

The background is a collage of various icons related to web development and technology. It includes a hand cursor pointing at a screen, a globe, a target, a magnifying glass, a notepad, a smartphone, a heart, a keyboard, a document with a checkmark, and a document with a greater-than sign. The icons are in shades of gray and are arranged in a circular pattern around the central text.

# Web-Entwicklung 2

## Vorlesung 5

Fachbereich Wirtschaft - Fachhochschule Münster  
Bachelor of Science Wirtschaftsinformatik Wintersemester 2013/2014



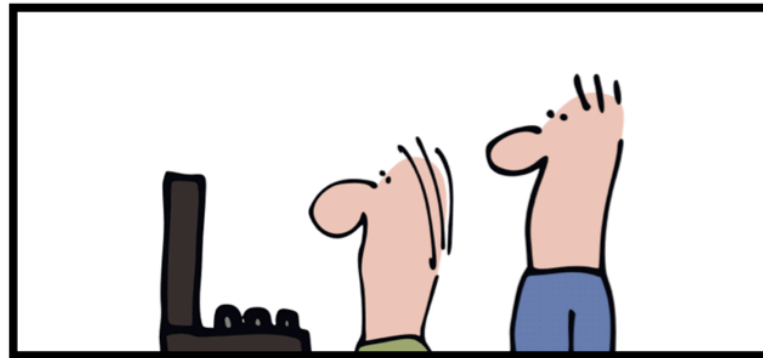


# Projektmanagement

**Disclaimer:**

Der folgende Cartoon ist völlig fiktiv. Eine solche Situation würde **niemals** in der Praxis auftreten ...

# SIMPLY EXPLAINED



PROJECT PLAN

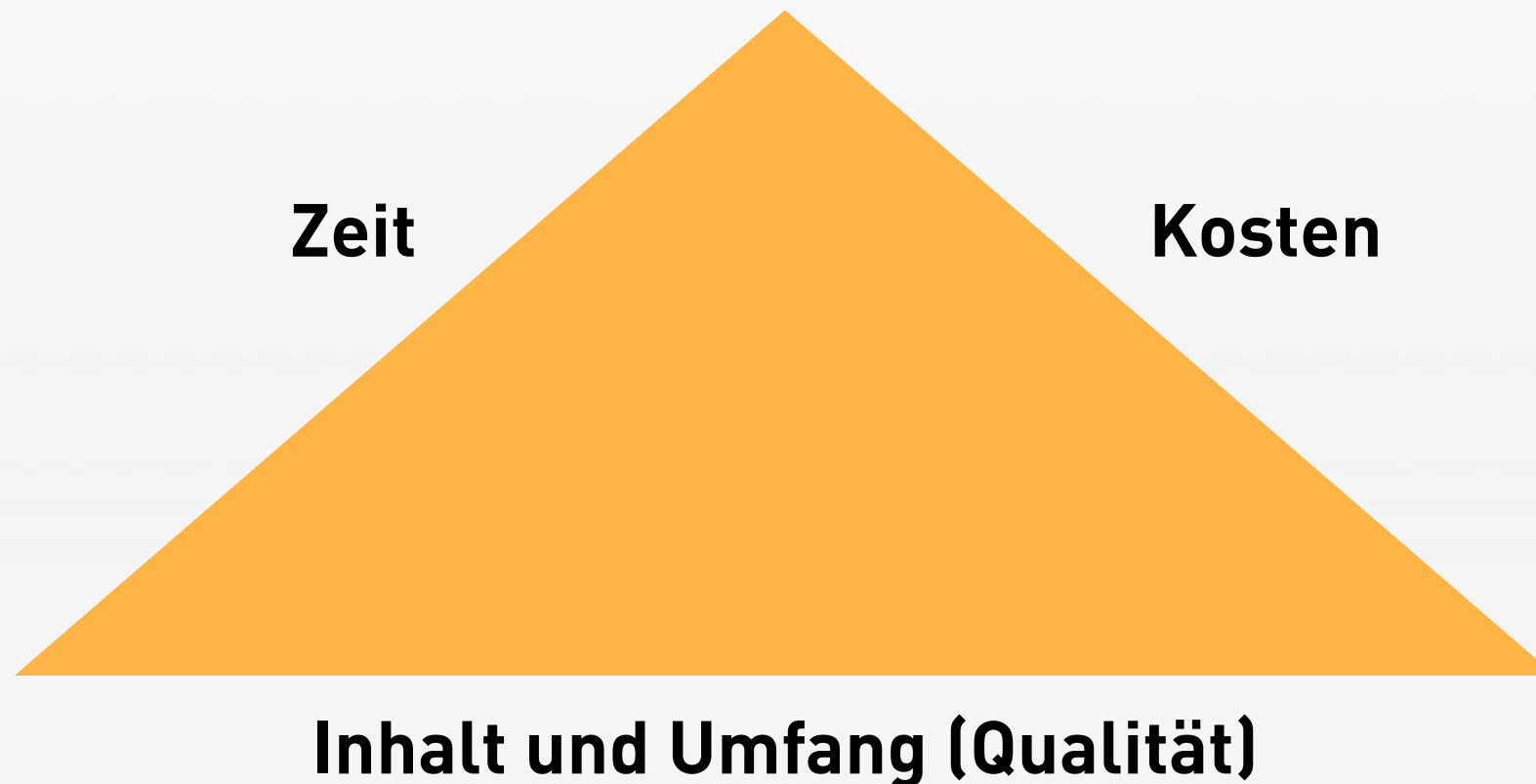
# Was ist ein Projekt?

