Agilität ist mehr als Scrum

Thomas Much
@thmuch

#SemicolonGFU

31.05.2016

Wer spricht heute Abend?

- **Thomas Much**
- Freiberuflicher Dipl.-Informatiker aus Hamburg

- Entwickler & Architekt (Java et al.)
- Agile Developer Coach

- Div. Branchen (E-Commerce, Montan u.a.)
- Div. Firmengrößen (Großkonzern, Mittelstand)

Wer spricht heute Abend?

- **Thomas Much**
- Freiberuflicher Dipl.-Informatiker aus Hamburg

- Entwickler & Architekt (Java et al.)
- Agile Developer Coach

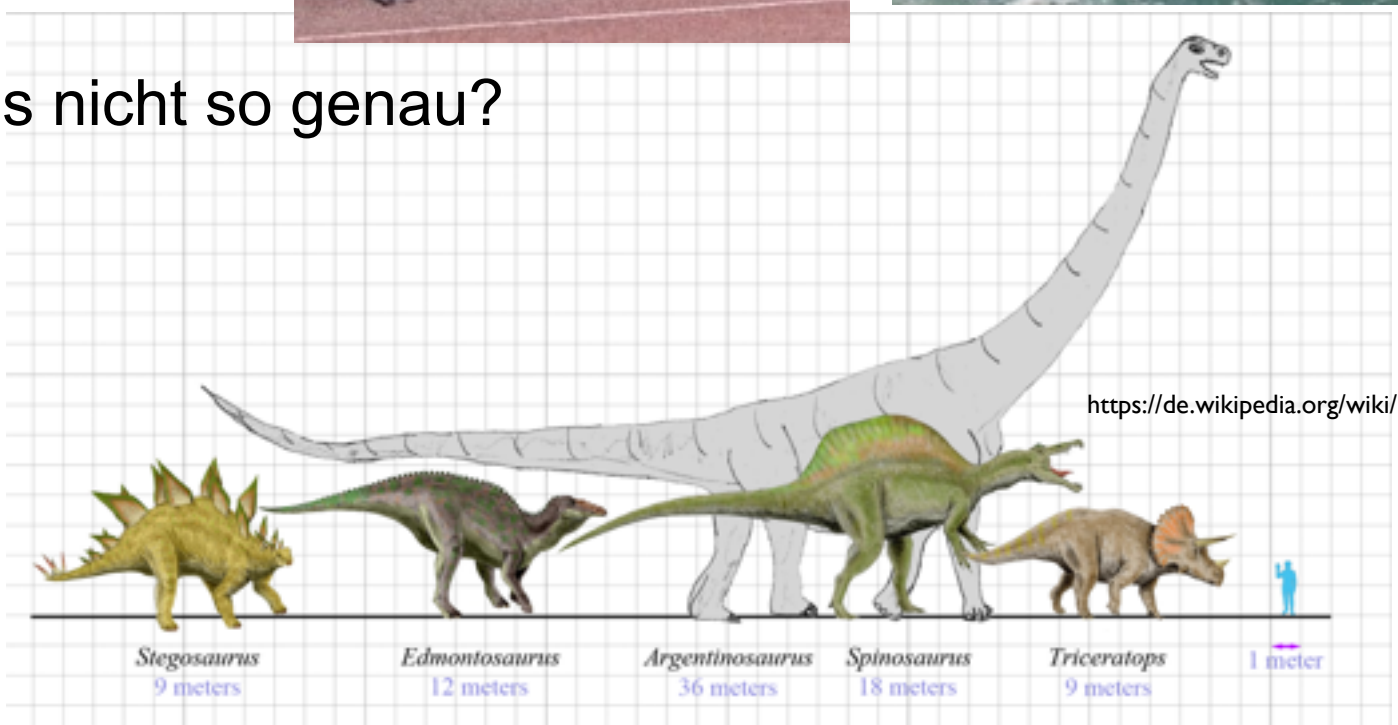
- Div. Branchen (E-Commerce, Montan u.a.)
- Div. Firmengrößen (Großkonzern, Mittelstand)
- Div. Vorgehensweisen bzw. Prozesse

Umfrage (1)

https://en.wikipedia.org/wiki/Running

https://de.wikipedia.org/wiki/Wasserfall

- Wer ist agil?
- Wer bewusst *nicht*?
- Wer weiß das nicht so genau?



https://de.wikipedia.org/wiki/Dinosaurier

Umfrage (2)

- Wer ist zufrieden mit seiner Softwareentwicklung?
- Wer möchte etwas ändern? Was?

