Komplex und Kompliziert

Überleben in der Dauerkrise Softwareentwicklung

Dr. Jendrik Johannes

Softwareentwicklung In der Industriellen Praxis, TU Dresden, WS 16

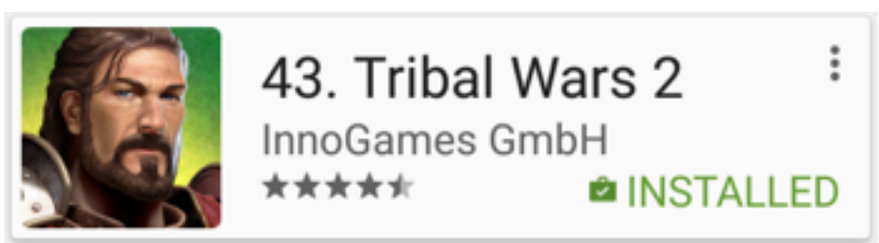
2001 - 2006
Studium Medieniformatik
(Schwerpunkt Software)

2007 - 2010
Promotion
(und Tool-Entwicklung)

2011
Projektarbeit
(Beratung / Entwicklung)

2012
alles ;)

2013 - 2015
Produktentwicklung
(vieles)



2016
Tool-/Open-Source-
Produktentwicklung

Bob Marshall
@flowchainsensei

Follow

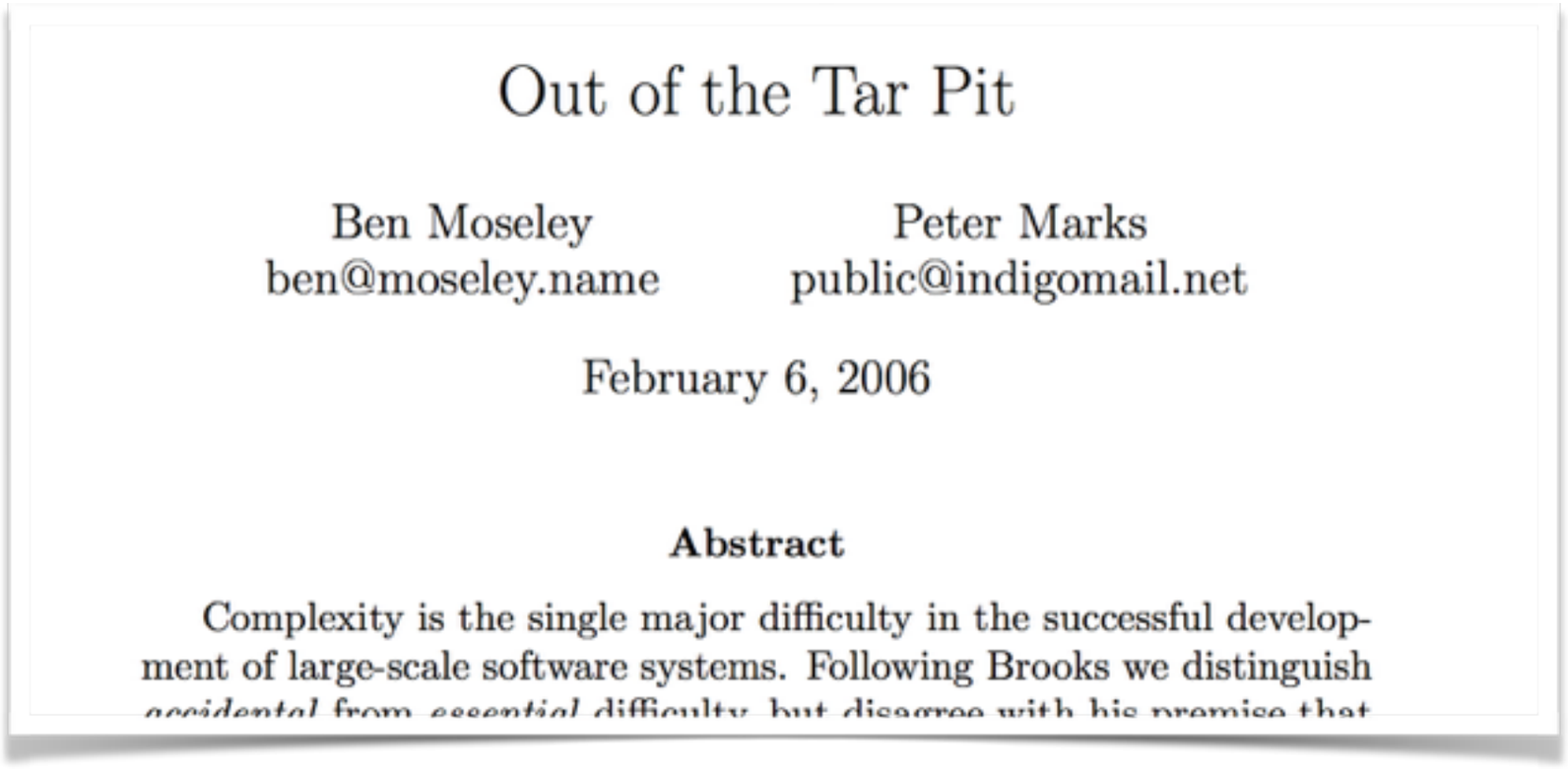
There’s so much we don’t know about software development that we might as well say we know nothing at all.

9:53 PM - 6 Oct 2016

20 12


WE KNOW NOTHING
ABOUT SOFTWARE DEVELOPMENT
memegenerator.net

Komplex? und **Kompliziert?**



Out of the Tar Pit

Ben Moseley
ben@moseley.name

Peter Marks
public@indigomail.net

February 6, 2006

Abstract

Complexity is the single major difficulty in the successful development of large-scale software systems. Following Brooks we distinguish *accidental* from *essential* difficulty, but disagree with his premise that

"Complexity is *the* root cause of the vast majority of problems with software today."

Mosely, Marks; - http://shaffner.us/cs/papers/tarpit.pdf
summary by Adrian Colyer's *the morning paper* - https://blog.acolyer.org/2015/03/20/out-of-the-tar-pit

Komplex? und **Kompliziert?**

Essential Complexity

is inherent in, and the essence of, the problem
(*as seen by the users*)

Accidental Complexity

is *all* the rest —
complexity with which the development team
would not have to deal in the *ideal world*

(e.g. complexity arising from performance issues
and from suboptimal language and infrastructure)

Moseley, Marks; - http://shaffner.us/cs/papers/tarpit.pdf
summary by Adrian Colyer's *the morning paper* - https://blog.acolyer.org/2015/03/20/out-of-the-tar-pit

Komplex? und Kompliziert?

Essential Complexity

= Komplex

Accidental Complexity

= Kompliziert

Moseley, Marks; - http://shaffner.us/cs/papers/tarpit.pdf
summary by Adrian Colyer's *the morning paper* - https://blog.acolyer.org/2015/03/20/out-of-the-tar-pit

Komplex? und **Kompliziert?**

1. [Accidental] Complexity caused by State
2. [Accidental] Complexity caused by Control

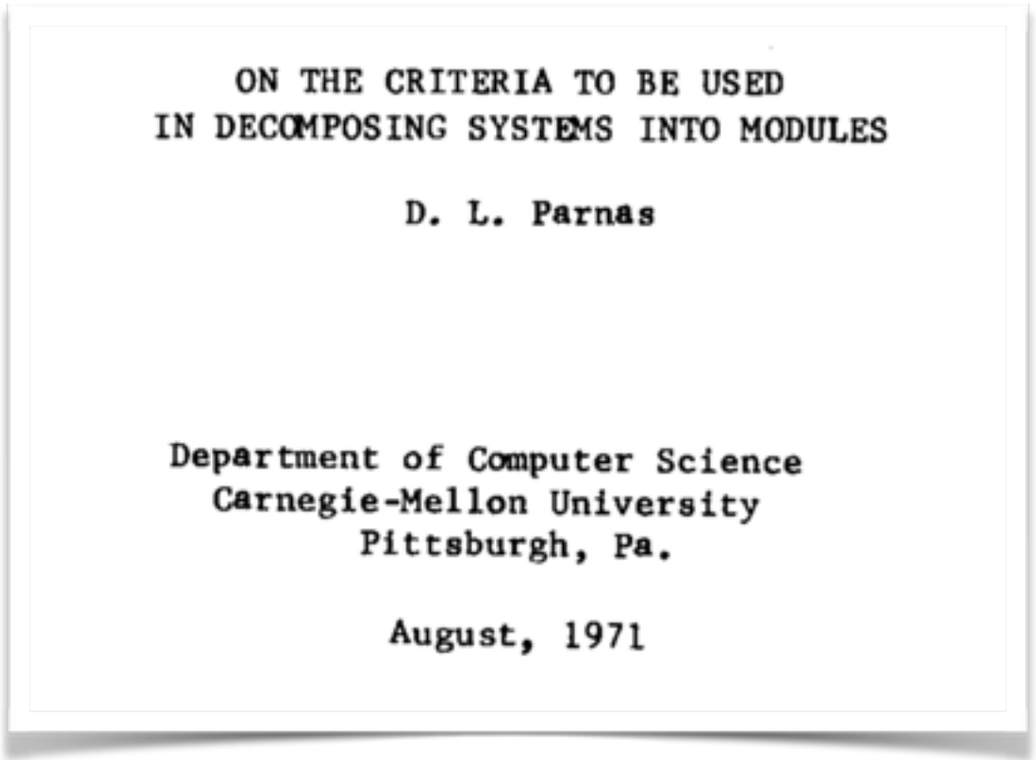
```
public class Bibliothek {

   private Buch [] meineBuecher = new Buch[10];
   private int anzahl = 0;

   public void aufnehmen(Buch buch) {
      meineBuecher[anzahl] = buch;
      anzahl += 1;
      System.out.println("Ich habe das Buch " + buch + " aufgenommen!");
   }
}
```

LV Softwaretechnologie TU Dresden - Übung 1 - http://st.inf.tu-dresden.de/files/teaching/ws07/SWTI/Bibliothek.java

Dekomposition des Systems



ON THE CRITERIA TO BE USED
IN DECOMPOSING SYSTEMS INTO MODULES

D. L. Parnas

Department of Computer Science
Carnegie-Mellon University
Pittsburgh, Pa.

