benennt (optional) sein Sprint Ziel, mit dem das Team einverstanden sein muss.
  2. Das Team plant autonom (ohne Mitsprache des Product Owners) im Detail, wie es sein gegebenes Versprechen einlösen kann, indem es die betreffenden UserStories in Tasks herunterbricht und letztere zu einem Sprint Backlog konsolidiert.
4. **Daily Scrum Meeting** - das Daily Scrum Meeting findet zu Beginn eines jeden Arbeitstages statt. Es dauert nicht länger als 15 Minuten (Time-Box) und dient dem Team dazu, sich abzustimmen, gegenseitig zu informieren und Aufgaben zu übernehmen.
5. **Sprint Review Meeting** - am Ende jedes Sprints präsentiert das Team dem Product Owner und allen interessierten Stakeholdern das Ergebnis seiner Arbeit live am System und sammelt Feedback (Meinungen, Verbesserungsvorschläge, Lob und Kritik) ein.
6. **Sprint Retrospective Meeting** - Meeting innerhalb einer 3-stündigen im Anschluß an das Sprint Review Meeting, moderiert vom ScrumMaster. Das Team diskutiert rückblickend den soeben zu Ende gegangenen Sprint und überlegt sich, was weshalb gut oder schlecht gelaufen ist

# Wesentliche Begriffe

1. **UserStories:** Die Anforderungen werden als Geschichten aufgeschrieben: "Als Max möchte ich beim Betätigen des Blinkers in meinem Auto ein akustisches Signal hören".
2. **Epics** sind zusammengefasste **UserStories**.
3. **Personas:** fiktive Personen mit Namen, die den Service / die Software nutzen sollen. Die UserStories werden aus der Sicht jeder fiktiven Person geschrieben.
4. **Product Backlog** = Priorisierte Liste von **UserStories** mit Schätzwerten, welche den jeweiligen Funktionsumfang, ggf. auch ihre Komplexität, relativ zueinander widerspiegeln. Wichtig ist, dass es sich um eine "gefühlte Größe" handelt, nicht um eine absolute Aufwandschätzung in Personentagen. Als Einheit der Schätzwerte werden **StoryPoints** verwendet. Je höher die Priorität einer Anforderung, desto feingranularer sind die Schätzwerte, da das Product Backlog im Wesentlichen nach absteigender Priorität abgearbeitet wird.
5. **Sprint Backlog** = eine Liste von Tasks, welche den Arbeitsumfang des Teams für den Sprint festlegt. Die Liste präzisiert sich während des Sprints und wird täglich von allen Team Mitgliedern gepflegt, so dass sie immer den aktuellen Bearbeitungsstand reflektiert. Der Sprint Backlog ermöglicht es dem ScrumMaster, jederzeit zu erkennen, wo das Team steht.
6. **Burndown Chart** = Graphik, die den Projektfortschritt eines Sprints oder Releases in einer Kurve visualisiert. Die Kurve gibt für jeden Punkt auf dem horizontalen Zeitstrahl an, wieviel Arbeit nach jeweils aktueller Schätzung zu jedem Zeitpunkt noch übrig war/ist, um das Ziel zu erreichen. Mittels einer auf den vergangenen Schätzwerten basierenden Trendlinie durch die Kurve lässt sich das voraussichtliche zeitliche Ende des Projektes vorhersagen bzw. können sich andeutende Abweichungen vom Zeitplan innerhalb eines Sprints prognostiziert und Gegenmaßnahmen ergriffen werden.