Was unter "agil sein" verstanden wird …

"Wir sind agil, wir machen Scrum"

Was unter "agil sein" verstanden wird …

"Wir sind agil, wir machen Scrum"

Ok, nicht ganz nach Lehrbuch, wir haben uns das beste rausgesucht.

Wir machen Daily Standup Meetings (aber nicht jeden Tag).

Ok, die Standup-Meetings sind viel zu lang und zu langweilig...

Aber wir haben unsere JIRA-Tickets am Board!

Was unter "agil sein" verstanden wird …

"Wir sind agil, wir haben JIRA"

Was unter "agil sein" verstanden wird …

"Wir sind agil, wir haben JIRA"

*Ok, das Board sieht aus wie S**, da hat leider niemand den Überblick...*

Wer priorisiert unsere Aufgaben eigentlich?

Darf man mehr als 10 Tickets gleichzeitig in Arbeit haben?

Was unter "agil sein" verstanden wird …

"Wir sind agil, wir haben Jenkins"

Was unter "agil sein" verstanden wird …

"Wir sind agil, wir haben Jenkins"

Jeden Dienstag wird nachts ein Build angetriggert und eine Programmdatei gebaut, die wir dann an Ops zum Deployment geben.

Ok, man muss aktiv die Jenkins-Seite aufrufen, um den aktuellen Build-Status zu erfahren.

Wie, Jenkins kann Code-Metriken ermitteln und anzeigen?

Klarstellung

- Scrum ist eine gute Vorgehensweise!
- JIRA ist ein gutes Werkzeug!
- Jenkins ist ein gutes Werkzeug!

Klarstellung

- Scrum ist eine gute Vorgehensweise!

  … *wenn sie zum Projekt / zur Abteilung / … passt.*

- JIRA ist ein gutes Werkzeug!

  … *wenn es richtig eingesetzt wird.*

- Jenkins ist ein gutes Werkzeug!

  … *wenn es sinnvoll genutzt wird.*

Auf der Suche nach der Agilität

- Viele, viele Hilfsmittel, die Agilität unterstützen.

- Aber was bedeutet es, "richtig" agil zu sein?
- Wann stellt Agilität in der Softwareentwicklung einen Mehrwert dar?

- **Machen wir uns auf die Suche!**

  Ohne Anspruch auf Vollständigkeit.
  Dafür mit persönlichen Erfahrungen, Ideen
  – und hoffentlich reichlich Gesprächsstoff :-)

Was bedeutet "agil sein"?

- "Agil" ist derzeit die Lösung für viele (alle) Probleme …

- Für viele ist *"agil"* = *"schneller"*

  Im Sinne von: Dieselbe Arbeit wird schneller erledigt
  (= kurzfristig weniger Kosten)

Was bedeutet "agil sein"?