August, 1971

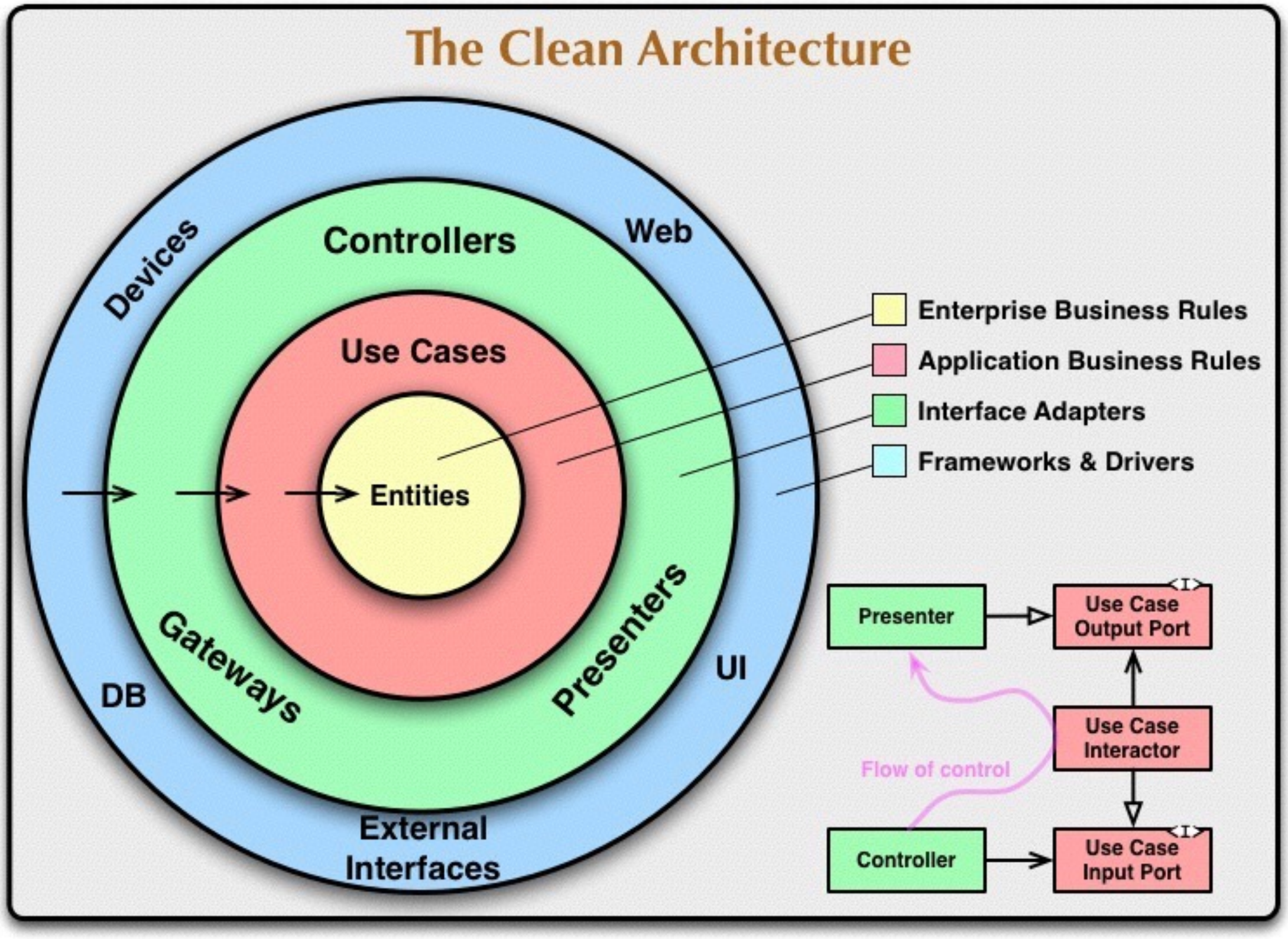
"The effectiveness of a *modularization* is dependent upon the criteria used in dividing the system into modules."

http://repository.cmu.edu/cgi/viewcontent.cgi?article=2979&context=compsci
summary by Adrian Colyer's *the morning paper:*
https://blog.acolyer.org/2016/09/05/on-the-criteria-to-be-used-in-decomposing-systems-into-modules

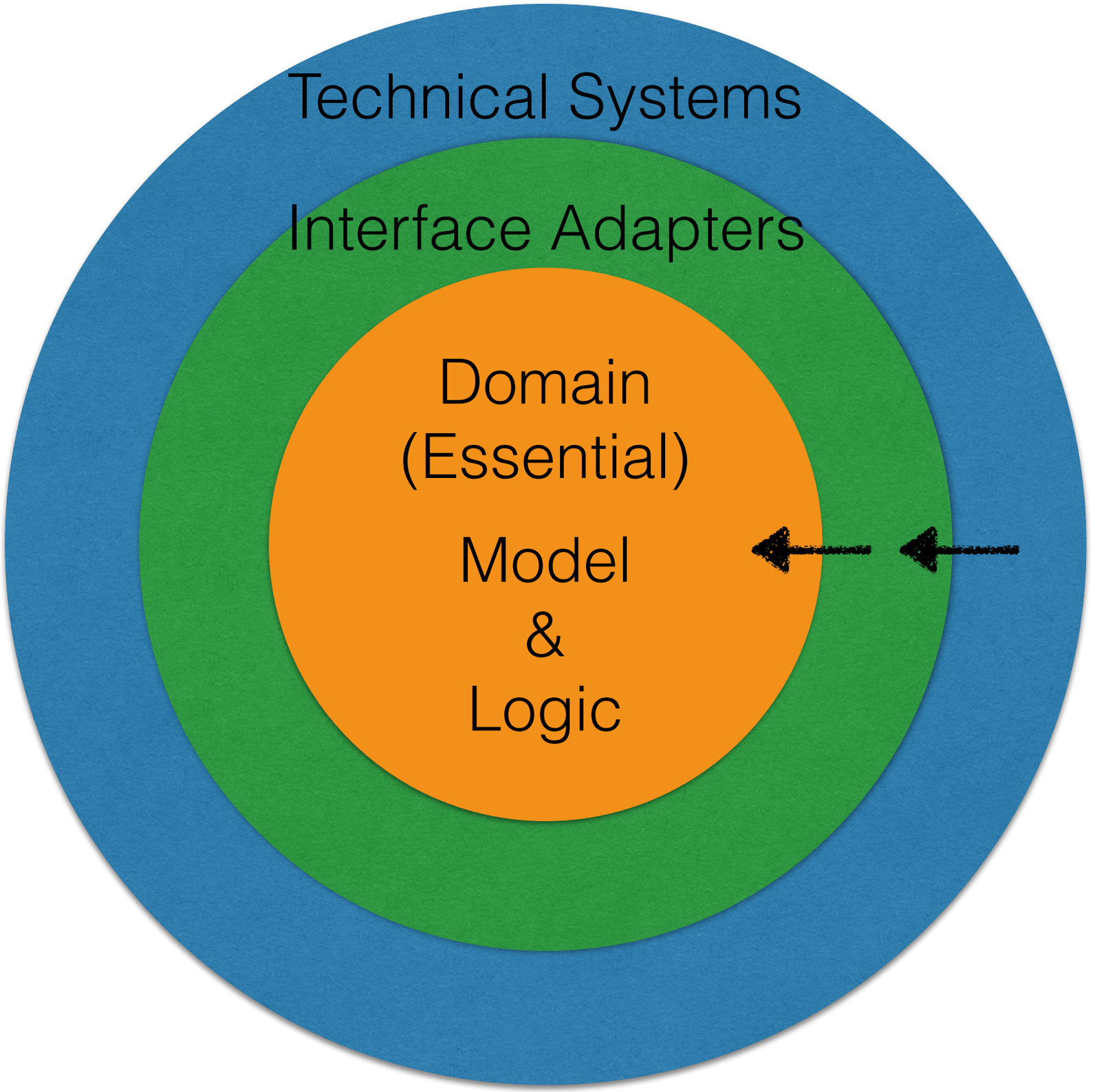
<u>Kriterium für Dekomposition:</u> Planbarkeit erzeugen durch Trennung von Planung und Umsetzung

technische Umsetzung

Planung



Uncle Bob; The Clean Architecture - https://8thlight.com/blog/uncle-bob/2012/08/13/the-clean-architecture.html



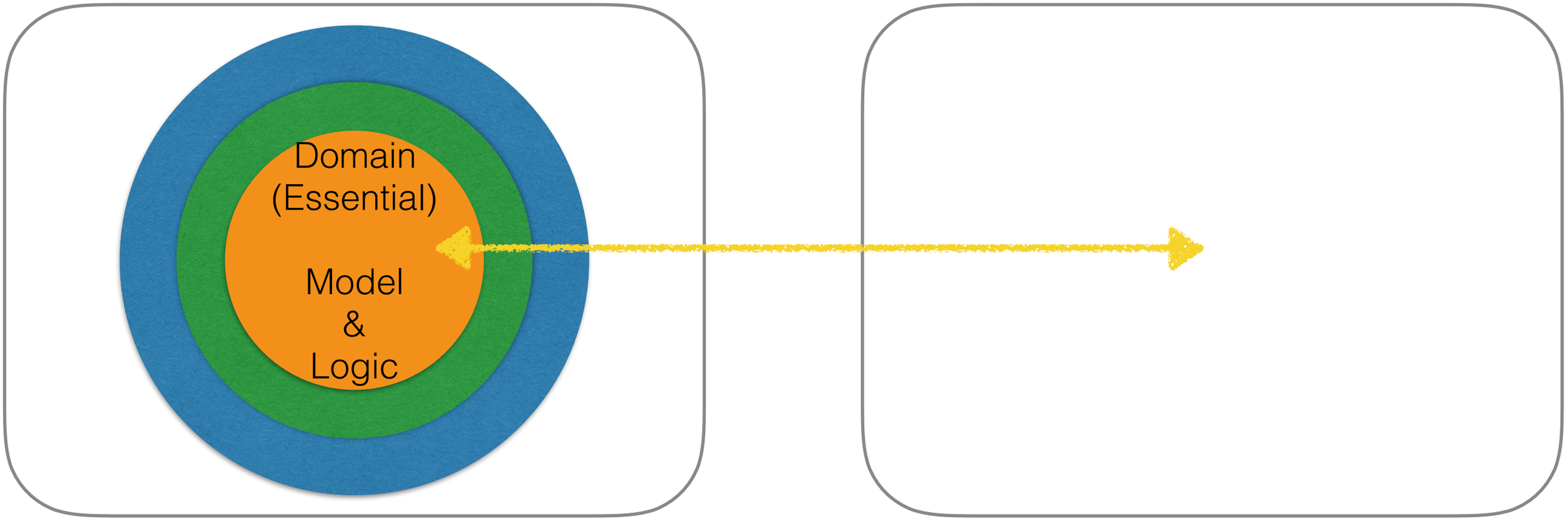
"[The Dependency] Rule says that source code dependencies can only point inwards."

