

# 80/20 Softwareentwicklung als Erfolgsfaktor

Dresden, 6. November 2017



Finanziert durch:



GERMANY  
AT ITS  
BEST



---

NORDRHEIN-WESTFALEN

Video: <https://www.youtube.com/watch?v=5ozkFMoB3oM>

# MATERIALS PURCHASING DIRECTLY ONLINE ist ein B2B-Startup für (Stahl-)Werkstoffe



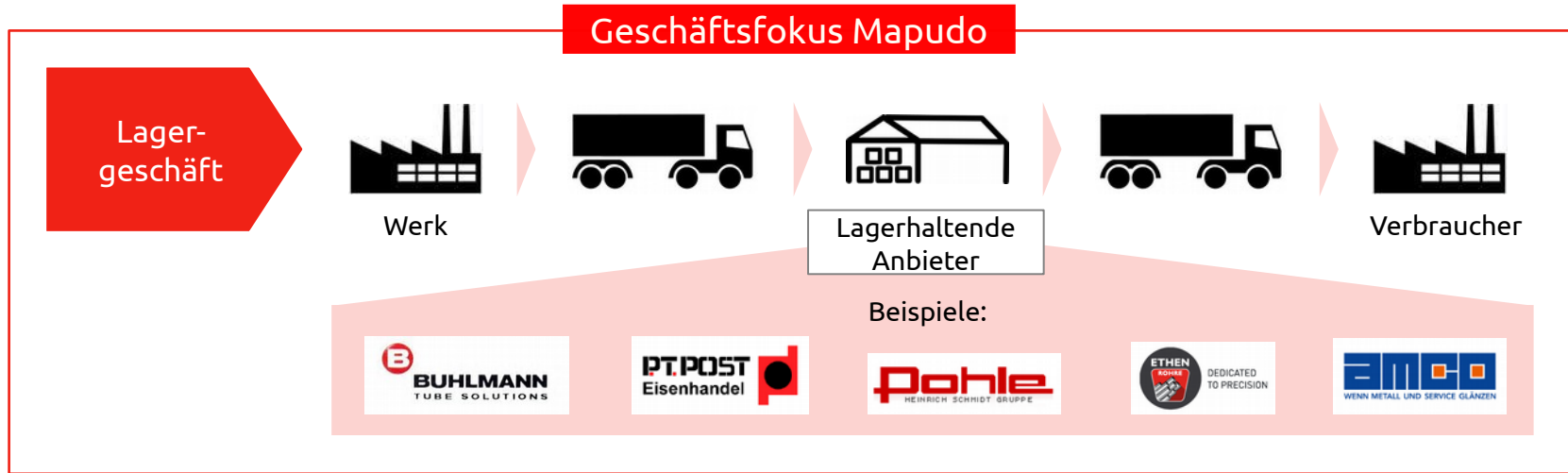
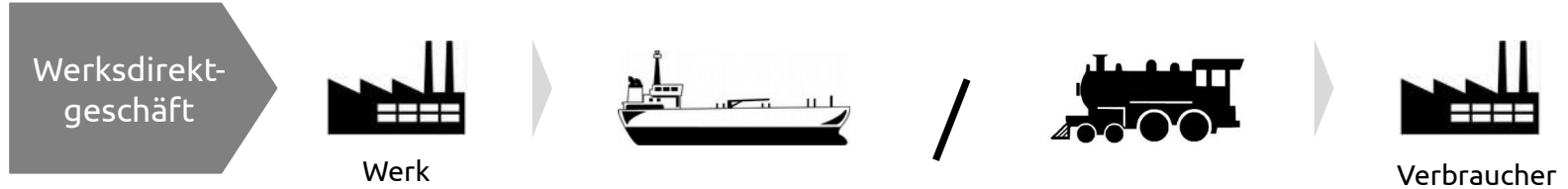