- "Agil" ist derzeit die Lösung für viele (alle) Probleme …

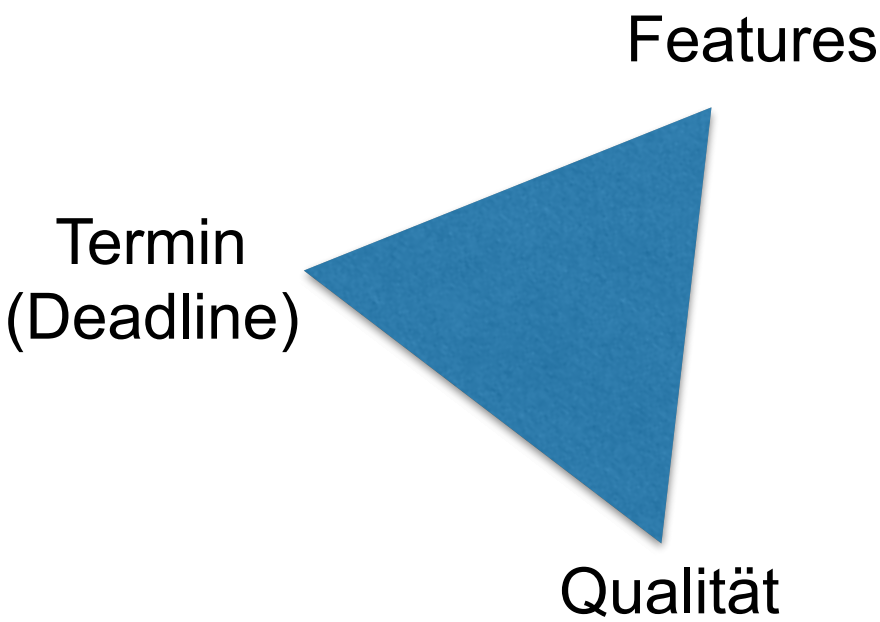
- Für viele ist *"agil"* = *"schneller"*

  Im Sinne von: Dieselbe Arbeit wird schneller erledigt
  (= kurzfristig weniger Kosten)

  Funktioniert nicht.

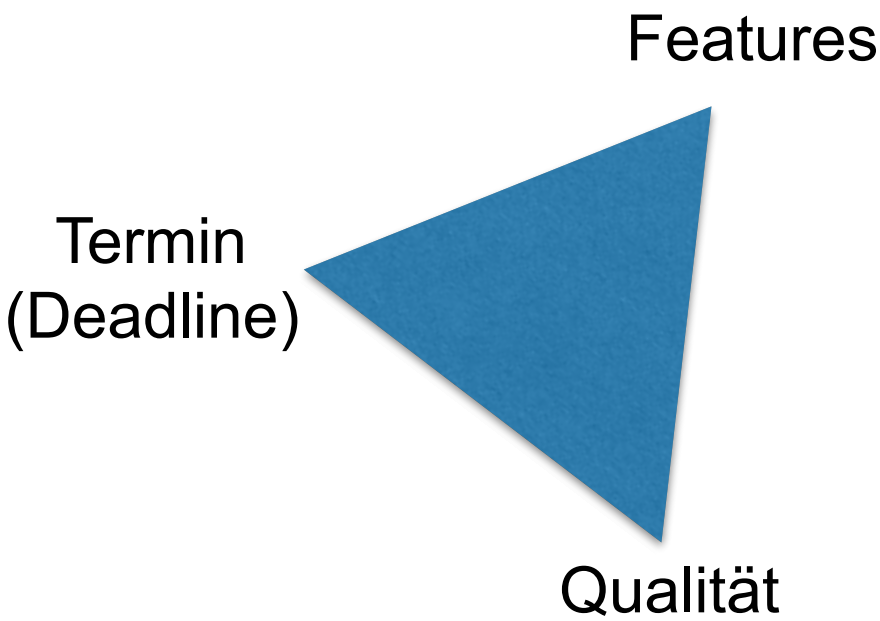
  Warum?

Herausforderungen bei der Softwareentwicklung



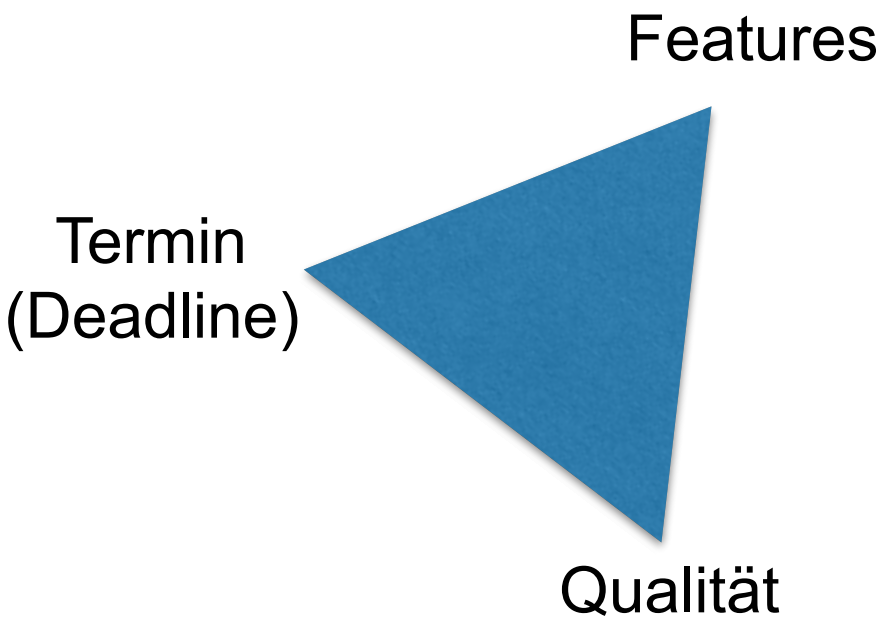
- Was wird eingehalten?
  - Wähle zwei.