Uncle Bob; The Clean Architecture - https://8thlight.com/blog/uncle-bob/2012/08/13/the-clean-architecture.html

<u>Kriterium für Dekomposition:</u> Planbarkeit erzeugen durch Trennung von Planung und Umsetzung

technische Umsetzung

Planung



https://www.youtube.com/watch?v=e5NANrVnqLc

The IT Crowd, S1E1 (2006)

Wahrnehmung von Technologie / “Technikern”

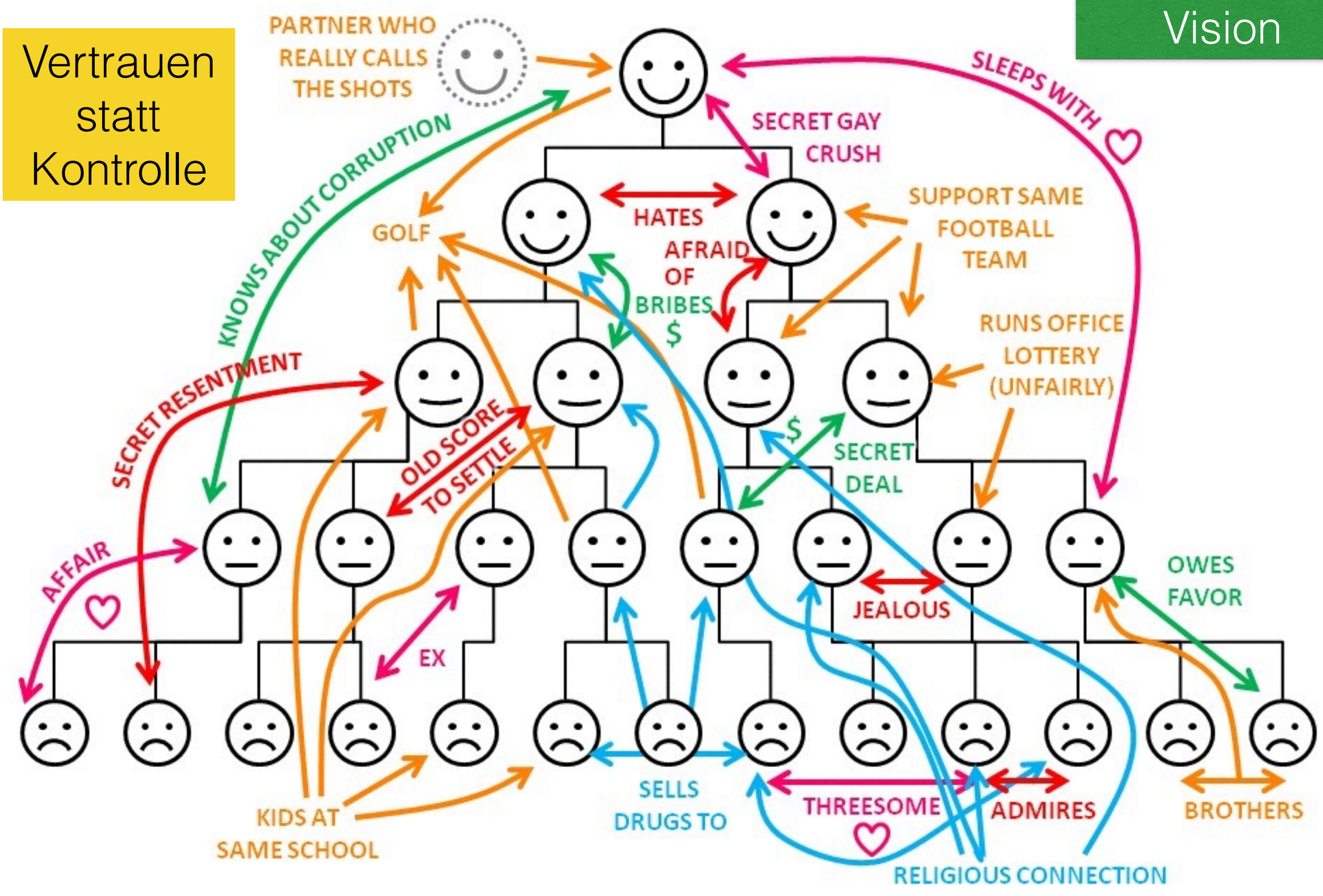
https://pbs.twimg.com/media/B5j9Z1mIcAAj-p0.jpg

Wahrnehmung von Technologie / “Technikern”

http://beyondthecubicle.org/wp-content/uploads/2014/12/cubiclesA.jpg

Vertrauen statt Kontrolle

Gemeinsame Vision



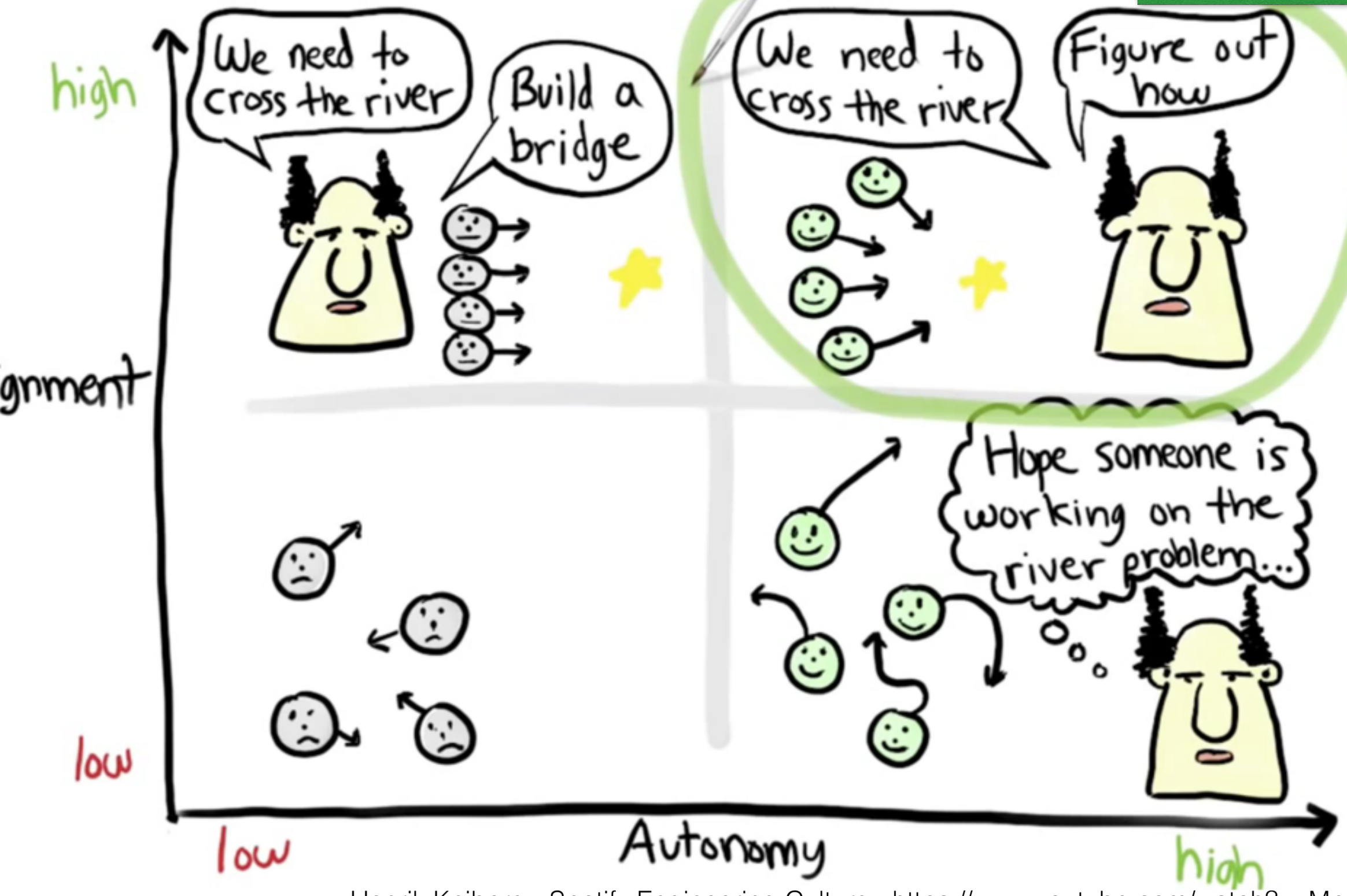
https://de.pinterest.com/pin/492581896490684O7