# Markus Weiland, CTO bei Mapudo



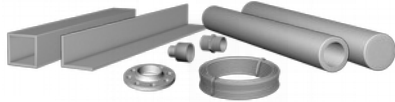
- 2008 Dipl.-Medieninf., TU Dresden
  - Schwerpunkte: Medientechnik, Entwicklungsmethoden und -werkzeuge
  - Nebenfach: BWL
- 2008-2015 in Montreal, Kanada
  - Software für
    - Mobile Endgeräte
    - Fluglotsen
    - Wach- und Sicherheitsdienste
  - Versuch, eigenes Startup aufzubauen
- Seit 2015 in Düsseldorf für Mapudo

# Übersicht Stahlvertrieb



# (Stahl-)Werkstoffe sind keine Katalogartikel

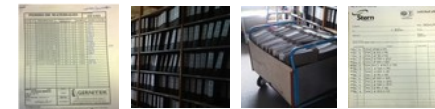
Merkmalsbasierte  
Produkteigenschaften



Komplexe Preisgebung



Regionale Teilmärkte



Manuelle Vertriebs-  
prozesse

# Kundennutzen zentral für Erfolg der digitalen Transformation

## Beispielhafte „Offline“-Branchen

Reise



Unterhaltung

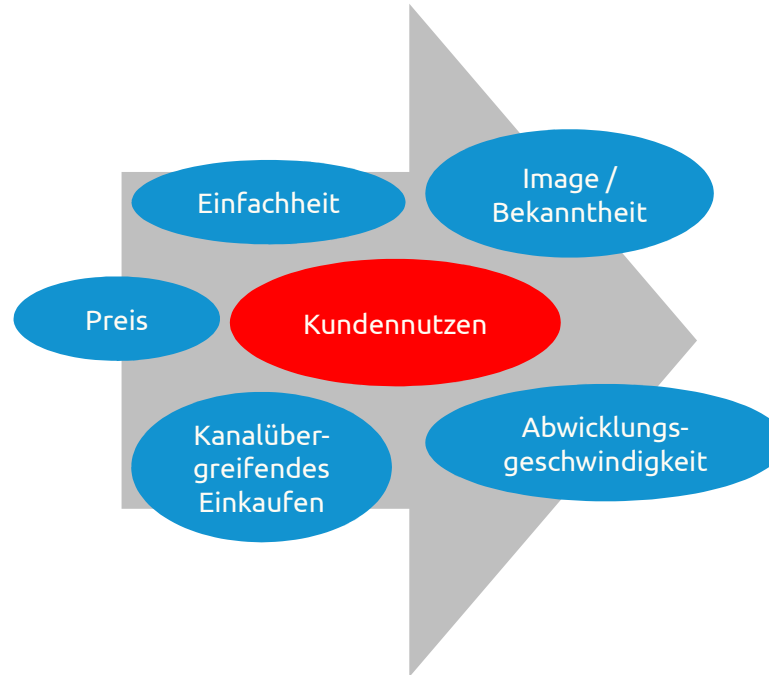


Taxi



...

## Transformationsphase / Erfolgsfaktoren



## Neue Online-Marktsituation



NETFLIX





# Was ist ein Startup?

- „Ein Startup ist eine **temporäre** Organisation auf der Suche nach einem **Geschäftsmodell**, das **wiederholbar, skalierbar** und **profitabel** ist.“
- “Ein Startup ist eine Organisation welche **neue Produkte** oder **Lösungen** unter Umständen **extremer Ungewissheit** entwickelt.”