## Projekt:

Ein Projekt ist ein **zielgerichtetes, einmaliges** Vorhaben, das aus einem Satz von abgestimmten, gelenkten **Tätigkeiten mit Anfangs- und Endtermin** besteht und durchgeführt wird, um unter Berücksichtigung von Zwängen bezüglich **Zeit, Ressourcen** und **Qualität** ein Ziel zu erreichen.

siehe: <http://de.wikipedia.org/wiki/Projekt>

# Magisches Dreieck

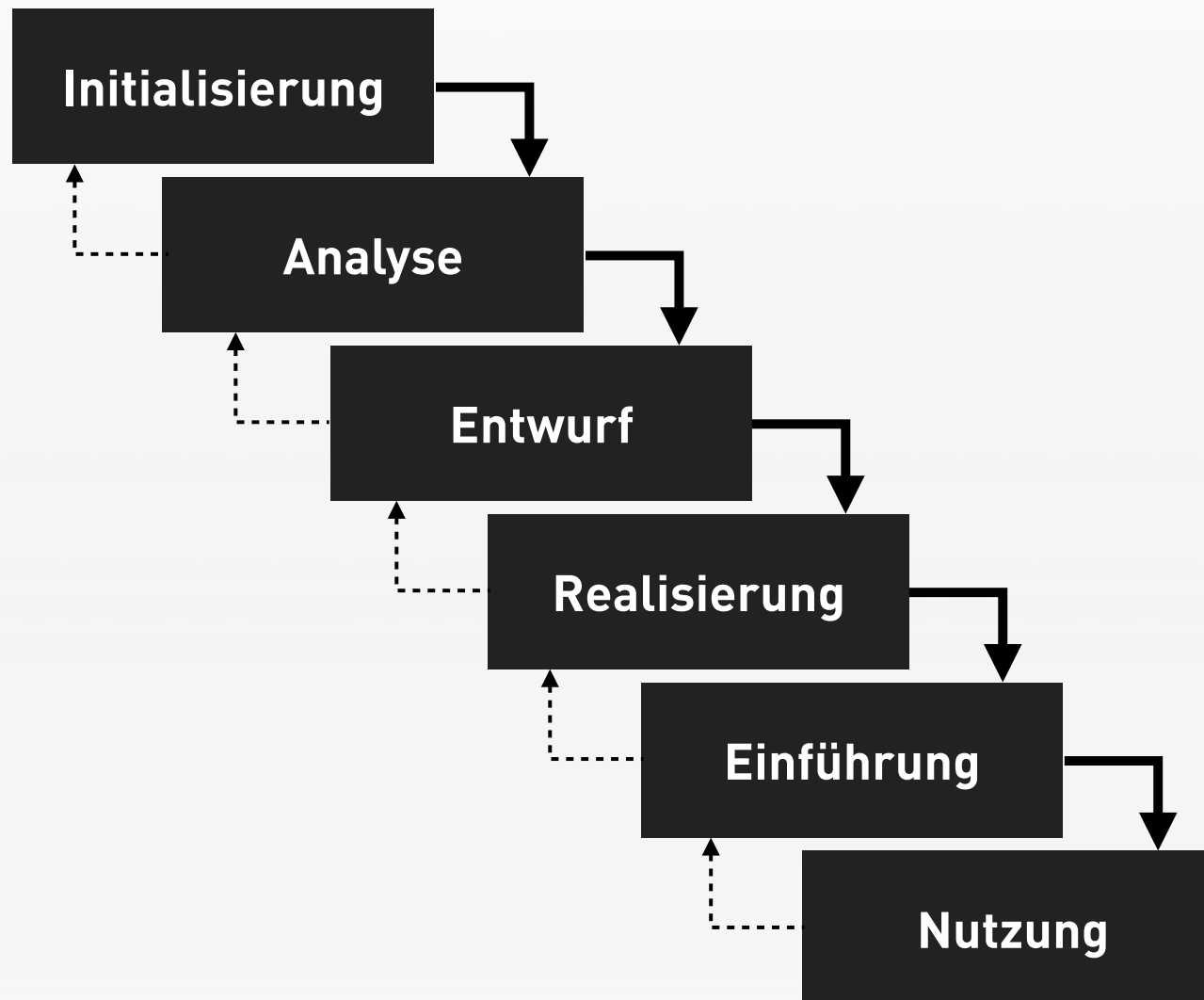






# Klassische Modelle

# Wasserfallmodell





# Probleme klassischer Modelle

- **Komplette Planung im Vorfeld schwierig**
  - Komplexität ist im Vorhinein unbekannt
  - Spätere Änderungen aufwändig bis unmöglich
  - Schätzung vs. Plangenaugigkeit
  - Planabweichungen erfordert aufwändige Change Request-Verhandlungen

# Probleme klassischer Modelle

- Funktionale Arbeitsteilung
  - Hohe Distanz zwischen Implementation und Geschäftsanforderungen
  - Team denkt in starren Verantwortlichkeiten
  - Fehlende Gesamtverantwortung im Projektteam