Herausforderungen bei der Softwareentwicklung



- Was wird eingehalten?
  - Wähle zwei.
- Alternativ Kosten statt Qualität
  - D.h. Qualität kostet (kurzfristig)

Herausforderungen bei der Softwareentwicklung



- Was wird eingehalten?
  - Wähle zwei.
- Alternativ Kosten statt Qualität
  - D.h. Qualität kostet (kurzfristig)
  - Keine Qualität kostet langfristig …

Herausforderung Fachbereich

Ständig neue / wechselnde Anforderungen.

Ohne Wertung!

Der Kunde ist König.

Fachbereich muss auf Marktgeschehen reagieren.

Herausforderungen bleiben

- An diesen Herausforderungen kann auch Agilität nicht viel ändern.
- Aber Agilität kann helfen, besser mit den Herausforderungen umzugehen.

Was soll Agilität leisten?

- Schneller Features (& Bugfixes) *zum Kunden liefern.*
- Schnelleres Feedback
  (gerade auch bei längerer Entwicklung)
  um Fehlentwicklungen frühzeitig zu erkennen
  und zu vermeiden.
- Schneller auf *wechselnde Anforderungen* reagieren.
- Kurzfristige & langfristige Kundenzufriedenheit

Warum agil? (Management Summary)

Flexibilität

durch

- schnelles Feedback
- kleinere Feature-Pakete
- kleinere Deployment-Einheiten

(Keine kurzfristige Kostenoptimierung!)

Agilität braucht …

- Feedback & daraus lernen und besser werden
- Automatisierung
- Überschaubare Aufgaben(pakete), die fertig werden können
- Gute Visualisierung der Aufgaben und des Ist-Zustands

Und Fachbereiche, UX, Entwickler, QA, Ops …
… die das umsetzen wollen!

Agilität braucht … Feedback & daraus lernen

Nötige Werte & Rituale für

kontinuierliches Lernen aus Fehlschlägen & Erfolgen:

- **Ehrlichkeit, Offenheit, Mut, Vertrauen**
- Reviews, Retrospektiven, Post-Mortems
  - kein Finger-Pointing, keine negativen Konsequenzen!
  - mutig sein, Fragen zu stellen; Wissenslücken zugeben
  - sich selbst in Frage stellen lassen, andere hinterfragen dürfen
- Respekt & Vertrauenskultur! *(Nicht Command&Control)*

Eine ganz große AUFGABE!

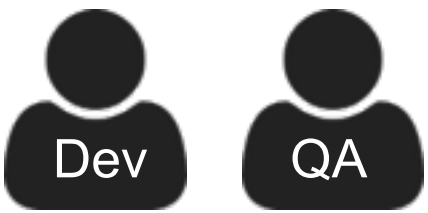
Agilität braucht … Automatisierung

- Build
- Testen (Unit, Integration, UI/Akzeptanz)
- Deployments
- Monitoring (inkl. Logfiles)
- Umgebung / Infrastruktur
- Datenbanken
- Dokumentation
- …

Agilität braucht … überschaubare Arbeitspakete

- Gut geschnittene User Stories & Tasks
  die iterativ und inkrementell entstehen

- Nicht zu große Commits
  mit Änderungen nur zu einem Ticket



- MVP (Minimum Viable Product)
  statt Release mit viel zu vielen Features

Agilität braucht … Visualisierung & Sichtbarkeit

- Analog: *Gute Boards. Haptisch, physikalisch, reell.*

  Gut strukturiert, Informationen erfrischend aufbereitet.