# Rollen in Scrum

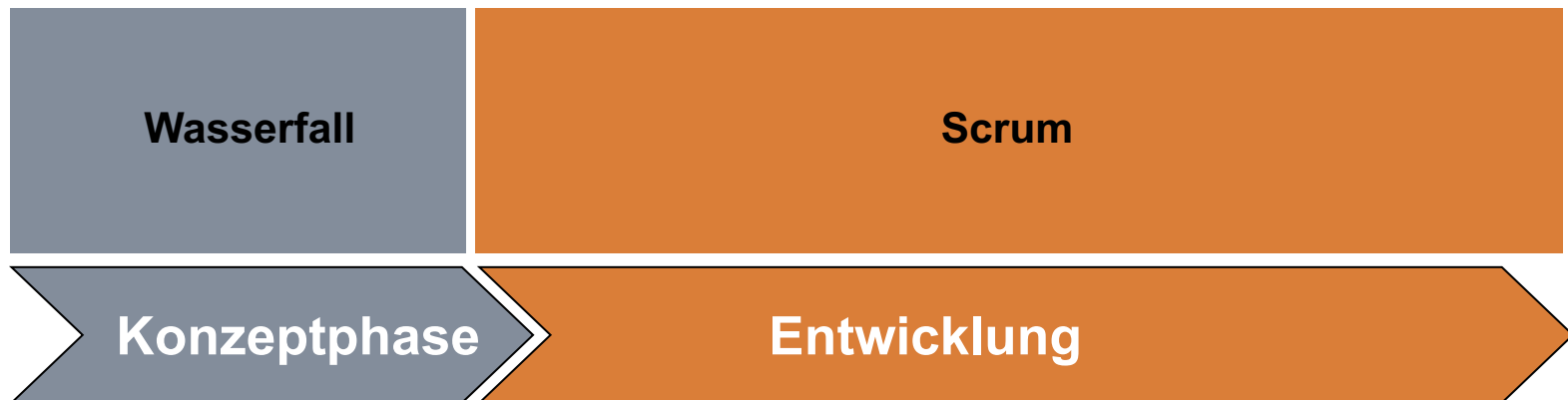
1. **Product Owner:** immer, wenn Software entwickelt wird - egal ob im Kundenprojekt oder für ein Standardprodukt - gibt es einen Auftraggeber, der die fachliche Sicht vertritt, Anforderungen stellt und die spätere Umsetzung seiner Wünsche im Hinblick auf Funktionalität, Benutzbarkeit (Usability), Performanz und Qualität beurteilt. In Scrum ist das der Product Owner. Er schreibt die User-Stories und priorisiert diese. Ferner nimmt er die bearbeiteten UserStories ab und erklärt diese als „fertig“.
2. **ScrumMaster:** agile Prozesse zeichnen sich - deshalb auch diese Bezeichnung - durch eine hohe Dynamik aus. Damit der Prozeß zielgerichtet verläuft und aus Dynamik nicht Chaos wird, gibt es den ScrumMaster. Er ist die Seele des Prozesses und sorgt dafür, daß die Regeln eingehalten werden.
3. **Team:** das Scrum-Team, meistens kürzer einfach Team genannt, steht im Zentrum des Scrum-Prozesses, weil es für die Umsetzung der Anforderungen in Produktfunktionalität sorgt. Das Team insgesamt ist einer von drei Managern(!) in Scrum. Es entscheidet selbständig über das Zerlegen von UserStories in Tasks und deren Verteilung an einzelne Mitglieder (Erstellung des Sprint Backlog aus dem aktuell anstehenden Teil des Product Backlog). Das Team trifft sich täglich zum Daily Scrum, in dem die Team Member einander jeweils einen kurzen Statusbericht liefern. Jedes Team Member kennt das Big Picture des Projekts. Das Team lässt sich nicht vom Product Owner oder ScrumMaster seine Arbeitsweise vorschreiben.

# Vorgehensweise bei der agilen Produktentwicklung

1. Erstellen eines ersten Fachkonzepts
2. Service Design Workshops
  - a. Definition der Personas
  - b. Erstellen der Epics und UserStories
3. User-Experience Design (UE/UX)
4. Wireframes (für Web und Mobile Devices)
5. Click Dummy (für Mobile Applikationen)
6. Development (Sprints)
7. FUT (Friendly User Tests)
8. QA, Security, Pentesting