# Probleme klassischer Modelle

- Hohe Durchlaufzeit
  - Späte Integration der Teilfunktionen
  - Spätes Anwender-Feedback
- Test erfolgt spät und isoliert
- Getrieben durch statische Dokumente
  - Müssen aktuell und synchron gehalten werden
  - Haben keinen Primärnutzen

# Probleme klassischer Modelle

- **Konsequenzen**

- **Hohe Risiken:** Fachlich, Technologie, Qualität und Zielerreichung
- **Abweichungen:** Zeit, Budget und Leistung
- **Keine Ausrichtung an den Kernbedürfnissen des Kunden**





# Agile Entwicklung

# Inkrementelle Entwicklung

- Kern-Idee: Keine vollständige Spezifikation der Funktionen am Projekt-Anfang (Wasserfall-Modell), sondern ...
  - Erstellen einer minimalistischen initialen Grob-Spezifikation  
Implementation eines funktionierenden Systems
  - Iterative Erweiterung und Änderung der Funktionen in Zyklen
  - Kontinuierliches Einholen und Berücksichtigen von Kunden- und Nutzerfeedback

# Agile Manifesto (2001)

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan
- (siehe: <http://www.agilealliance.org/>)

# 12 Prinzipien des Agile Manifesto

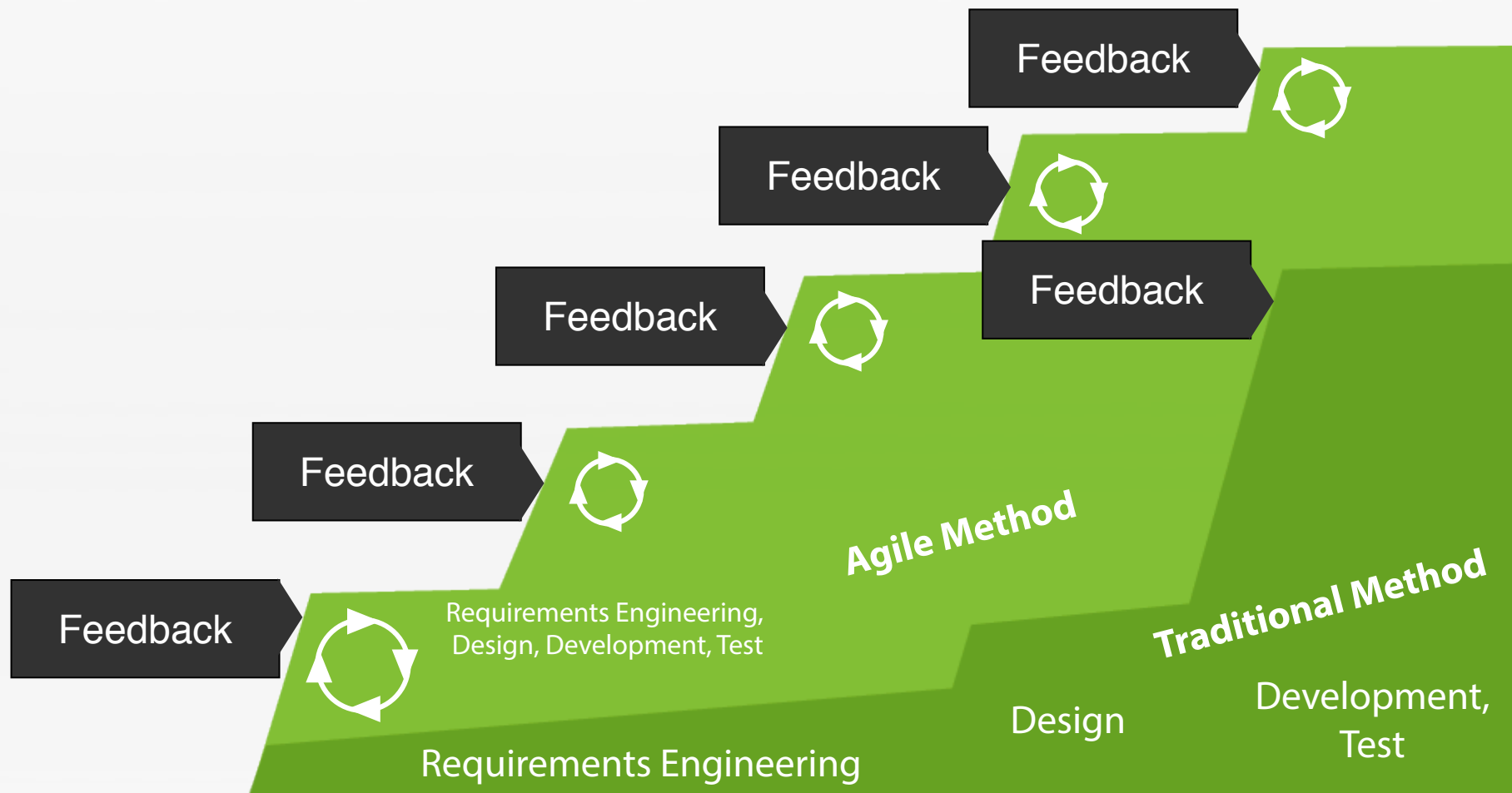
- 1: Our highest priority is to satisfy the customer through **early and continuous delivery** of valuable software.
- 2: Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- 3: Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- 4: Business people and developers must work together daily throughout the project.
- 5: Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- 6: The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- 7: Working software is the primary measure of progress.