Vertrauen statt Kontrolle

http://dilbert.com/strip/1995-03-11

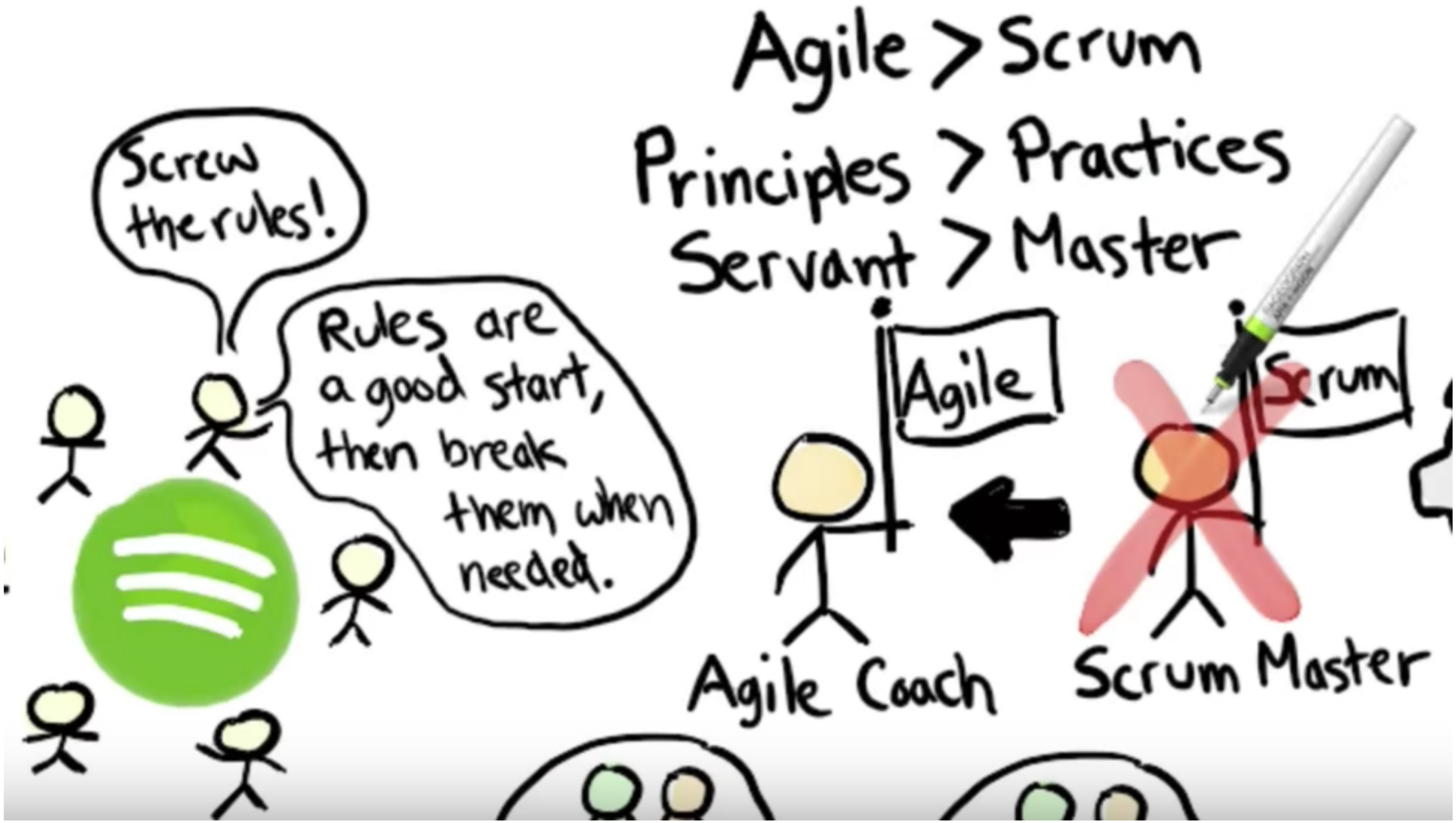
Vertrauen statt Kontrolle

Gemeinsame Vision



Henrik Kniberg - Spotify Engineering Culture - https://www.youtube.com/watch?v=Mpsn3Wal_4k

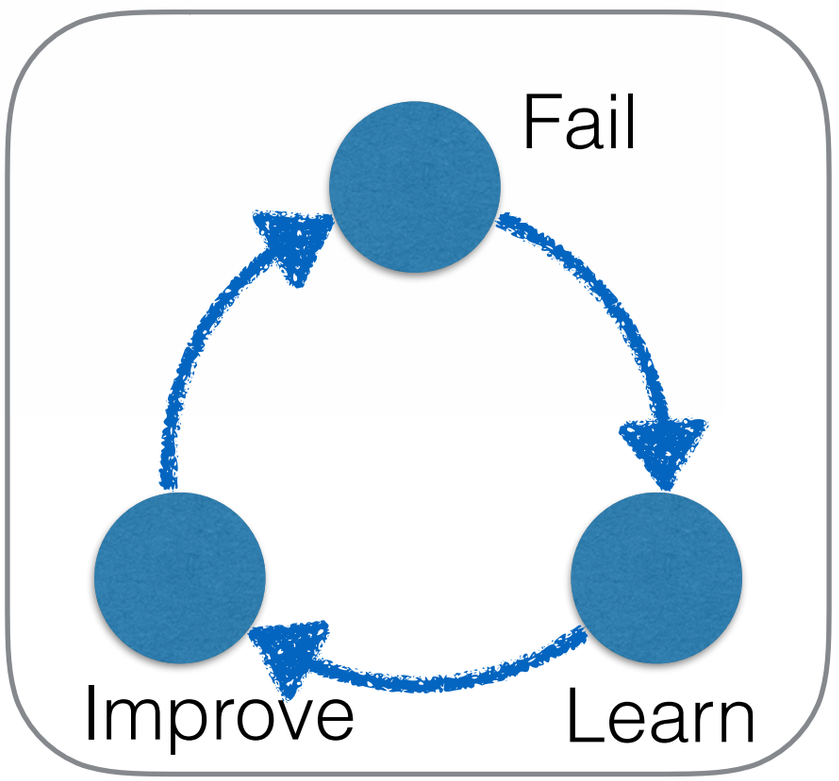
Fähigkeit der Systems sich selbst zu verändern, Strukturen und Prozessen anzupassen



Henrik Kniberg - Spotify Engineering Culture - https://www.youtube.com/watch?v=Mpsn3WaI_4k

Fail Fast ➡ Learn Fast ➡ Improve Fast

~~Planung~~

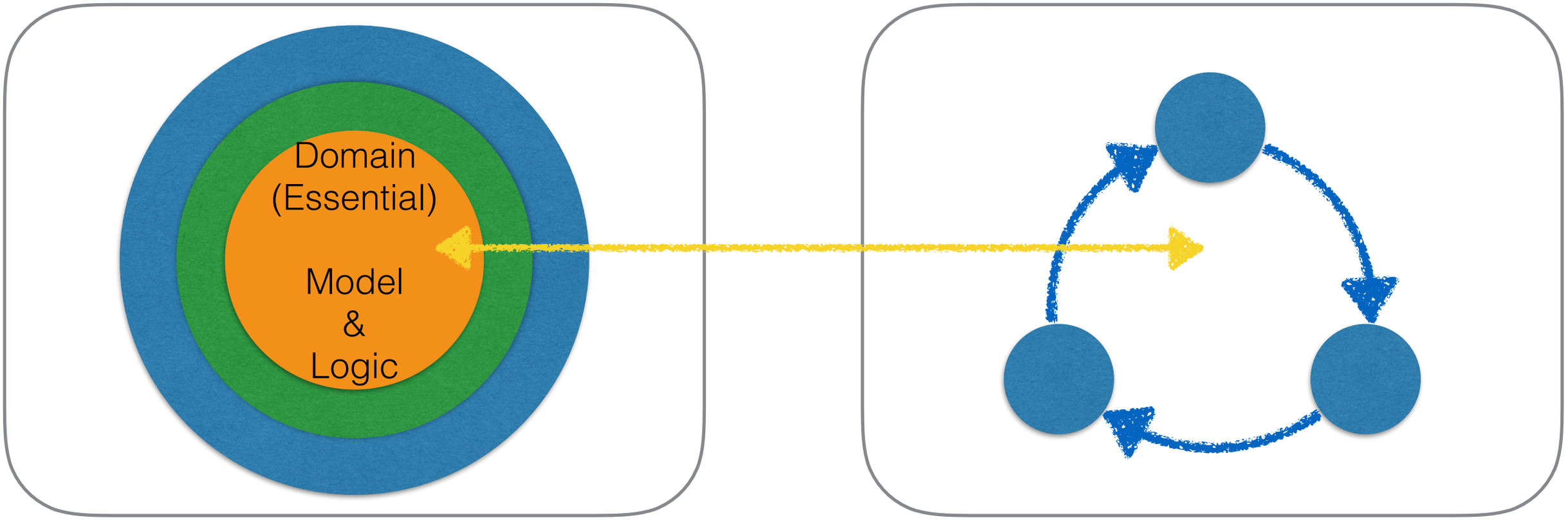


Henrik Kniberg - Spotify Engineering Culture - https://www.youtube.com/watch?v=Mpsn3Wal_4k

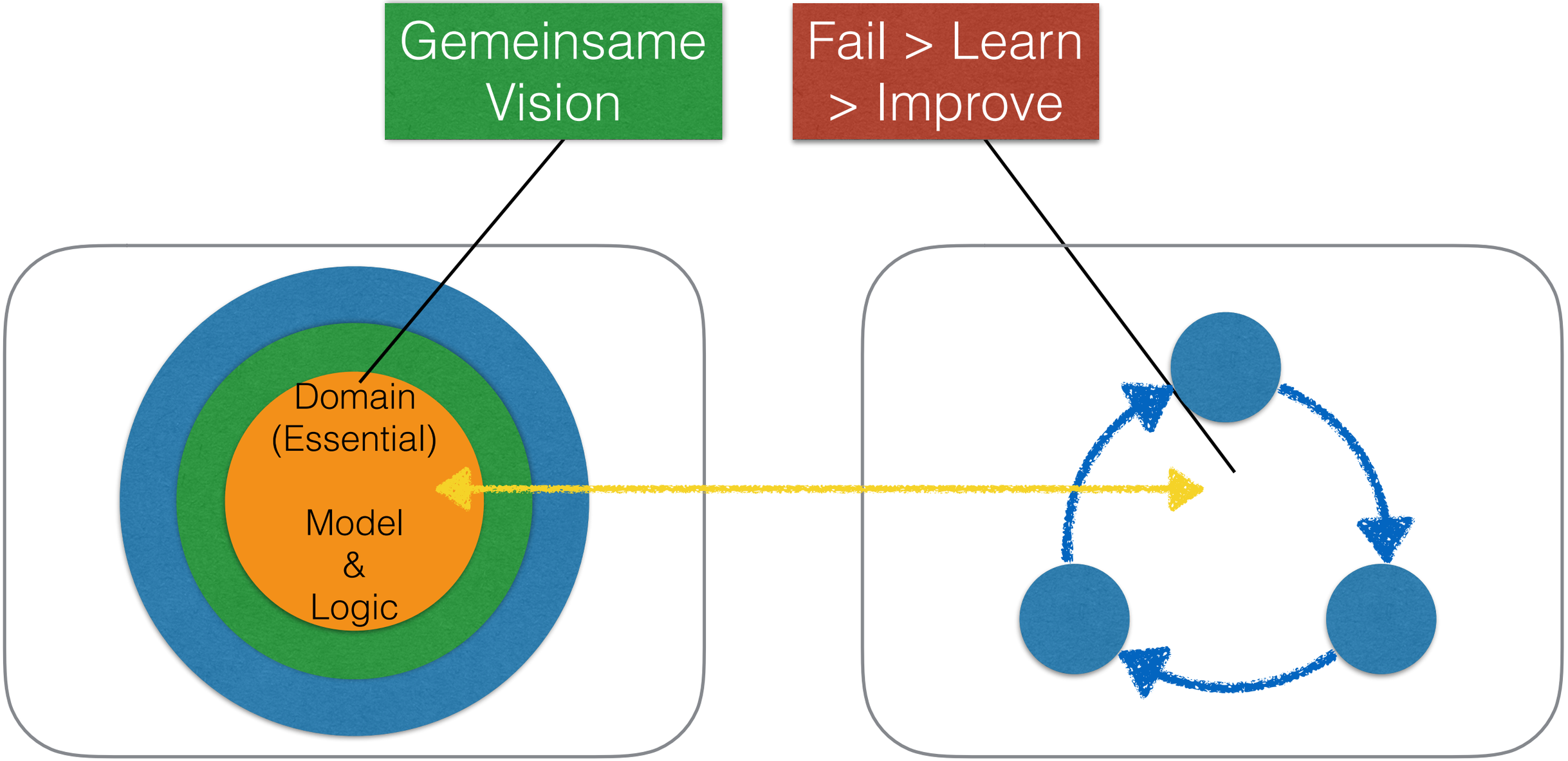
Kriterium für Dekomposition: ~~Planbarkeit erzeugen durch Trennung von Planung und Umsetzung~~