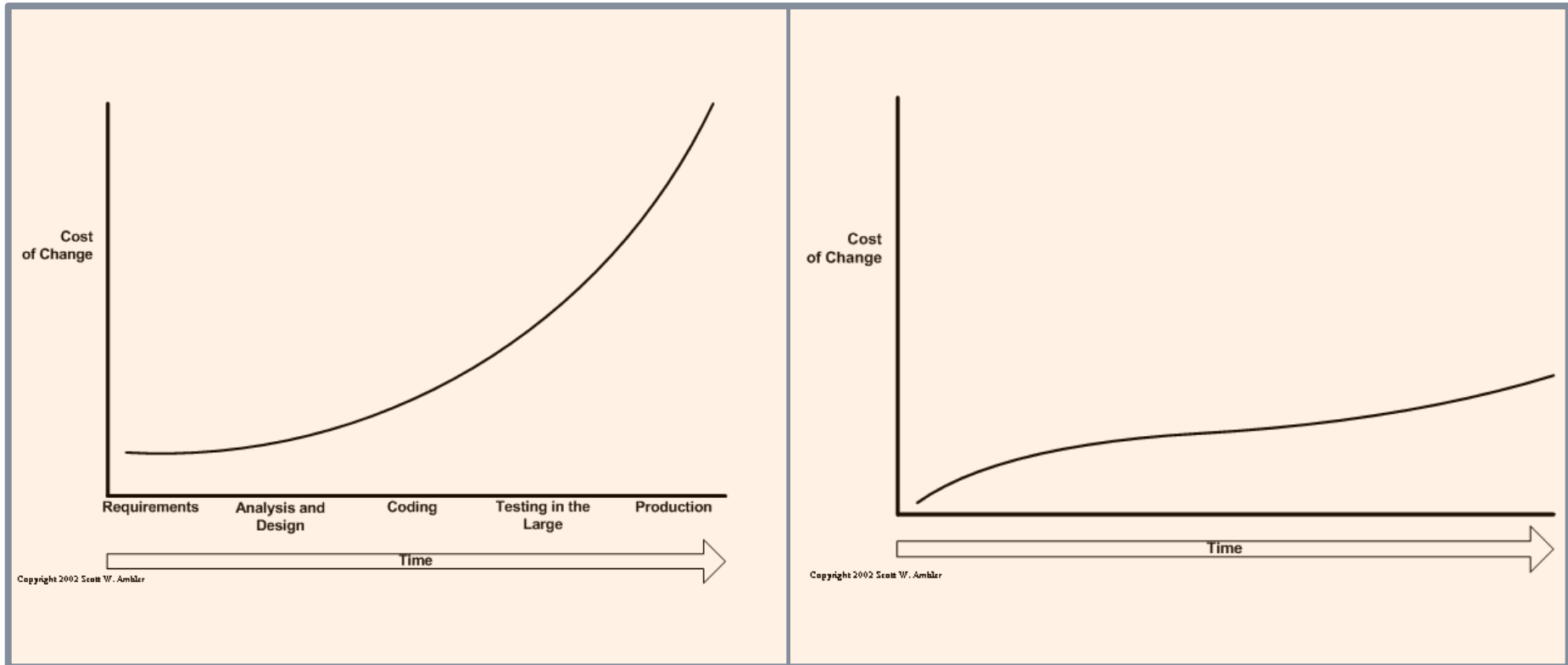
**Wie kann ich überhaupt Kosten und Aufwand eines Projektes schätzen, wenn im agilen Modus erst nach Vergabe der StoryPoints Kosten- und Zeitschätzungen vorliegen? Grobe Planungsphase voranstellen, anschließend in die agile Methodik wechseln.**

- Der gesamte **Scope eines Projektes** wird in der Konzeptphase **grob festgelegt** (Quick & Dirty Highlevel-Analyse durch erfahrenes Spezifikationsteam: Projektmanagement, UserExperience, Architektur)
- Frühzeitige Planungsmöglichkeiten für die **Belange der betrieblichen Teams**
- **Kurzfristige Änderungen später** können schnell umgesetzt werden