  Lieber mehrere Boards als ein eierlegendes Wollmilchboard.

- Digital: *Informations-Radiatoren*

  Build-Zustand, Software-Qualität,
  Laufzeit-Monitoring (auch für Entwickler!)

  Immer präsent statt Push (E-Mail) oder Pull.

Wo können wir Agilität fördern?

- Entwicklungsteam
- Deployments
- Infrastruktur
- Architektur

Wichtige Basis & ein paar Buzzwords …

Agilität im Entwicklungsteam

- Entwickler werden agil, wenn sie
  - kleine Arbeitspakete haben
  - Freiheit zur Selbstorganisation haben
  - Feedback bekommen (technisch & menschlich)
  - schnell entwickeln können (Hardware!)
- Stichworte:
  - Continuous Integration, TDD, Pair Programming
- Braucht
  - geeignete agile Vorgehensweise *pro Team*

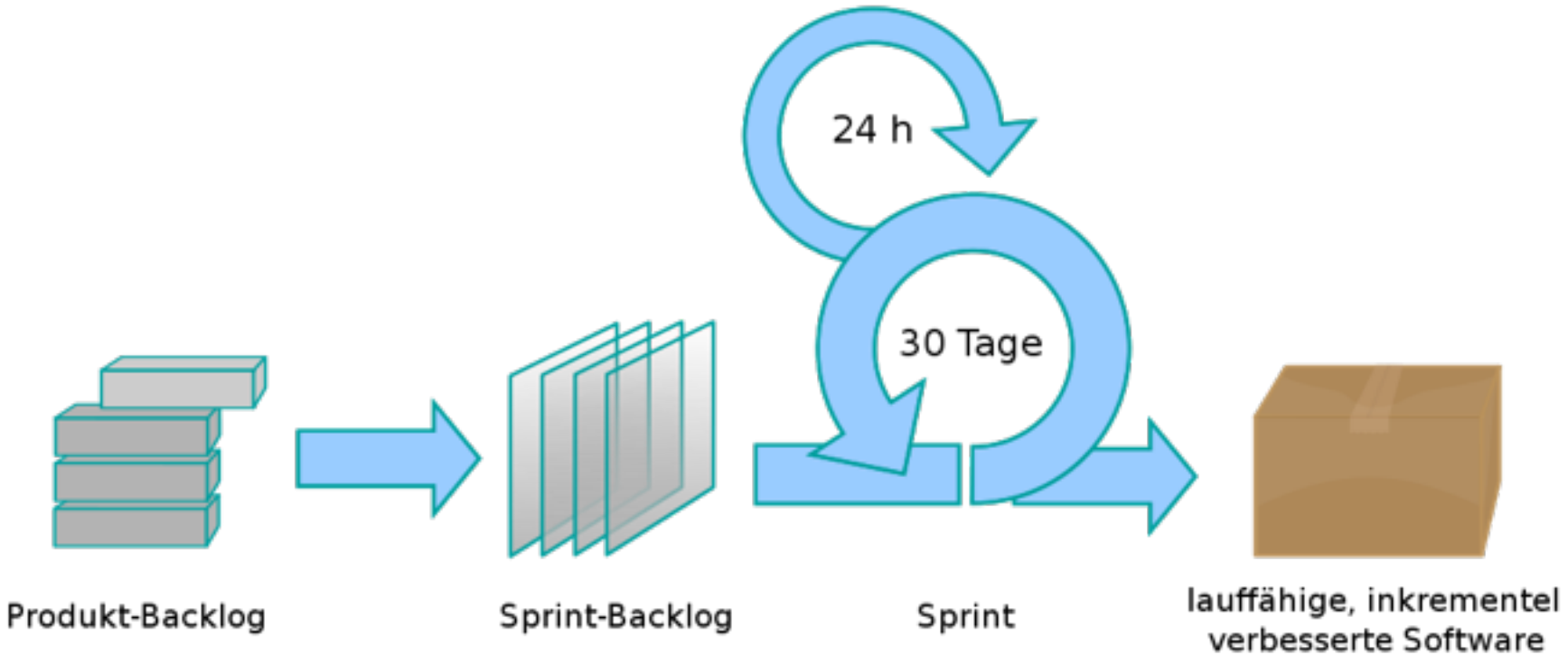
Agile Vorgehensweise: Scrum passt nicht immer