~~technische Umsetzung~~

~~Planung~~

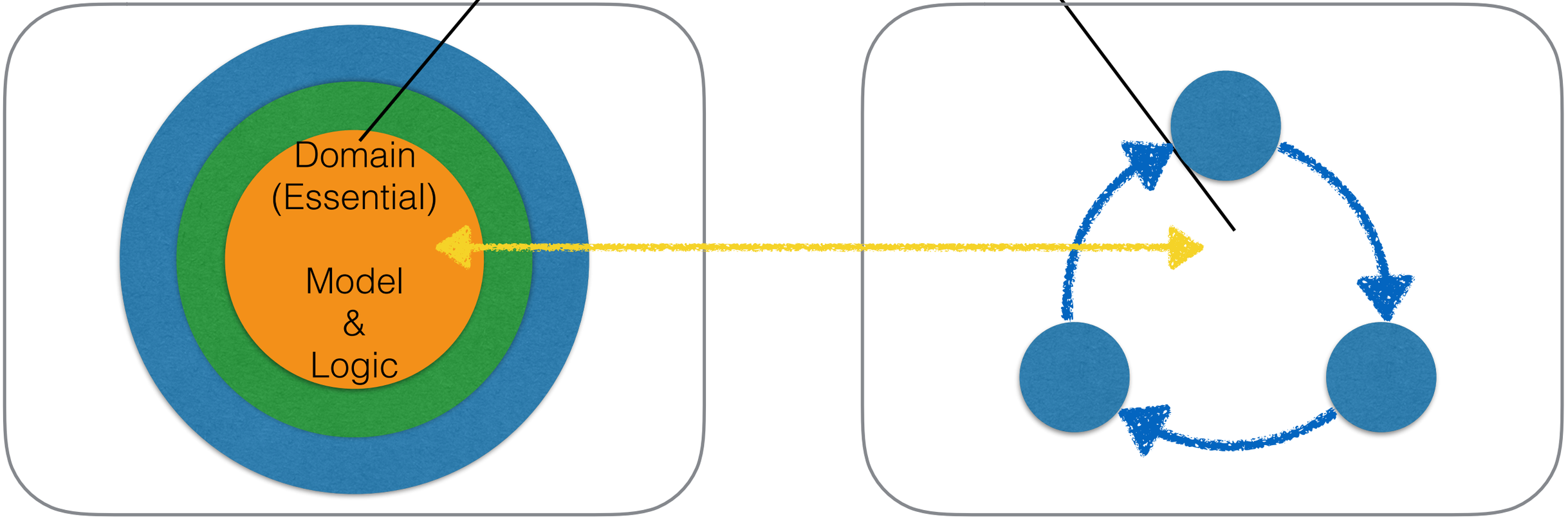


Kriterium für Dekomposition: ~~Planbarkeit erzeugen durch Trennung von Planung und Umsetzung~~



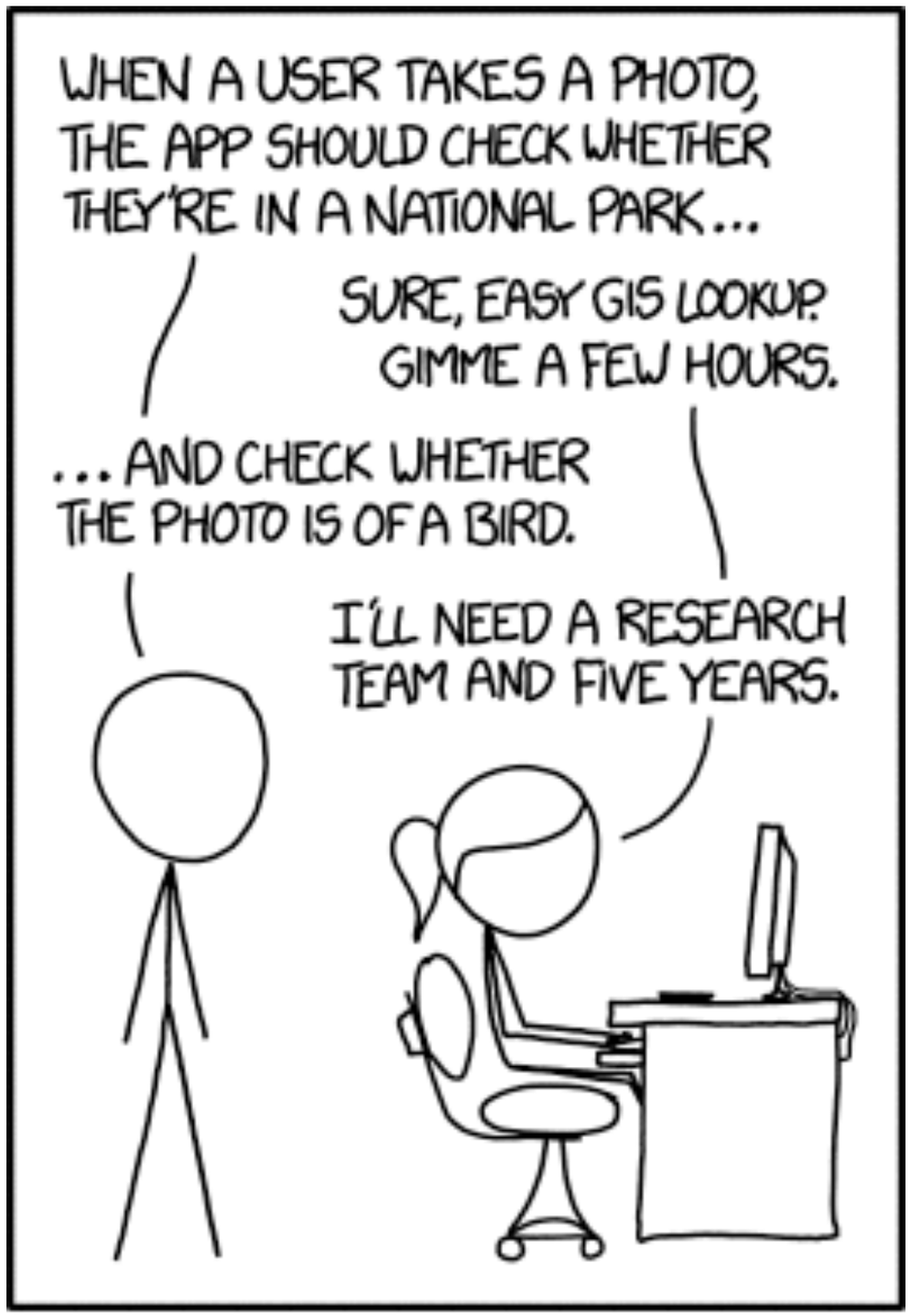
Gemeinsame Vision

Fail > Learn > Improve



Wahrnehmung von Technologie/Technikern

Vertrauen statt Kontrolle



Tasks - http://xkcd.com/1425

Tree.js

1:0 drawScene () Written by me

```
1
2  //=========================================================
3  //
4  //  scene
5  //
6
7  var ctx, canvasWidth, canvasHeight;
8
9  function drawScene (canvas) {
10     ctx = canvas.getContext("2d");
11     extendCanvasContext(ctx);
12
13     canvasWidth = parseInt(canvas.getAttribute("width"));
14     canvasHeight = parseInt(canvas.getAttribute("height"));
15
16     drawSky();
17     drawMountains();
18     drawTree();
19 }
20
21
22 //---------------------------------------------------------
23 //
24 //  sky
25 //
26
27 function drawSky () {
28     ctx.save();
29
30     var gradient = ctx.createLinearGradient(0,0,0,canvasHeight);
31     gradient.addColorStop(0, "#b4e0fe");
32     gradient.addColorStop(1, "#d3f8ff");
33
34     ctx.fillStyle = gradient;
35     ctx.fillRect(0,0,canvasWidth,canvasHeight);
36
37     ctx.restore();
38 }
```

Javascript Spaces (4) LF Western (Mac OS Roman) 97w

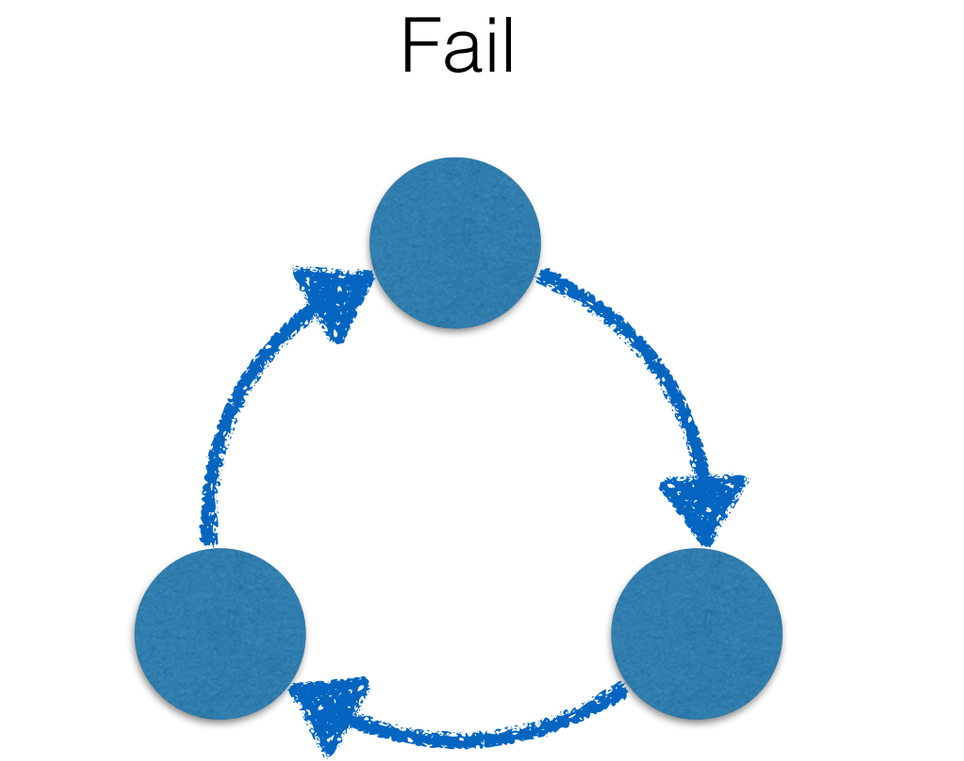
Bret Victor - Inventing on Principle (2012) - http://worrydream.com/#!/InventingOnPrinciple