# 12 Prinzipien des Agile Manifesto

- 8: Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- 9: Continuous attention to technical excellence and good design enhances agility.
- 10: Simplicity – the art of maximizing the amount of work not done – is essential.
- 11: The best architectures, requirements, and designs emerge from self-organizing teams.
- 12: At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

# Inkrementelle Entwicklung



# eXtreme Programming

- "Vorläufer" agiler Entwicklung
- Grundsatz: Kurze Zyklen, häufige Änderungen
- 12 Prinzipien, v.a.:
  - Pair Programming
  - Test-driven Development
  - Continuous Integration
  - Refactoring
  - Collective Code Ownership
  - Small Releases
  - Sustainable Pace

# User Story

Ein registrierter Nutzer **<Rolle?>**  
soll sich einloggen können **<Was?>**,  
um Zugriff auf nutzerspezifische  
Daten zu haben **<Nutzen?>**





# Scrum

SCRUM

# SCRUM

- Framework für Projektmanagement
- Kurze, iterative Zyklen
- Abgeschlossenes Ergebnis nach jedem Zyklus
- Stetige Änderungsfähigkeit
- Enge Teilhabe des Kunden
- Hierarchielosigkeit
- Gemeinsame Verantwortung

# Rollen

- **Rollenkonzept:**

- **Product Owner:**

- Hat Entscheidungsautorität, welche Features gemacht werden
    - Erstellt & priorisiert Product Backlog

- **Scrum Master:**

- Überwacht die Einhaltung des Prozesses
    - Kümmert sich um Hindernisse

- **Team:**

- Entscheidet, wie viele Stories in einem Sprint umgesetzt werden
    - Organisiert sich selbst

# Planung eines Sprints

- Vorbereitung:
  - Anforderungen sammeln
  - Stories schreiben & priorisieren
  - Input sammeln / erstellen
  - Wireframes, Designs, techn. Spezifikationen etc.
- Sprint Planning
  - Product Owner stellt Stories vor
  - Team füllt Sprint Backlog nach Prio
  - (Planung der technischen Umsetzung)