Scrum ist nur *eine* mögliche agile Vorgehensweise. Weitere:

- IT-**Kanban**
- **XP** (eXtreme Programming)
- …

- Passende Vorgehensweise auswählen!
  - Reinform (nach Lehrbuch) *nicht* zwingend notwendig.
  - Notwendig: *Ständige Verbesserung der eigenen Umsetzung!*
  - Pro Team.

Agile Vorgehensweise: Scrum



https://de.wikipedia.org/wiki/Scrum

- Regelmäßiges Feedback durch Sprints (Iterationen).
- Fokus: Lauffähige Features zum Kunden ausliefern.

Agile Vorgehensweise: (IT-)Kanban

https://de.wikipedia.org/wiki/Kanban-Tafel

- Work in Progress ("Doing") begrenzen
- Fokus: Durchlaufzeiten minimieren

Agile Vorgehensweise: eXtreme Programming (XP)

- Vielleicht die umfassendste agile Vorgehensweise für Softwareentwicklung
- Sehr konsequent; als Einstieg für Wasserfall-Firmen evtl. schwieriger als Scrum

https://de.wikipedia.org/wiki/Extreme_Programming

- XP-Praktiken aber extrem relevant:
  - **Continuous *, TDD, Pair Programming**, Iterationen, Stories …
  - Feedback & Kommunikation :-)

XP: Code im Fokus

- Besinnung auf die Handwerkskunst der Programmierung: *"Software-Craftsmanship"*
- Nicht das Vorgehen ist Garant für guten Code, sondern *guter Code ermöglicht einen ordentlichen Ablauf.*

"Was nützt der beste Architekt,

wenn die Maurer so schlecht mauern,

dass ständig nachgebessert werden muss?"

Pair Programming

"Convinced team 2 make **pair programming** - finally.

It's like everything changed in days.

Had more impact than DoD / DoR, Code Review or Rules"

https://twitter.com/HenningSpille/status/733298682016235520

Agilität bei Deployments

- Continuous Delivery
  Automatisierung, aber teilweise manuelle Trigger
- Continuous Deployment
  Vollautomatische Deployments bis in die Live-Systeme
- Stichworte:
  - Pipelines, DevOps
- Braucht
  - gute Testautomatisierung, Golden Binaries, Trunk-Release mit Feature-Toggles etc.

Agilität bei der Infrastruktur

- Umgebungen reproduzierbar auf Knopfdruck aufbauen und abreißen
- Live-ähnliche Umgebungen für Entwickler!

- Stichworte:
  - Infrastructure as Code
  - Virtualisierung, Container, Docker

Agilität bei der Architektur

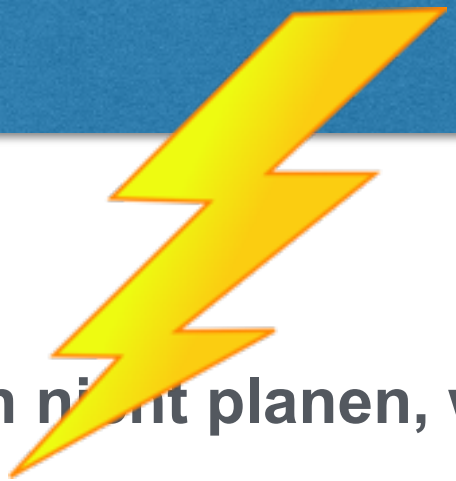
- Technologien ändern sich recht schnell …
- Architektur muss auf *Ersetzbarkeit* ausgerichtet sein
- Team muss Architektur *bei Bedarf* ändern können (!)

- Braucht
  - gute Testautomatisierung
  - Architekten (?) als Team-Player, Servant Leader, Entwickler
- Stichworte:
  - Microservices statt Monolithen… Oder irgendwas dazwischen.
  - Aber: Continuous Delivery kommt vor Microservices!