Fokus auf kreative Arbeit

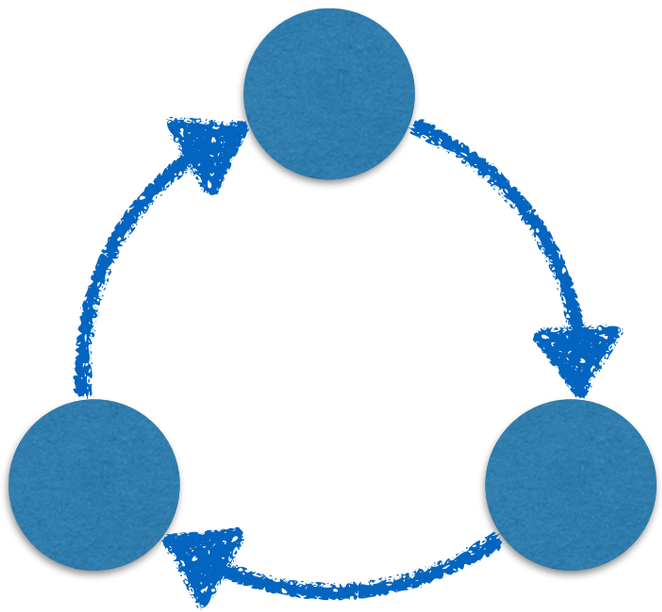
“creators need an immediate connection to what they create”

— Brett Victor, Inventing on Principle

Bret Victor - Inventing on Principle (2012) - http://worrydream.com/#!/InventingOnPrinciple