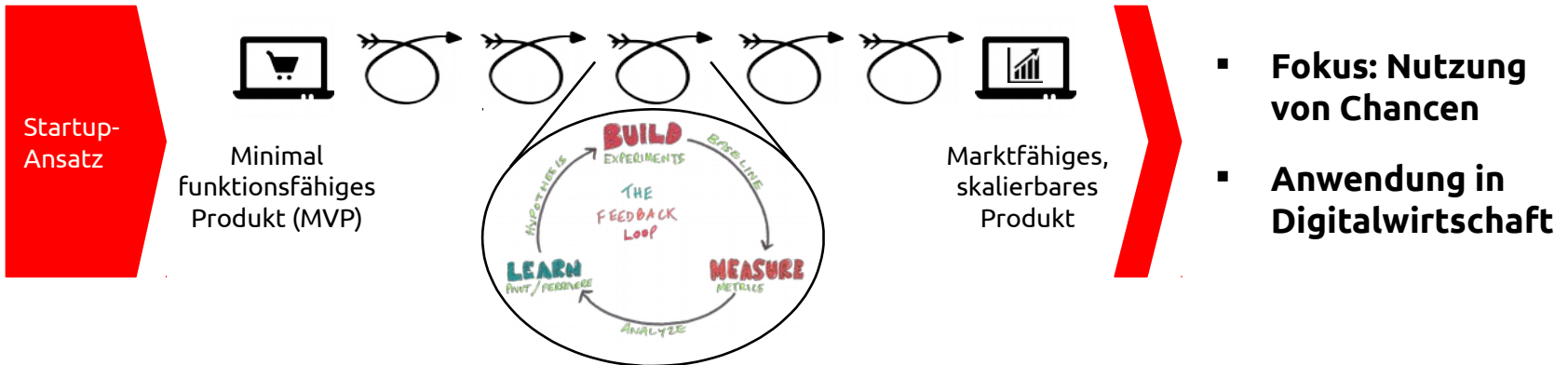


**Großes  
Potenzial,  
hohes Risiko**

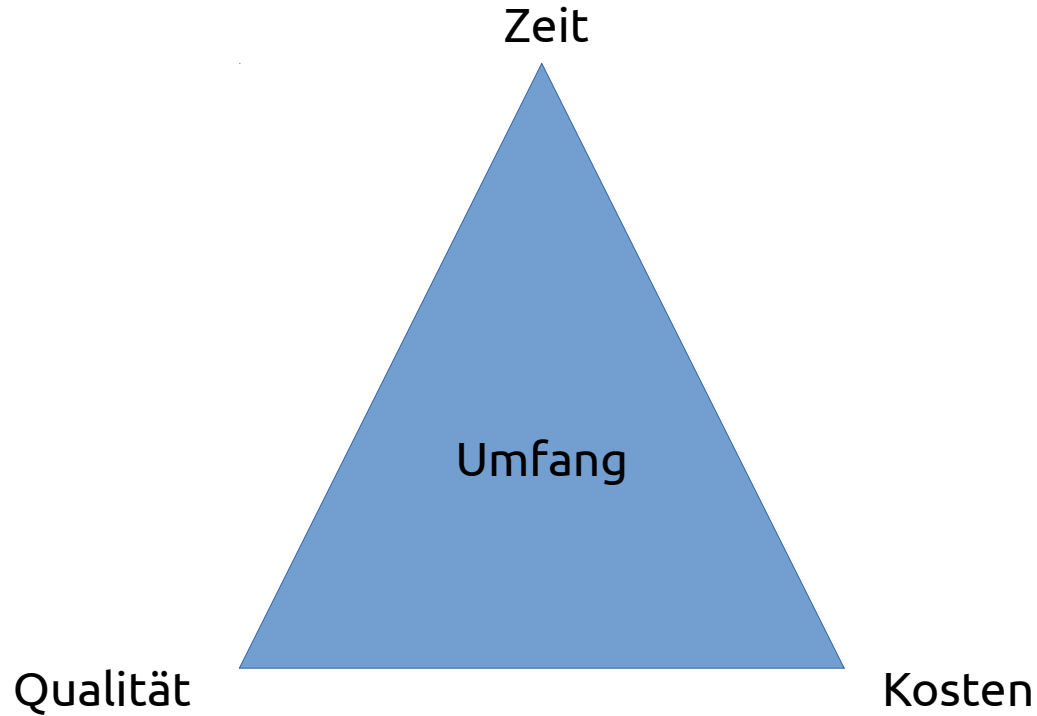
**Agile  
Umsetzungs-  
methoden**

- Eric Ries, *The Lean Startup*

# Der Startup-Ansatz zur iterativen Realisierung neuer Geschäftsmodelle



# Welche Faktoren beeinflussen Produktentwicklung?



# Einschränkungen im Kontext eines B2B Startups

- Finanzielle Mittel (Lebenszeit des Unternehmens)
- Verfügbare Arbeitskapazität
- Wissen / technisches Verständnis der
  - Mitarbeiter
  - Benutzer / Kunden
- Verfügbare Technologie
  - Auch hier: Zeit, Kosten, Qualität