Agilität & Planung

"Wir müssen nicht planen, wir sind agil"

Agilität & Planung



"Wir müssen nicht planen, wir sind agil"

- Agile Entwicklung ist mittelfristig nur gut mit agiler Anforderungsbeschreibung!

- Stichworte:
  - Epics, Stories, Tasks, User Stories, User Story Mapping
- Braucht
  - Zusammenarbeit von Fachbereich und Dev-Team
    und nicht "Stories als Dekret"

    http://martinfowler.com/bliki/ConversationalStories.html

Agilität & Selbstorganisation

- Dem Team Freiheiten lassen!

  Selbstorganisierte Teams erfordern Vertrauen.

- Aber einer muss wissen, wo die Reise hingeht.

  Ziel im Blick behalten, Aufgaben Priorisieren.

- **Kein Mikromanagement!**
- **Aber auch nicht *kein* Management.**

Zum Schluss noch ein paar Hinweise

"Das kaufen wir" aka "Agil in N Tagen"

- Agilität ist kein Tool, das man kaufen kann.*
- Agilität ist keine Schulung, die man besuchen kann.*

*) Tools können unterstützen, Schulungen können helfen.

Wichtig ist aber vor allem, dass im Unternehmen eine passende Kultur für Agilität aufgebaut wird.

(Gilt entsprechend für DevOps.)

Agilität als Kultur(wandel)

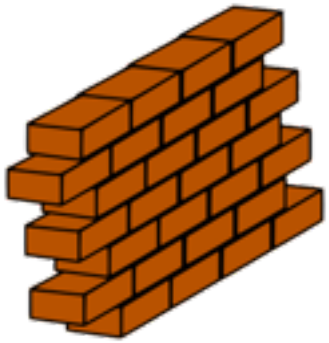
- **Agilität ändert Ihr Unternehmen!**

- Zusammenarbeit *vs.* Zaunmentalität
- Inkrementelle, häufige *vs.* seltene, sehr große Releases
- Ehrliches & korrektes Feedback (menschlich & technisch)
- …

- Braucht *Rückhalt* von ganz oben.
- Ist ein langer Weg, der nie ganz zu Ende sein wird.

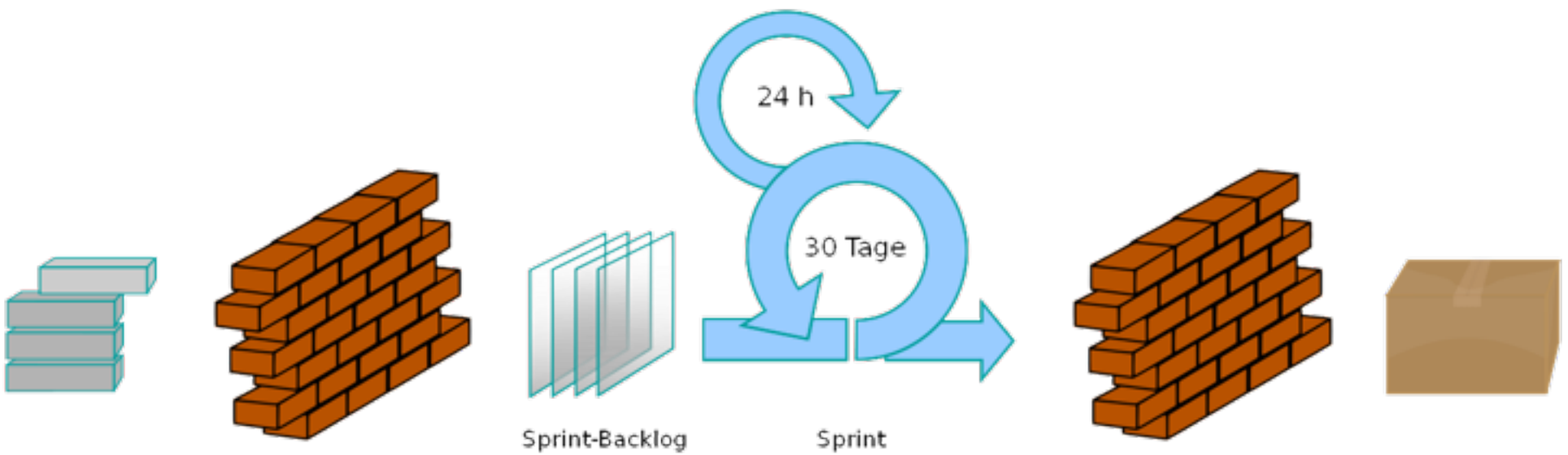
Agile Inseln: Guter Einstieg, langfristig problematisch