# Produktentwicklung

## - Costs of Change – Wasserfall vs. agile Entwicklung



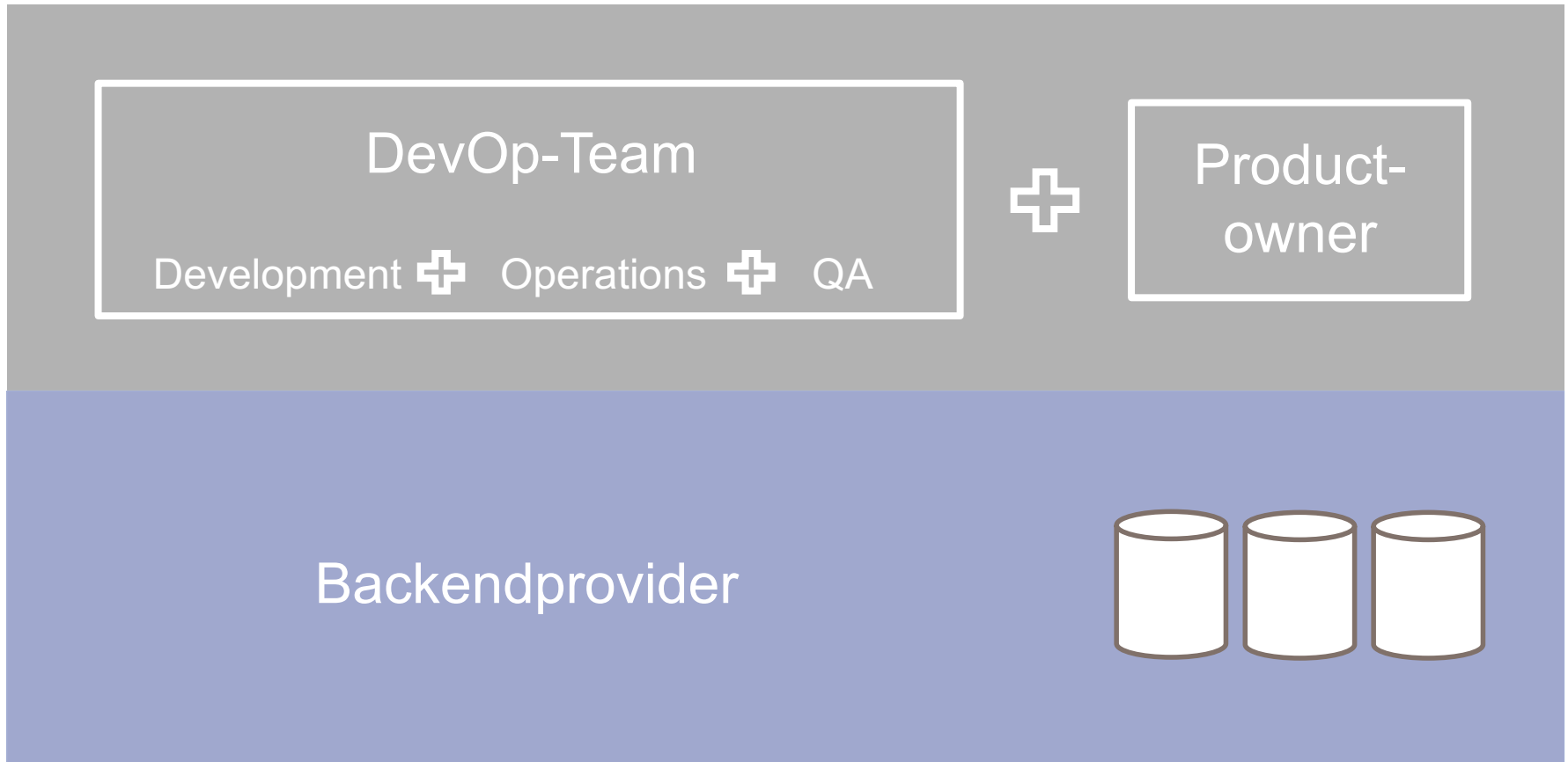
Kurzfristige Softwareänderungen sind bei agiler Entwicklung deutlich kostengünstiger. (Rechtliche Anpassungen, Verbraucherschutzvorgaben, Kundenwünsche,...)

Wenn Software nicht nur entwickelt, sondern auch „betrieben“ werden soll, so hat sich der Aufbau sogenannter DevOp-Teams bewährt.

Experten aus dem operativen Bereich + Experten aus der Entwicklung + der ProductOwner bilden die ausführende Einheit. So wird gewährleistet, dass alle operative Anforderungen in allen Phasen des Projektes berücksichtigt werden.



# DevOp-Teams verfügen über Kompetenzen in Entwicklung, Operations bis QA und realisieren mit dem PO den Service



Radikal verknüpfte Teams mit Operations und Development in einer Einheit.

# Tools

Das **Standard-Toolset** im Scrum Prozess besteht aus 2 Komponenten:

- \* **Jira**
- \* **Confluence**

In **Jira** werden die UserStories vom ProductOwner eingetragen, die Bewertung von den Teams erstellt, die Priorisierung angepasst, der Aufwand geschätzt und dokumentiert, das Burndown-Chart erzeugt etc.

In **Confluence** werden alle Informationen zum Projekt wie die Vereinbarung der „Definition of Done“ aufbewahrt. Confluence ist ein Wiki System, dass jeder Beteiligte nutzen kann.

Beide Tools sind für alle Beteiligten vom Kunden bis zum Entwicklerteam nutzbar und sorgen für die Transparenz im Entwicklungsprozess.

Jira ([www.atlassian.com](http://www.atlassian.com)) wird weltweit von Software-Entwicklungsunternehmen eingesetzt, so dass verteilte Teams oder Projekte - auch über Unternehmensgrenzen hinaus - einfach zu verwirklichen sind.