# Abwägung der geplanten Produkteigenschaften

- In Bezug auf
  - Einschränkungen (i.d.R. bekannt und quantifizierbar)
  - Erwartete (i.d.R. nicht vollständig bekannte und nicht immer quantifizierbare)
    - 1. Stabilität der Anforderungen
    - 2. Bedeutung für das Unternehmen
    - 3. Iterationsgeschwindigkeit
    - 4. Umsetzungsdauer

# 1. Stabilität der Anforderungen

- Fiktives Beispiel: Zwei Jahre Finanzierung des Unternehmens
- Neue Funktion soll innerhalb von 10% der Lebenszeit des Unternehmens fertig sein
- Folglich von erster Idee bis zu Auslieferung 48 Arbeitstage
- Bei 1/3 Anteil der Planungsphase nur 16 Arbeitstage, das Problem zu verstehen und zu spezifizieren
- In B2B aber oft
  - wochenlange Vorplanung für Gesprächstermine
  - komplexe Prozesse, die abgebildet werden müssen
- Anforderungen können demnach nie in allen Details erfasst werden
- Instabile Anforderungen also unvermeidbar

# Warum und wie verändern sich externe Anforderungen?

- Bestehende Kunden besser bedienen
  - Bestehende Funktionen passender umsetzen
  - Neue Funktionen (z.B. Gehrungsschnitt)
  - Einfachere Benutzung (UX)
- Neue Partner und Kunden
  - Andere inhaltliche und technische Vorkenntnisse (z.B. Metallbauer vs. Pferdehof)
  - Bedarfe, Ansprüche (z.B. IT-Sicherheit bei Konzernen)
- Geplante Kooperationen kommen nicht zu Stande
  - Anforderungen entfallen
  - Technische Einschränkungen entfallen
- Veränderter Wettbewerb
  - Neue nicht-funktionale Anforderungen (z.B. Suchmaschinen-Auffindbarkeit ("SEO"), Server-Antwortzeit)

# Warum und wie verändern sich interne Anforderungen?

- Neue Mitarbeiter
  - Vorkenntnisse
  - Technische Fähigkeiten
    - z.B. Regular Expressions zur Bildzuordnung?
- Wachsende Integration mit anderen Geschäftsprozessen
  - CRM (Customer Relationship Management)
  - PIM (Product Information Management)
- Steigende Komplexität durch neue Funktionen
  - Erfordert Überarbeitung der internen Werkzeuge



# USS McCain collision ultimately caused by UI confusion

CO ordered duties of helmsman split—but all of them got sent to another console.

SEAN GALLAGHER (US) - 2/11/2017, 16:41



Quelle: <https://arstechnica.co.uk/information-technology/2017/11/uss-mccain-collision-ultimately-caused-by-ui-confusion/>

## 2. Bedeutung für das Unternehmen

- Nicht alle Anforderungen sind gleich
- Bestimmte Funktionen sind für das Fortbestehen des Unternehmens unabdingbar
  - Gewinnung von Partnern
  - Erreichen der gesetzten Unternehmensziele, die Bedingung für Finanzierung sind

# 3. Iterationsgeschwindigkeit

- Kleinsten Zeitraum von Idee bis Lernprozess
- Wie schnell kann die erste Umsetzung einer Anforderung anhand neuer Erkenntnisse korrigiert und verbessert werden?
- Welche Geschwindigkeit ist technisch realisierbar im Rahmen der Einschränkungen eines Startups?
- Aufwand steigt mit Verkürzung der Iterationszyklen, aber schnelle Iteration erlaubt, schneller den Weg zum funktionsfähigen Geschäftsmodell zu finden



# Implikationen der Iterationsgeschwindigkeit

- Mögliches Vorgehen im Software-Auslieferungsprozess in Bezug auf verschiedene Iterationslängen