https://de.wikipedia.org/wiki/Hasen



https://en.wikipedia.org/wiki/Tortoise

- Vorsicht vor 2-Speed IT / Bimodal IT bzw. Water-Scrum-Fall als *Dauerzustand*
- Gefahr sehr reell, dass der nicht-agile Bereich den agilen ausbremst bzw. wieder nicht-agil macht!



Also: Wie wird man agil?

Also: Wie wird man agil?

- Nicht alles auf einmal machen! *
- Ein kleiner Anfang ist auch ok …
- … wenn *mutig Schritt für Schritt* weitergegangen wird.

*) Natürlich kann man eine komplette Abteilung auch *disruptiv* umstellen – das muss dann aber sehr konsequent mit allen nötigen Freiheiten, Vertrauen etc. erfolgen.

Mögliche Schritte (nicht zwingend sequenziell)

Dev-Teams agil machen

QA + Ops agil machen

Agilität (er)leben

Unternehmen agil machen

1. Lassen Sie die Entwickler handwerklich gut arbeiten:
   TDD, CI, Pair Programming, Visualisierung.

Nutzen Sie die XP-Praktiken

2. Implementieren Sie Continuous Delivery
   Automatisieren Sie die Infrastruktur.

Werden Sie "continuous"!

3. Lernen Sie, kleine Aufgabenpakete fertig zu bekommen.
   In crossfunktionalen Teams.

Leben Sie inkrementelle Iterationen

4. Nehmen Sie Fachbereiche, UX, … Geschäftsführung mit.

Streben Sie Business-Agilität an

Der Weg zur Agilität?

Kommunikationskultur

Fehlerkultur

Feedback möglich

Programmier-Handwerkszeug

Continuous *

Agilität?

Angrenzende Abteilungen und Fachbereiche
nach und nach mitnehmen

Agilität ist der Weg!

Lernen & verbessern

Kommunikationskultur

Fehlerkultur

Feedback möglich

Programmier-Handwerkszeug

Continuous *

Agiles Arbeiten in einem Team, in mehreren Teams,
crossfunktional, selbstorganisiert, geführt …

Angrenzende Abteilungen und Fachbereiche
nach und nach mitnehmen

Fazit

Agilität fängt bei der Entwicklung an, *hört dort aber nicht auf.*

Nachhaltige Agilität umfasst das gesamte Unternehmen!

Es ist schwer, agil zu werden.
Und genauso schwer, agil zu bleiben.

Aber der Weg lohnt sich.

Fragen?

Agilität Scrum **Kanban** **XP**
DevOps **Continuous Integration**
Continuous Delivery

Feedback & Fragen

Continuous Deployment
Schneller? **Besser?** Feedback
Fehlerkultur **Lernen**
Kommunikation

Veranstaltungshinweis

Es geht agil bzw. *lean* weiter:

"Die 12 Prinzipien des Agilen Manifests

für Lean Communication?"

27. September 2016

GFU Semicolon

Ein bisschen Literatur

- *"Agile Projekte mit Scrum, XP und Kanban"*
  H. Wolf (Hrsg.), 2. Aufl., dpunkt.verlag 2015
- *"Selbstorganisierte Teams führen"*
  S. Kaltenecker, dpunkt.verlag 2016
- *"Retrospektiven in agilen Projekten"*
  J. Andresen, Hanser Verlag 2014
- *"96 Visualization Examples"*
  J. Janlén, Crisp Publishing 2015
- *"The Phoenix Project"*
  G. Kim et al., IT Revolution Press 2013

Danke

Vielen Dank! 😀

thomas@muchsoft.com

www.javabarista.de

@thmuch

#SemicolonGFU