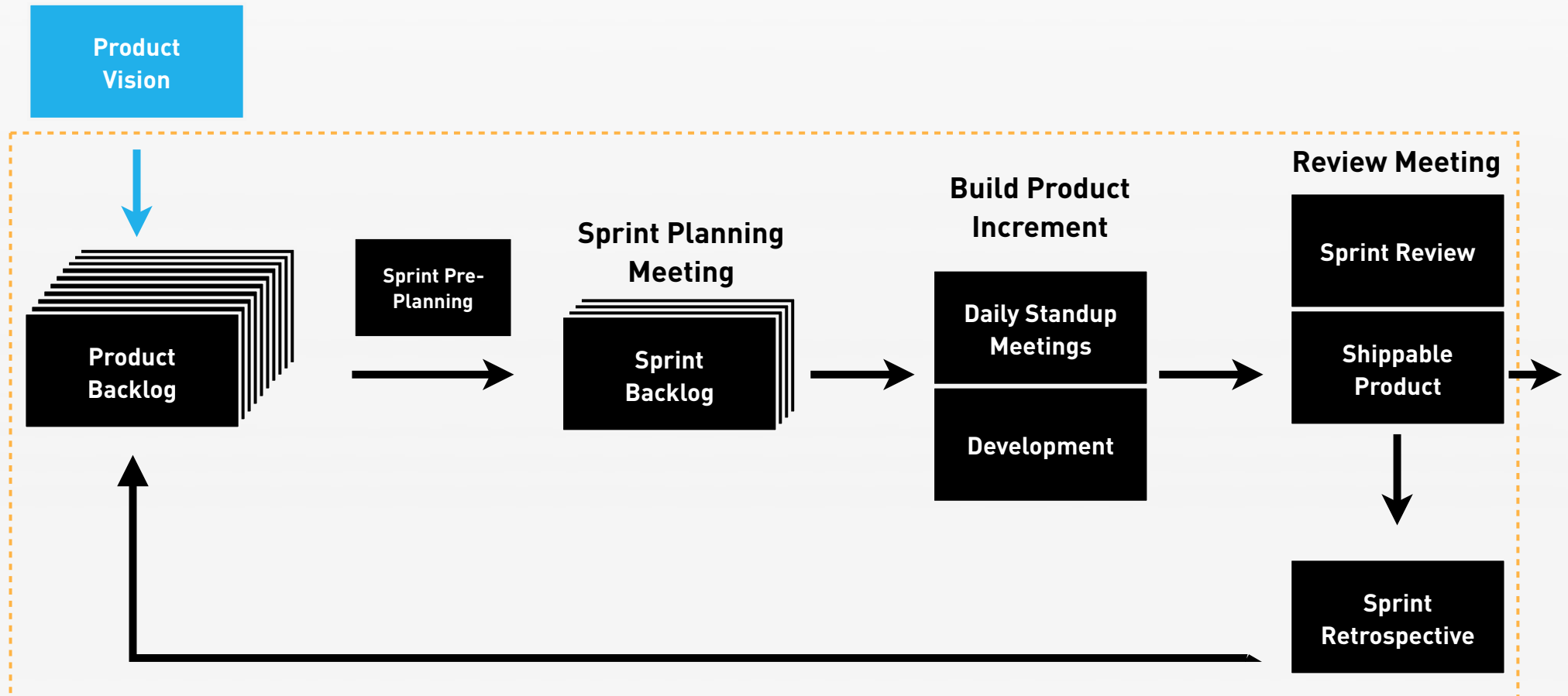


# Durchführung eines Sprints

- **Daily Scrum**
  - Bericht der Arbeitsergebnisse des Vortags
  - Aufnahme von Problemen
- **Sprint Review**
  - Team präsentiert Ergebnisse
  - Product Owner nimmt oder lehnt Stories ab
- **Retrospektive**
  - Rückblick auf Hindernisse
  - Lernprozess

## SCRUM

# Prozess



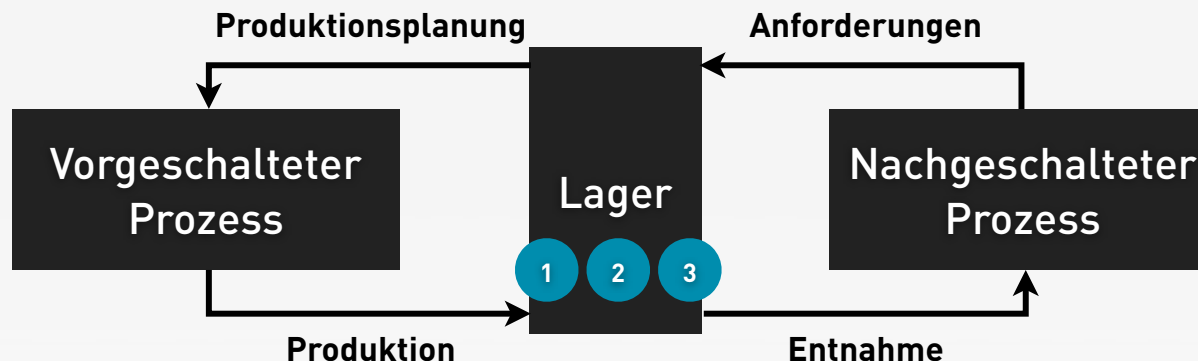


# Kanban

Lean Development

# Ursprung des Kanban-Prozesses

- Technik aus dem Toyota-Produktionssystem
- Lean Production
  - Pull-Prinzip  
Supermarktprinzip statt großer Puffer
  - Just-in-Time  
Autonome Teilsysteme optimieren ihre Effizienz



# Kanban in der Software-Entwicklung

- Jedes Feature durchläuft Prozessschritte  
Entwicklung -> Qualitätssicherung -> Deployment -> Feedback
- Lean Development
  - Visualisierung des Arbeitsflusses
  - Limitierung der Tickets / Schritt
  - Optimierung von Durchlaufzeit, Wartezeit & Durchsatz



# Kanban vs. Scrum

- Beide sind agil
  - Kontinuierliche Auslieferung
  - Herunterbrechen in kleine Stories, Ticket-Beschränkung
  - Selbst-organisierende Teams
  - Stete Optimierung

Scrum	Kanban
Zeitlich beschränkte Iterationen	Unabh. Prozesse („Timeboxing“ optional)
Neue Tickets warten bis nächste Iteration	Neue Tickets kommen ggf. als nächstes dran
Ticketbeschränkung per Sprint	Ticketbeschränkung pro Prozessschritt
Board wird jede Iteration neu befüllt	Board bleibt bestehen
KPI: Geschaffte Story Points / Iteration	KPIs: Durchlaufzeit, Wartezeit, Durchsatz

# Taskboard

The screenshot displays the AgileZen Taskboard interface for a project named 'Leeze.ms'. The browser address bar shows the URL <https://agilezen.com/project/49306/board>. The interface includes a navigation bar with tabs: Leeze.ms, Home, Board, Work, Performance, People, and Process. A user profile for Peter Grosskopf is visible in the top right corner, along with links for Dashboard, Settings, Help & Support, and Log Out. The main workspace is divided into five columns representing different stages of the workflow: Backlog, Ready, In Progress, In Code Review, and In Customer Review. The Backlog column contains four tasks, each with a number, a description, and the assignee's name and profile picture. The Ready column has two tasks, and the In Progress column has one task. The In Code Review and In Customer Review columns are currently empty. A vertical 'Archive' tab is located on the right side of the board.