Iterationslänge	Bau	Test	Freigabe	Auslieferung
Monate bis Jahre	Manuell	Manuell	Manuell	Manuell
Wochen bis wenige Monate	Teilautomatisiert	Teilautomatisiert	Manuell	Manuell
Tage bis Wochen	Automatisiert	Teilautomatisiert	Manuell	Teilautomatisiert
Stunden bis Tage	Automatisiert	Automatisiert	Automatisiert	Automatisiert

# 4. Umsetzungsdauer

- Messbare Ergebnisse müssen vor nächster Finanzierungsrunde vorliegen
  - In anderen Worten: Jede neue Funktion, deren Auswirkung erst nach der nächsten Finanzierungsrunde messbar ist, schadet dem Unternehmen
- Anforderungen müssen also immer herunter gebrochen werden auf kleine Schritte, deren Auswirkungen rechtzeitig messbar sind

# Beispiel: Verkauf von 2A Material

- Projekt: Verkauf von "B-Ware" per Auktionsmodell
- Umsetzung für einen neuen Investor
- Nachfolgend die Revision der Anforderungen (Iterationen unterschiedlicher Länge)

# Iteration 1

- Kundenanlage mit Anbindung an ERP
- Benachrichtigung per E-Mail bei Aktualisierung Sortiment
- Schnittstellen zum Upload von neuen Produkten
- Übergabe Bestelldaten an ERP
- Mehrsprachigkeit (DE, FR, EN, IT)
- Definierbare Reihenfolge der Angebotsabgabe (Bestandskunden zuerst)
- Reservierung von Paketen
- Reservierungszeitraum festlegbar
- Bestätigung aus ERP System heraus

# Iteration 2

- ~~Kundenanlage mit Anbindung an ERP~~ Selbstregistrierung mit Bestätigung
- ~~Benachrichtigung per E-Mail bei Aktualisierung Sortiment~~ Manuelle Benachrichtigung
- ~~Schnittstellen zum Upload von neuen Produkten~~ Manueller Upload durch Mapudo
- ~~Übergabe Bestelldaten an ERP~~ Erweiterung der bestehenden E-Mails um relevante Identifikatoren, danach händische Abarbeitung
- ~~Mehrsprachigkeit (DE, FR, EN, IT)~~ Auswahl von Kunden, die deutsche Plattform bedienen können
- ~~Definierbare Reihenfolge der Angebotsabgabe (Bestandskunden zuerst)~~
- ~~Reservierung von Paketen~~ Verwendung von Warenbestandslogik mit Anfangsmenge 1
- ~~Reservierungszeitraum festlegbar~~
- ~~Bestätigung aus ERP System heraus~~ Bestätigung per Hand via bestehender Mapudo Funktionen



# Iteration der Auktion im Detail 1

- Analyse: Keine klassische Auktion von immer einem Produkt, sondern Preisvorschlag pro Position, mit Möglichkeit für Anbieter abzulehnen
- Implementierung:
  - Kopie der bestehenden Funktion "Längenzuschnitt" und Umbau zu "Preiszuschnitt", also Preisvorschlag
  - Bestandsmenge pro Produkt muss nun gespeichert werden für Produkte, die über Auktion verkauft werden
  - Reduzieren des Produktbestands, sobald Auftrag angenommen
  - Ablehnen von Aufträgen bereits vorhanden
- Ergebnis: "Ja, aber wir haben zum Test einen Vorschlag eingeschickt und das Produkt ist trotzdem noch zu finden?!"

# Iteration der Auktion im Detail 2


- Analyse: Anbieter will immer nur einen Vorschlag auf einmal und gar keine Auktion
- Implementierung:
  - Umbau aller Events, wann Produktbestand dekrementiert / inkrementiert wird
  - Jetzt zahlreiche neue Race Conditions (z.B. Auftrag liegt vor, ist aber noch nicht geprüft und neue Produktdaten werden hochgeladen (Excel Liste), also Bestand wird überschrieben (Absicht), dann wird Auftrag abgelehnt)
- Ergebnis: "Funktional nutzbar, aber praktisch nicht einsetzbar"

???