Fail
Improve
Learn

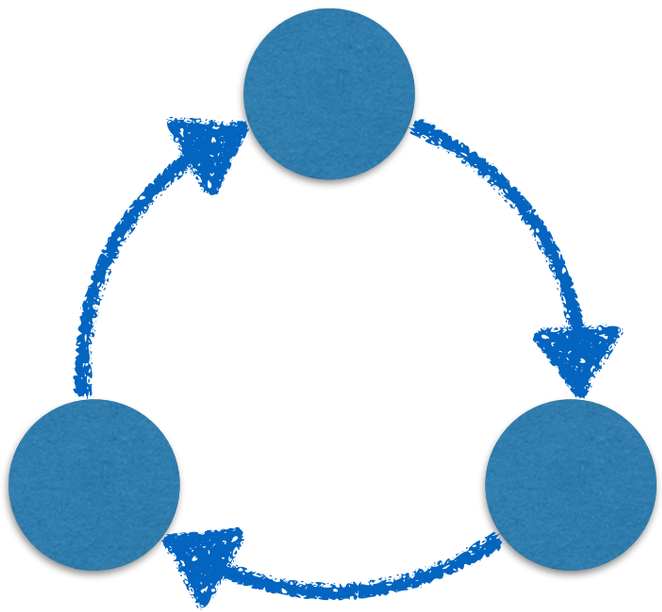
Compile / Run
= Fail



Write Code
= Improve

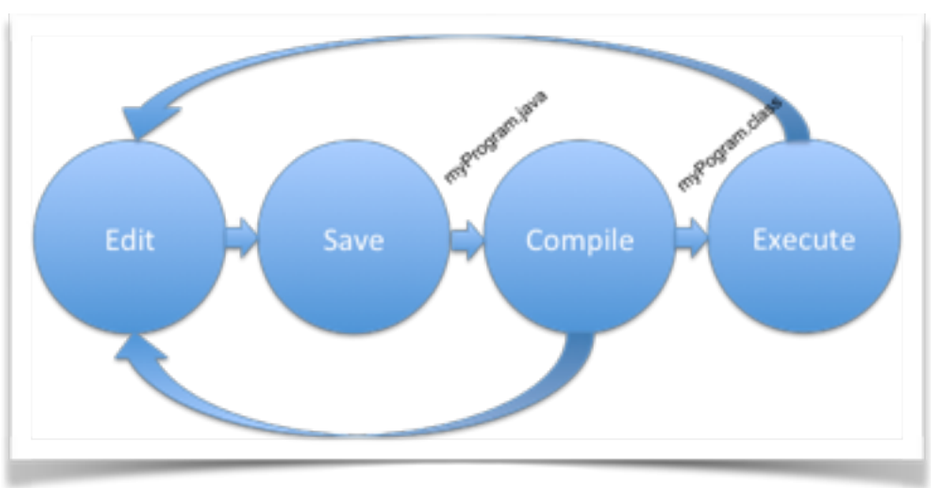
Observer
= Learn

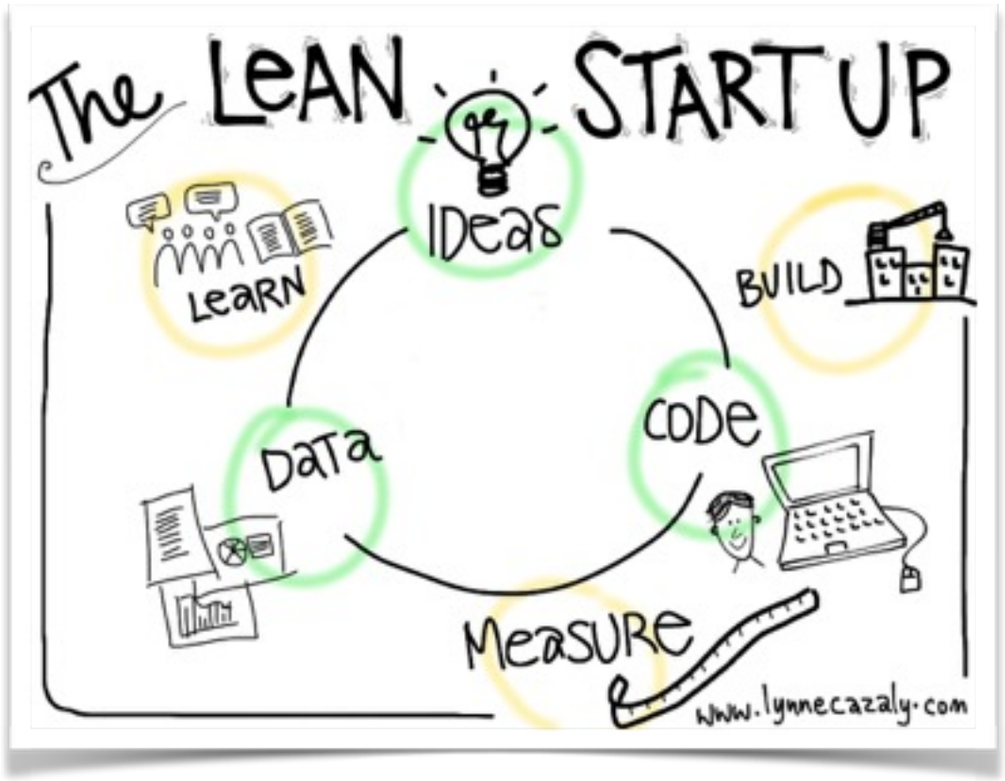
Fail

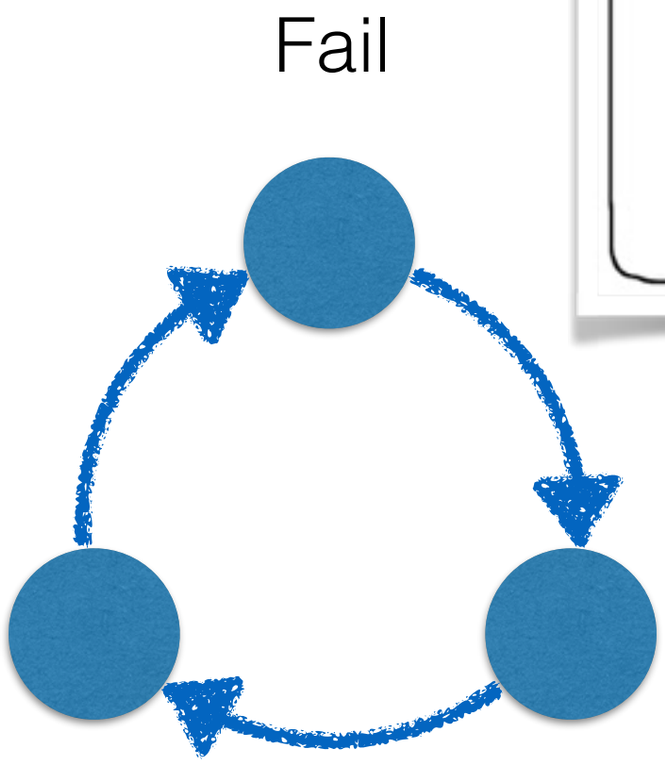


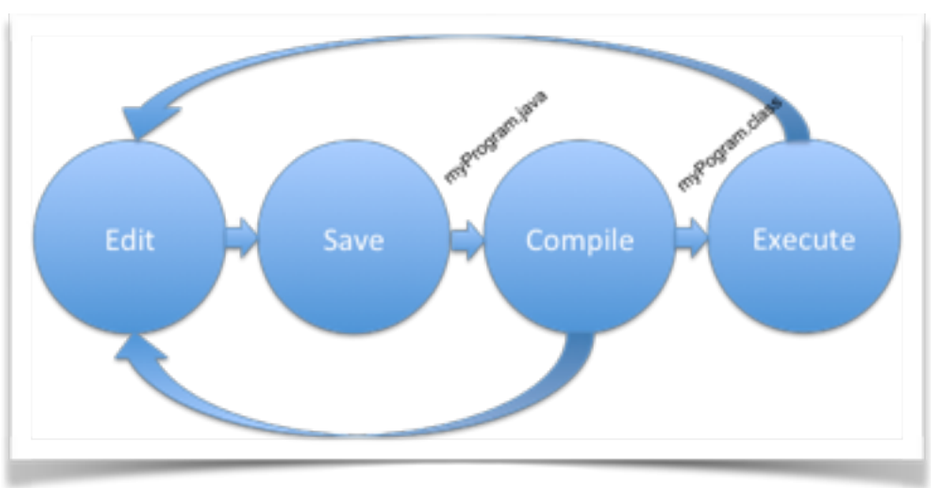
Improve

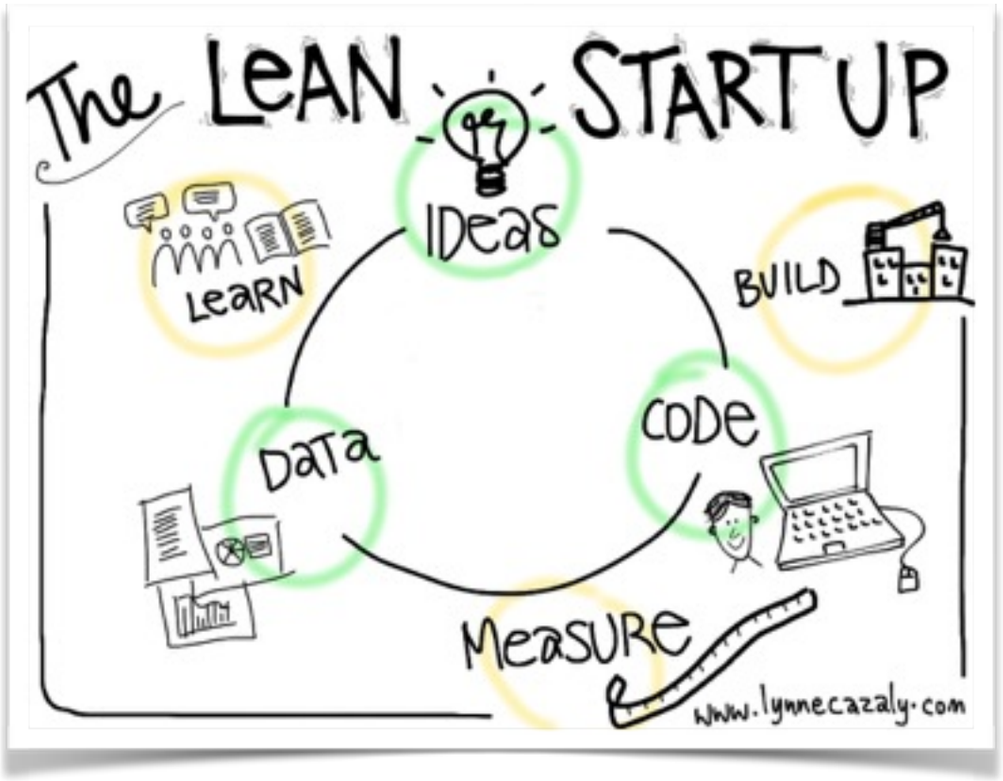
Learn

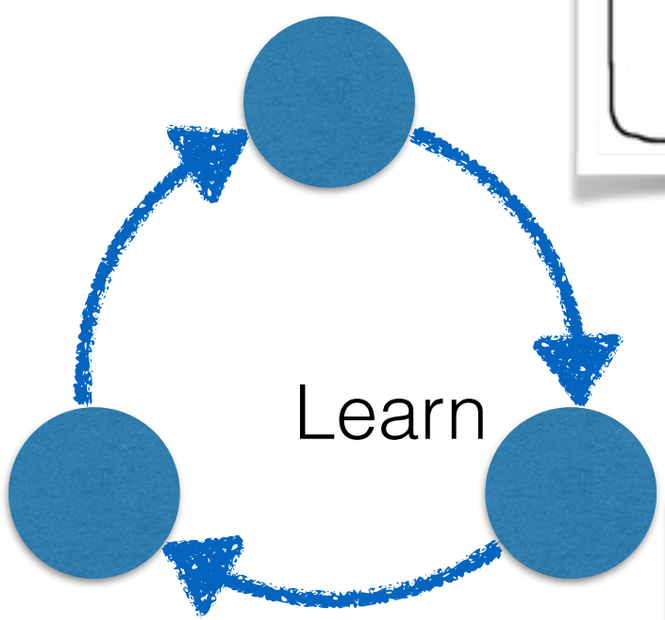


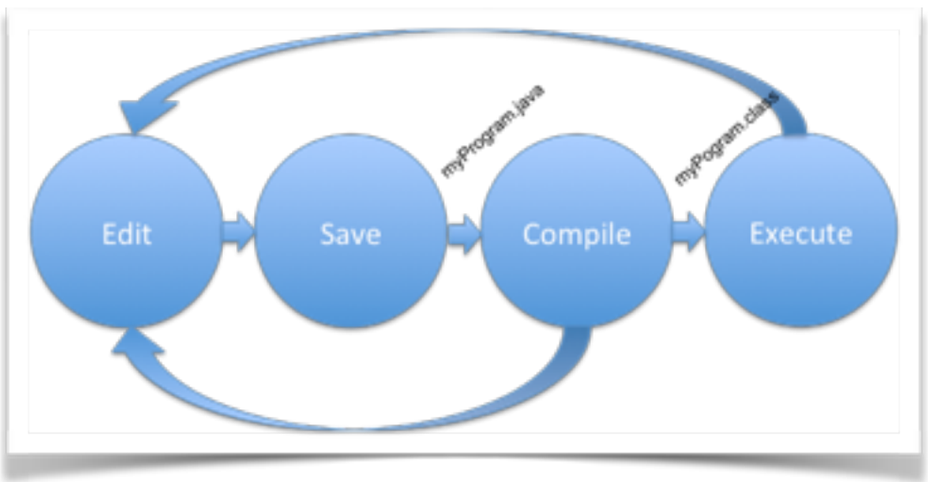

The LEAN START UP
IDEAS
LEARN
BUILD
CODE
DATA
MEASURE
www.lynnecazaly.com

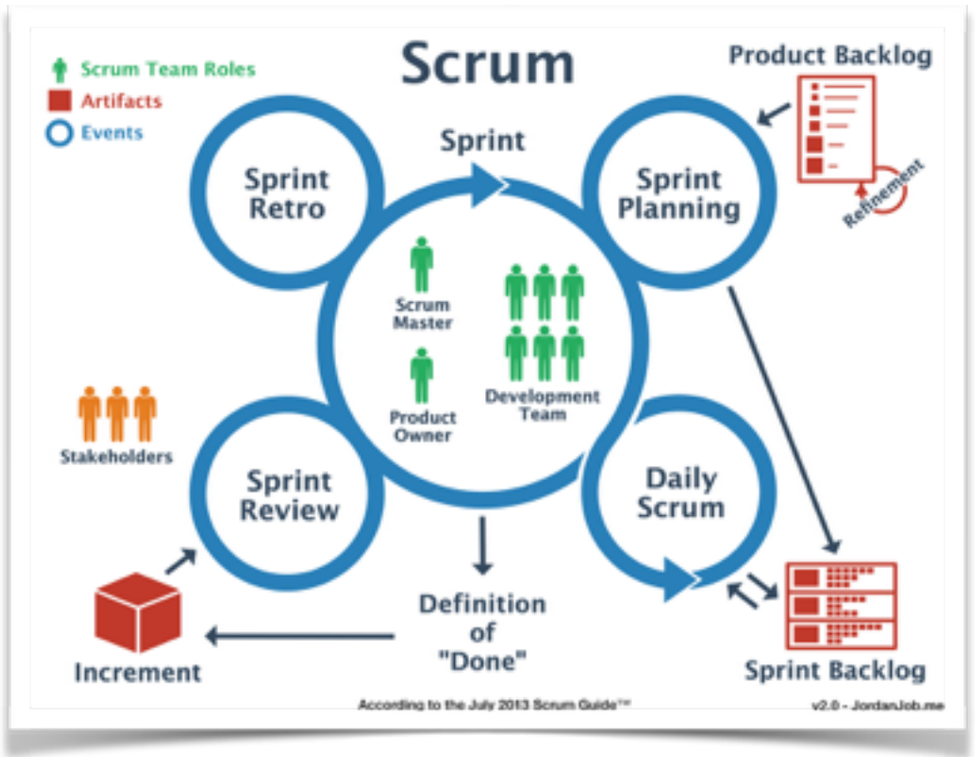

Fail
Improve
Learn

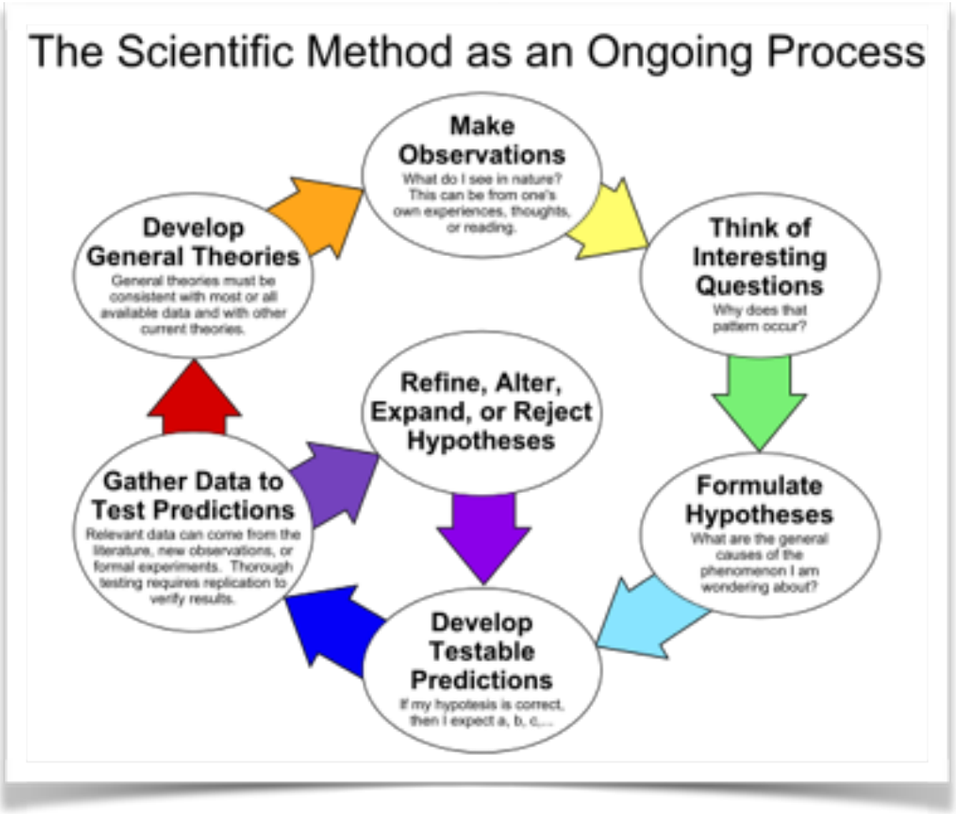

Edit
Save
myProgram.java
Compile
myProgram.class
Execute

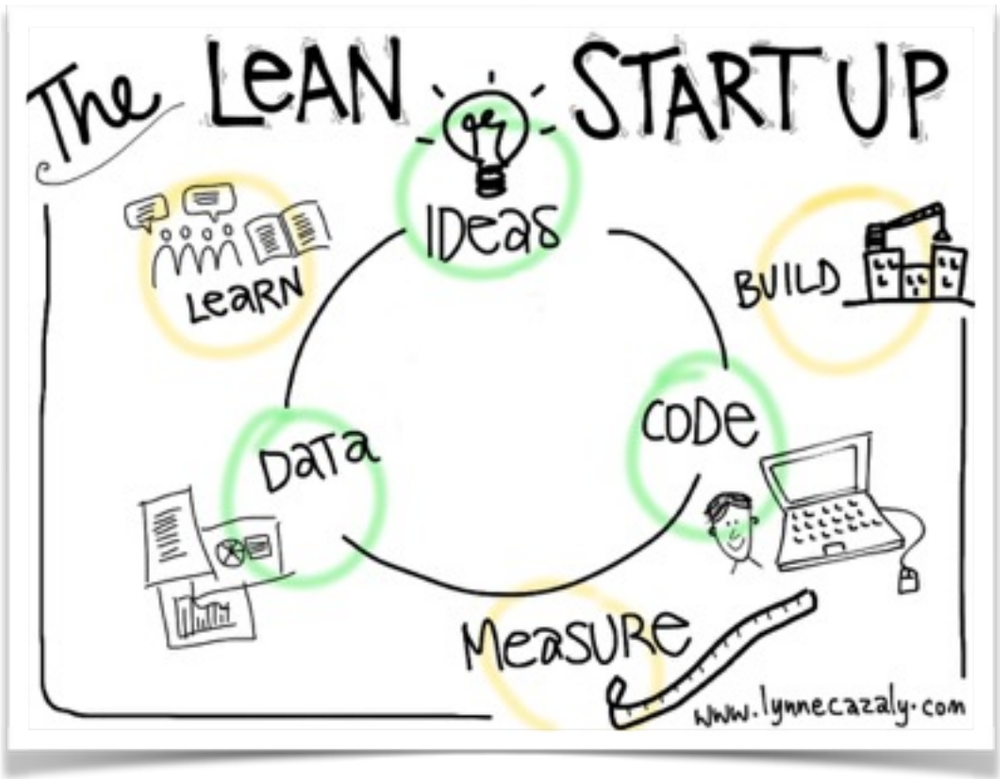

The LEAN START UP
IDEAS
BUILD
CODE
DATA
MEASURE
LEARN
www.lynnecazaly.com

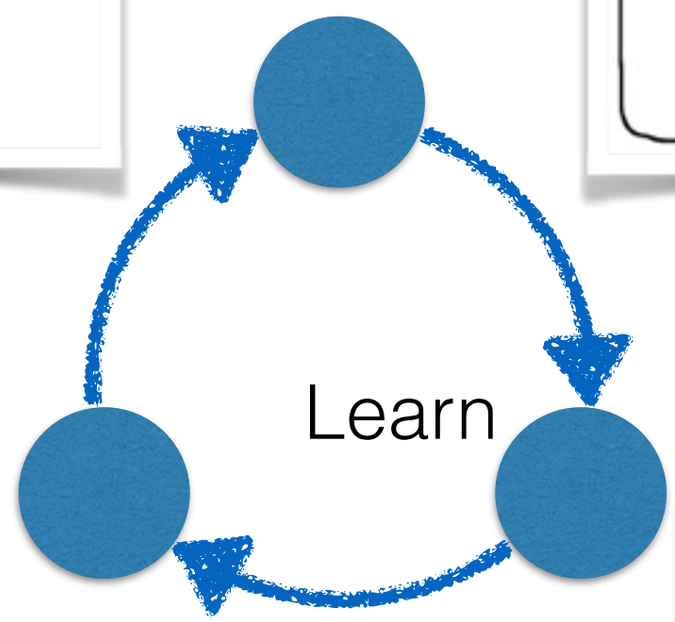

Fail
Learn
Improve

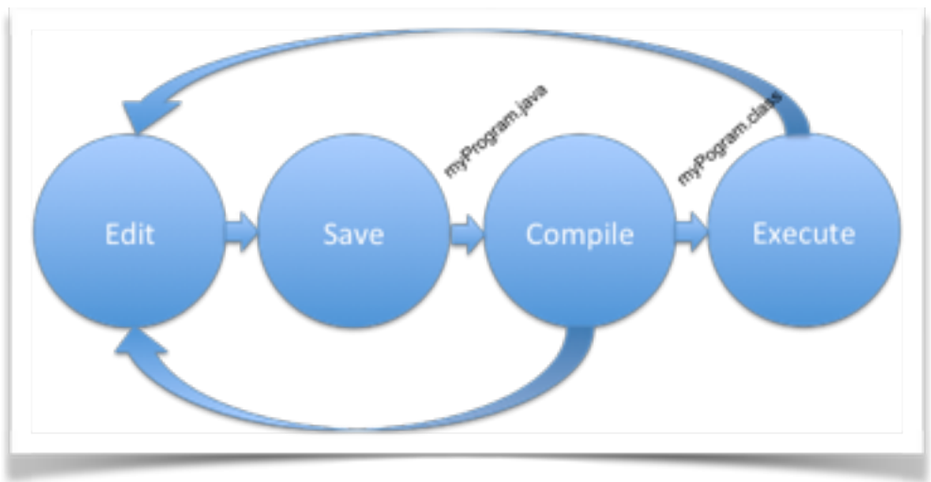

Edit
Save
Compile
Execute
myProgram.java
myProgram.class

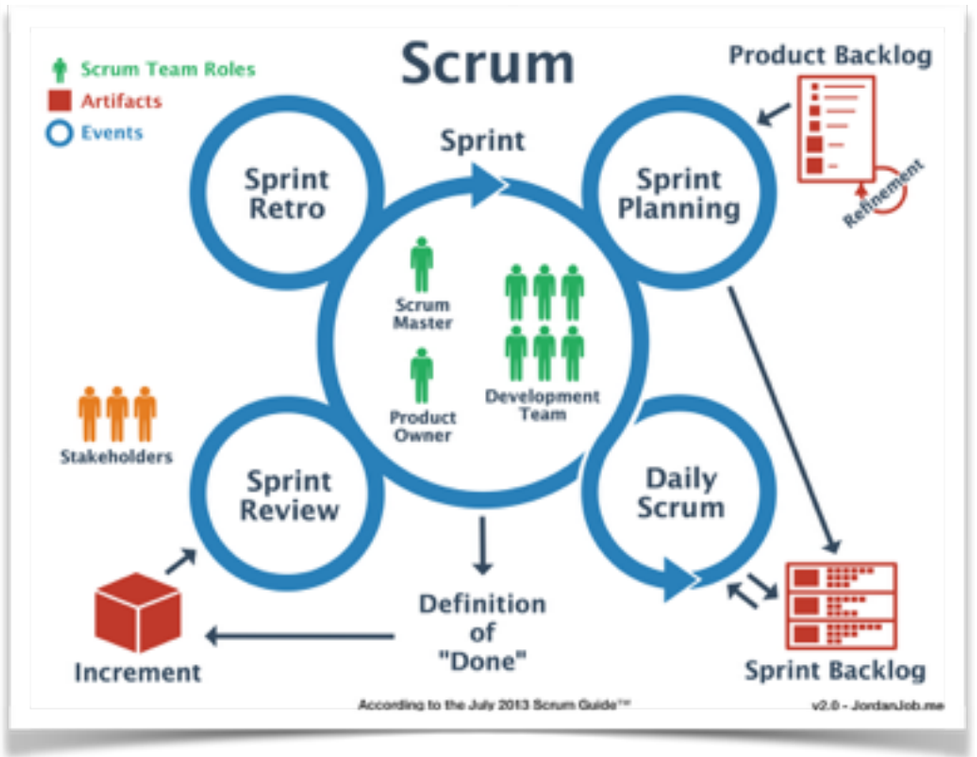

Scrum
Scrum Team Roles
Artifacts
Events
Product Backlog
Refinement
Sprint
Sprint Retro
Sprint Planning
Scrum Master
Product Owner
Development Team
Stakeholders
Sprint Review
Daily Scrum
Definition of "Done"
Increment
Sprint Backlog
According to the July 2013 Scrum Guide™
v2.0 - JordanJob.me