Column	Task ID	Assignee	Description
Backlog	4	Peter Grosskopf	Ein Nutzer soll alle seiner Touren auf seinem Profil abrufen können
	5	Peter Grosskopf	Ein Administrator soll Touren verwalten können
	6	Peter Grosskopf	Ein Nutzer soll seine Touren verwalten können
	7	Peter Grosskopf	Ein Nutzer soll Fotos für Touren hoch laden können
Ready	2	Peter Grosskopf	Ein Besucher soll sich registrieren können
	3	Peter Grosskopf	Ein Besucher soll sich einloggen können
In Progress	1	Peter Grosskopf	Ein Besucher soll eine Tour anlegen, abrufen, bearbeiten, löschen können
In Code Review			
In Customer Review			





# **The Lean Start-Up**

# Ausgangssituation

- Aufbau eines neuen Internetportals  
=> Aufbau eines Start-Up
  - Ungewissheit über ...
    - ... Ideen und Maßnahmen
    - ... Entwicklungsdauer
    - ... technische Machbarkeit
    - ... Verhältnis zwischen Investition und Return
- ➡ Reduktion der Ungewissheit durch Planung
- ➡ „Wasserfall“

# Probleme

- Das Start-Up basiert auf Vermutungen, ob die Produktideen ...
  - ... dem Nutzer gefallen
  - ... vom Markt angenommen werden
- Es werden Monate in die Entwicklung investiert, ohne Feedback vom Kunden zu bekommen

# The Lean Start-Up

**„Lean Start-Up is a disciplined, scientific and capital efficient method for discovering and building products and services that people love.“**

# The Lean Start-Up

- Buch von Eric Ries ([theleanstartup.com](http://theleanstartup.com))
- „Gegenmodell“ zum Wasserfall-Ansatz
- Ausgangssituation: Kunde und Features unbekannt
- Ziele:
  - Unsicherheiten beim Aufbau eines Start-Up reduzieren
  - Erfolg unternehmerischer Aktivitäten steigern
- Beispiele:
  - Zappos oder „Die Samwers“

# The Lean Start-Up

- **Thesen:**

- Eliminate Uncertainty
- Work smarter not harder
- Validated Learning
- Continuous Customer Interaction
- Revenue goals from day one, no scaling until revenue
- Build, measure and learn

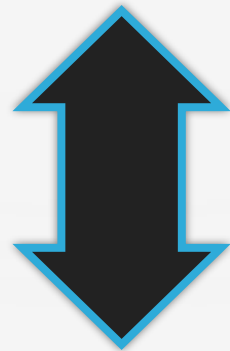
- **Bestandteile:**

- Customer Development
- Agile Product Development

# The Lean Start-Up

## Customer Development

- Rapid hypothesis testing about market, customers, ...
- Low cost, low burn, tight focus