# Iteration der Auktion im Detail 3

- Analyse: Endkunden kaufen gar nicht nach Preis pro Position, sondern nach Preis pro Gewicht
- Implementierung:
  - Erweiterung des Warenkorb-Codes mit Darstellung nach Preis pro Gewicht
  - Zahlreiche Abkürzungen in der Implementierung, da mittlerweile Zeitdruck
- Ergebnis: Freigabe zur Benutzung durch Kunden (Käufer)

# Iteration der Auktion im Detail 4

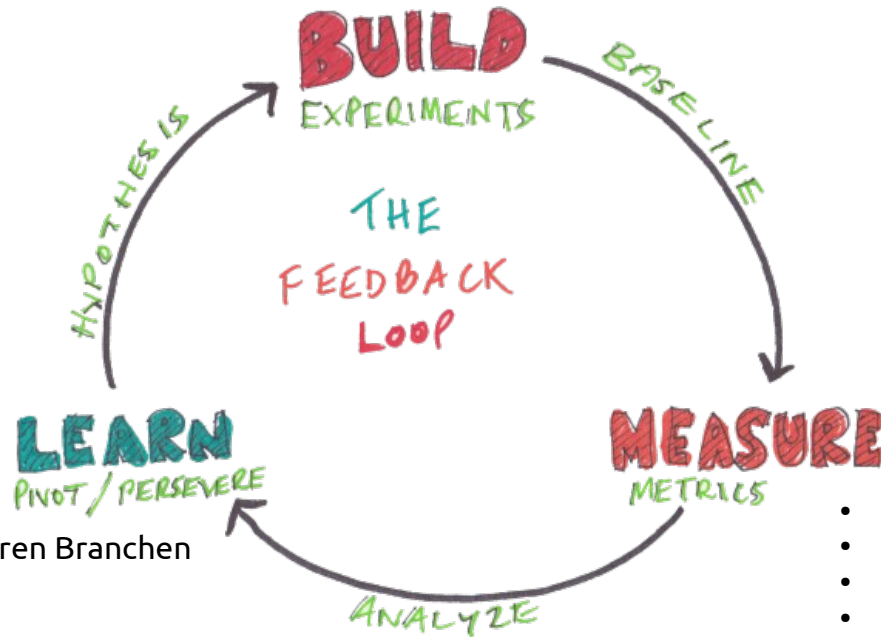
- Analyse: Käufer sagt: "Unbenutzbar, wir kaufen nach Preis pro Tonne, nicht pro Kilogramm"
- Implementierung:
  - 
  - Erweiterung der Benutzer um Einstellung, in welcher Maßeinheit nach Gewicht eingekauft wird
- Ergebnis: Benutzbar, aber einer der ersten Käufer schickt einen Auftrag mit Preisvorschlägen für 60 Positionen (bisher selten >10 Positionen) und Auftrag wird abgelehnt -> Der Käufer muss sich 60 Positionen wieder einzeln in den Warenkorb legen

# Iteration der Auktion im Detail 5

- Analyse: Käufer benötigt Möglichkeit, einen alten Vorschlag nochmals in den Warenkorb zu überführen
- etc. etc.

# Beispielhafte Arbeitspakete einer Iteration

- Spezifikation
- UI, UX
- Mock-Ups oder Implementierung



- Diskussion
- Vergleiche mit anderen Branchen
- Kreativität

- Vorträge
- Demos
- Benutzerstudie
- Analytics

# Rolle der IT in einem Software-Startup

- Umsetzung der Anforderungen
  - IT ist neutral gegenüber dem Produkt -> sachliche Diskussion
- Anforderungen stammen aus allen Bereichen des Unternehmens
- i.d.R. auf dem kritischen Pfad
  - d.h. zentrales Element bei der unternehmensweiten Vorgehensplanung
- Bereitstellung weiter Teile der digitalen Infrastruktur
  - Von WLAN bis Loadbalancer
- Ohne sehr gute Entwickler kein Erfolg
  - Offene Anforderungsdetails müssen aus Eigenantrieb eingeholt werden
  - Sehr breites Aufgabengebiet, auch neben der reinen Entwicklung (von "nebenbei drei Windows Laptops aufsetzen" bis "wie konfiguriere ich meine Frontend Build Tools so, dass nur für die Produktivumgebung minimiertes JavaScript ausgeliefert wird?")