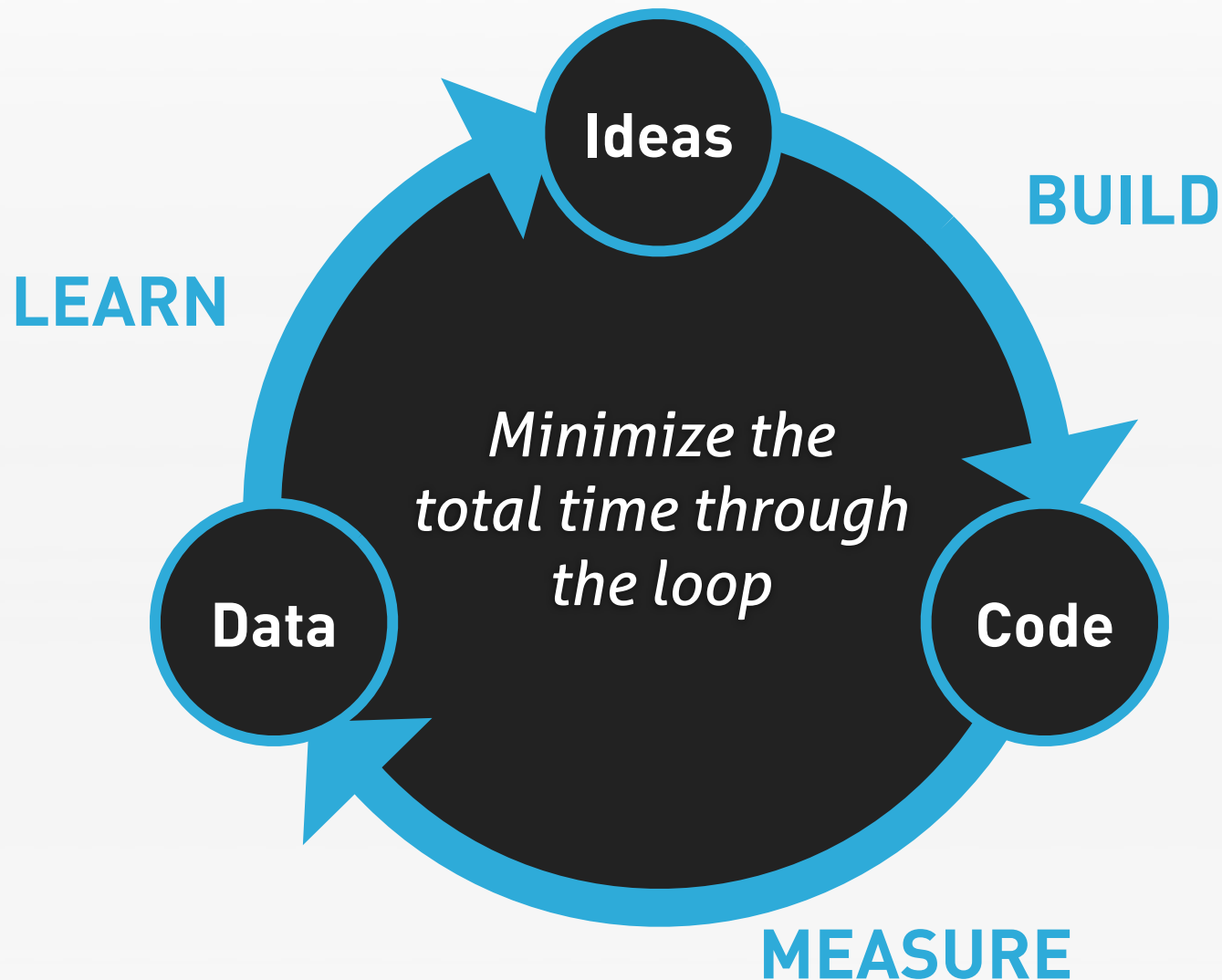


## Agile Product Development

- Tightly coupled with customer development
- Minimum feature sets, max. customer coverage
- Product release cycle hours not years



# Build - Measure - Learn



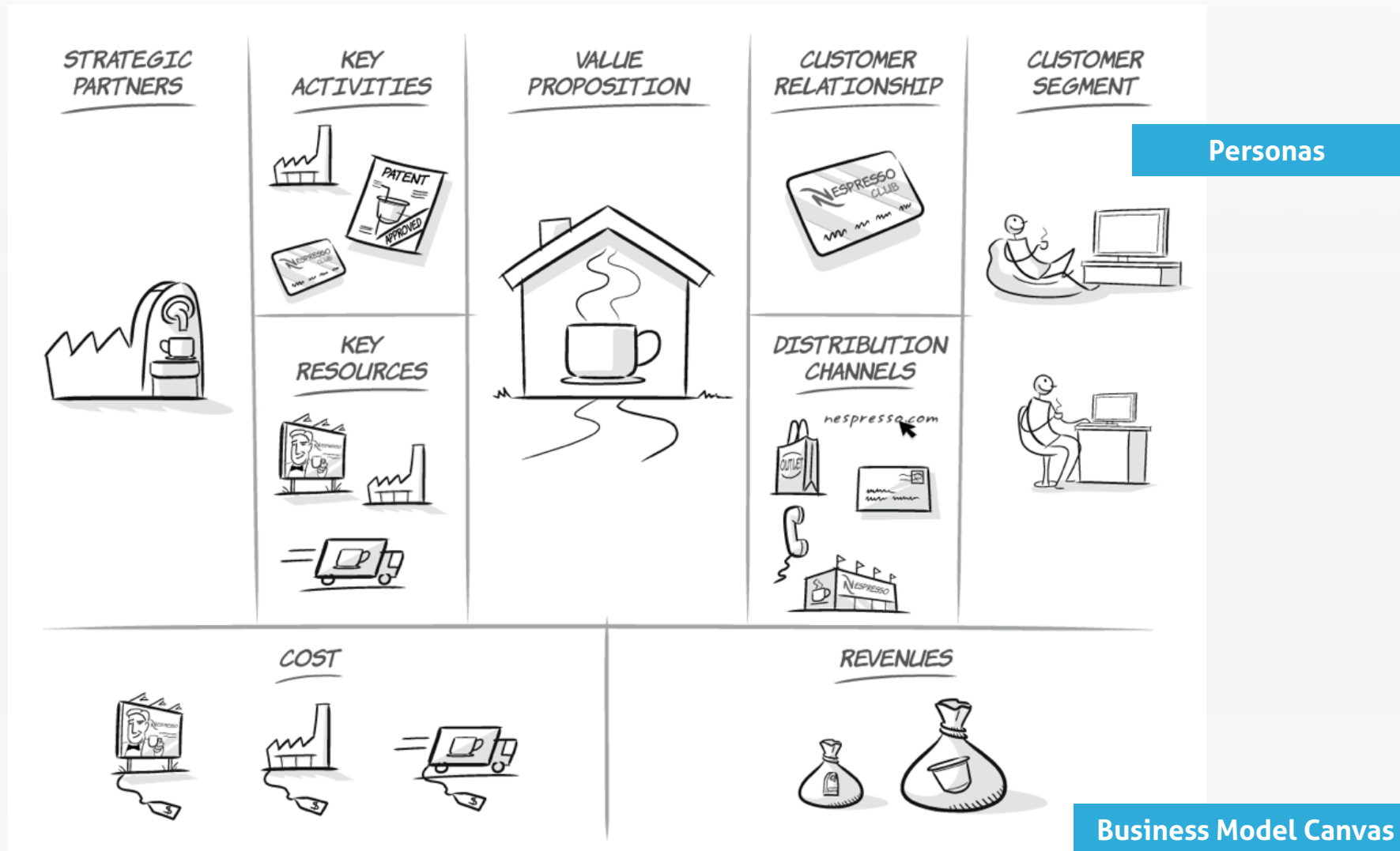
# Minimum Viable Product

„The minimum viable product is that version of a new product which allows a team to collect the maximum amount of validated learning about customers with the least effort.“

# Minimum Viable Product

- Nicht zu verwechseln mit „Minimalem Produkt“
- Startpunkt für einen iterativen Prozess bestehend aus:
  - Ideengenerierung / Brainstorming
  - Prototypenentwicklung
  - Customer Validation, Datensammlung und Analyse
- Beispiele:
  - ...