# Was lieben / hassen Entwickler?

- Zufriedene Entwickler durch
  - Ticketsystem als einzige Wahrheit (z.B. JIRA)
    - Klar definierter Ticket-Workflow, z.B. To Do -> In Progress -> To Review -> Done
  - Kein Kapern von Entwicklern
    - Ohne Ticket kein Code, ohne Absprache mit Produktmanager kein Ticket
  - Wertschätzung der Arbeit
  - Keine Überstunden durch schlechte Planung
  - Klar vereinbarte Regeln mit anderen "Abteilungen" vermeiden Konflikte
- Dadurch wenig Mitarbeiterfluktuation und volle Leistung



# Wie viele Mitarbeiter benötigt man in der IT?

- Anders herum: Wie viele Mitarbeiter kann man verantworten?
  - Abhängig von
    - der Unternehmensphase
    - finanziellen Mitteln
- Zu Beginn:
  - Architekturelle Entscheidungen
  - interne vs. externe Umsetzung
- Später:
  - Verschiebung hin zu enger definierten Aufgabenbereichen
- Gesuchtes Mitarbeiterprofil verändert sich im Verlauf der Zeit

# Wo tritt Entwicklungsaufwand häufig auf?

- Architecturally Significant Requirements
  - Von 0 auf 1
    - Erstimplementierung der Anforderungen (Datenstrukturen, Logik, Tests)
  - Von 1 auf n
    - Umstellung aller Referenzen von Instanzen auf Arrays
      - Probleme: z.B. Typsicherheit, Datenbankmigrationen
    - ggf. festzulegende Priorisierung / Reihenfolge
  - Neue nicht-funktionale Anforderungen, z.B. Antwortzeit, Verfügbarkeit

# Implementierung in der Praxis

- Arbeitsteilung (z.B. Frontend, Backend)
- Zerlegung der Software in Komponenten (Dekomposition)
  - Klare Verantwortungen
  - Code ist (automatisiert) testbar und wird getestet
  - Bei Änderung der Anforderungen müssen oft nur einzelne Komponenten geschrieben oder ausgetauscht werden
- Möglichst klare Akzeptanzkriterien

# Nützliche OOP Patterns in der Implementierung

- Strategy Pattern (z.B. Voter bei Zahlungsabwicklung, E-Mail Generatoren)
  - Strategy lets the algorithm vary independently from clients that use it. [1]
- Visitor Pattern (z.B. "Adjuster" bei Preisgebung)
  - Visitor lets you define a new operation without changing the classes of the elements on which it operates. [2]
    - There are many distinct and unrelated operations needed to be performed.

[1] <http://www.oodesign.com/strategy-pattern.html>

[2] <http://www.oodesign.com/visitor-pattern.html>

# Fazit

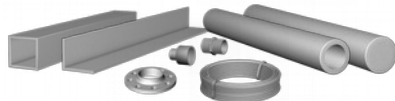
- Mit 20% der Anforderungen deckt man i.d.R. 80% der tatsächlich verwendeten Funktionen ab
- Erfolgreiches Vorgehen in einem B2B Startup bedeutet, die richtigen 20% zum richtigen Zeitpunkt mit den richtigen Mitarbeitern umzusetzen
- Den Rest im Verlauf erledigen, sobald konkreter Bedarf identifiziert
- Durch passende interne Organisation und Planung werden Konflikte mit "der IT" vermieden
- Sehr gute und gleichzeitig zufriedene Entwickler sind Voraussetzung für ein erfolgreiches Software-Startup

# Interesse an Mapudo?

- Praktikum
- Festanstellung

Sprecht mich an oder  
schreibt mir unter

[markus.weiland@mapudo.com](mailto:markus.weiland@mapudo.com)



# WE STEEL

## Industrial Materials Marketplace

**Markus Weiland**

Head of Technology

E-Mail: [markus.weiland@mapudo.com](mailto:markus.weiland@mapudo.com)

Mapudo GmbH  
Erkrather Straße 162  
40233 Düsseldorf

Registered office and commercial register:  
Düsseldorf, HRB 73002  
Managing Directors: Sebastian Grethe, Niklas Friederichsen