# Weitere Tools





# Projektmanagement ...

... in der Praxis

# Projektvorgehen

- Ziel/Vision
- Brainstorming
- Strukturierung/User Stories
- Projektplanung (Timeline, Budget, Ressourcen, Technologiewahl)
- Umsetzung (...)

# Umsetzung

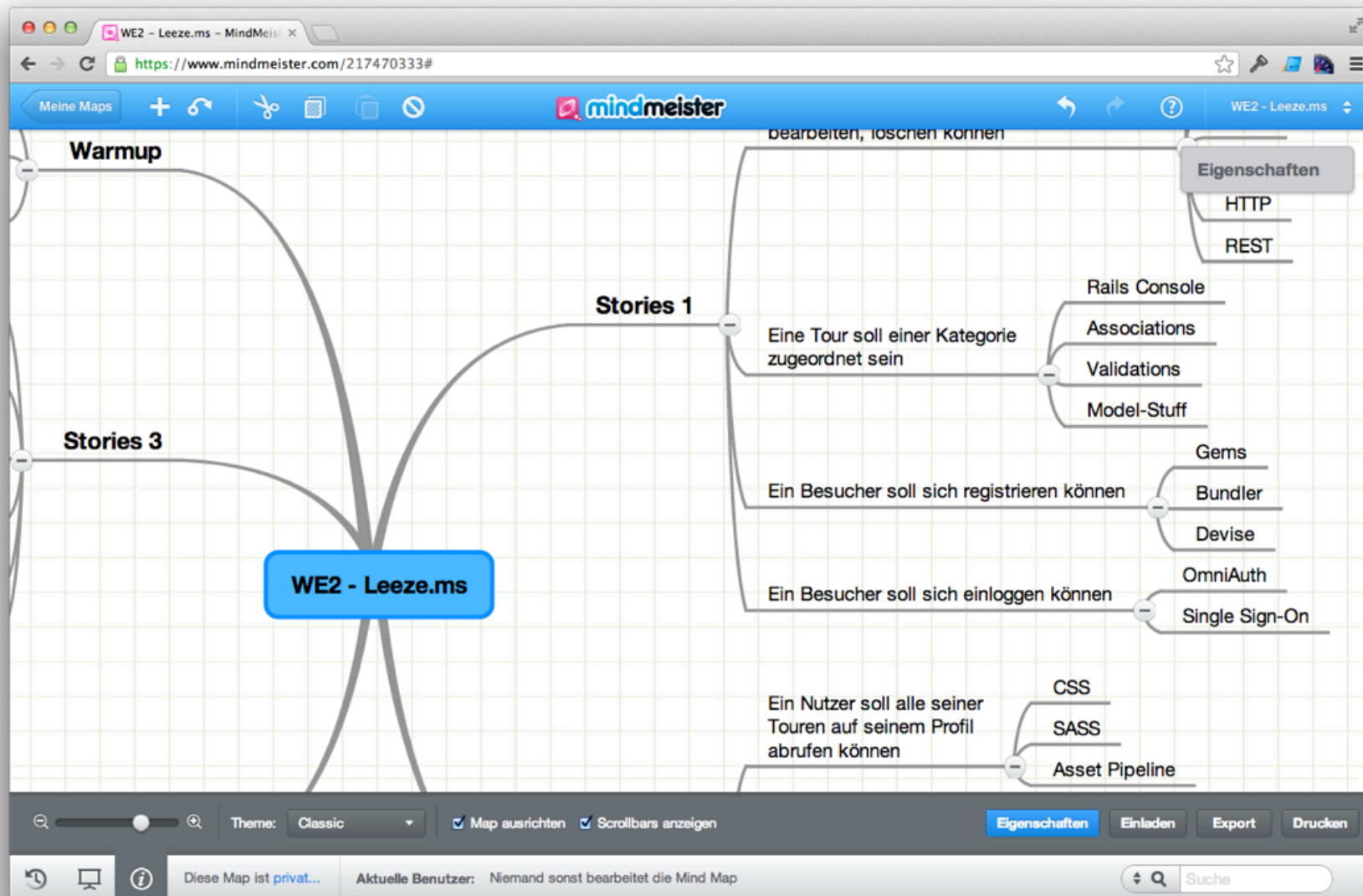
- Wahl der Projektvorgehensweise (z.B. agil)
- Definition MVP
- Umsetzung MVP
- Deployment / Akzeptanztest
- Iterative Weiterentwicklung



# Toolchain

- [mindmeister.com](http://mindmeister.com) - Brainstorming/Strukturierung
- [basecamp.com](http://basecamp.com) - Projektkommunikation
- [pivotaltracker.com](http://pivotaltracker.com)/[agilezen.com](http://agilezen.com) - Projektmanagement
- [github.com](http://github.com) - Source Code Management
- [campfirenow.com](http://campfirenow.com) - Echtzeitkommunikation (unter Entwicklern)
- [dropbox.com](http://dropbox.com) - Filesharing

# Toolchain



# Fazit

**Eine iterative Vorgehensweise und die Beachtung der Lean Start-Up Prinzipien hilft bei der Steuerung von Projektrisiken und führt nachhaltiger zum Erfolg.**