The Scientific Method as an Ongoing Process
Make Observations
What do I see in nature? This can be from one's own experiences, thoughts, or reading.
Think of Interesting Questions
Why does that pattern occur?
Formulate Hypotheses
What are the general causes of the phenomenon I am wondering about?
Develop Testable Predictions
If my hypotesis is correct, then I expect a, b, c....
Gather Data to Test Predictions
Relevant data can come from the literature, new observations, or formal experiments. Thorough testing requires replication to verify results.
Develop General Theories
General theories must be consistent with most or all available data and with other current theories.
Refine, Alter, Expand, or Reject Hypotheses


The LEAN START UP
IDEAS
BUILD
CODE
DATA
MEASURE
LEARN
www.lynnecazaly.com


Fail
Learn
Improve


Edit
Save
Compile
Execute
myProgram.java
myPogram.class


Scrum
Scrum Team Roles
Artifacts
Events
Product Backlog
Refinement
Sprint
Sprint Retro
Sprint Planning
Scrum Master
Development Team
Product Owner
Stakeholders
Sprint Review
Daily Scrum
Definition of "Done"
Increment
Sprint Backlog
According to the July 2013 Scrum Guide™
v2.0 - JordanJob.me

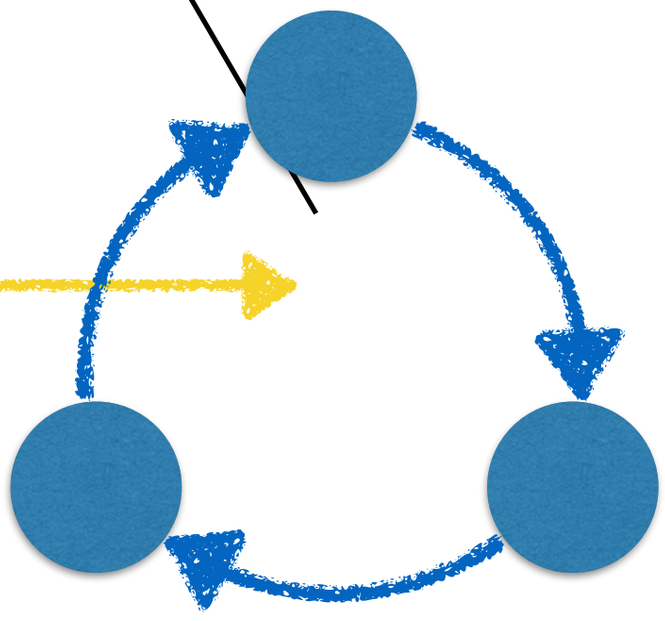
Gemeinsame Vision

Fail > Learn > Improve

Domain (Essential)

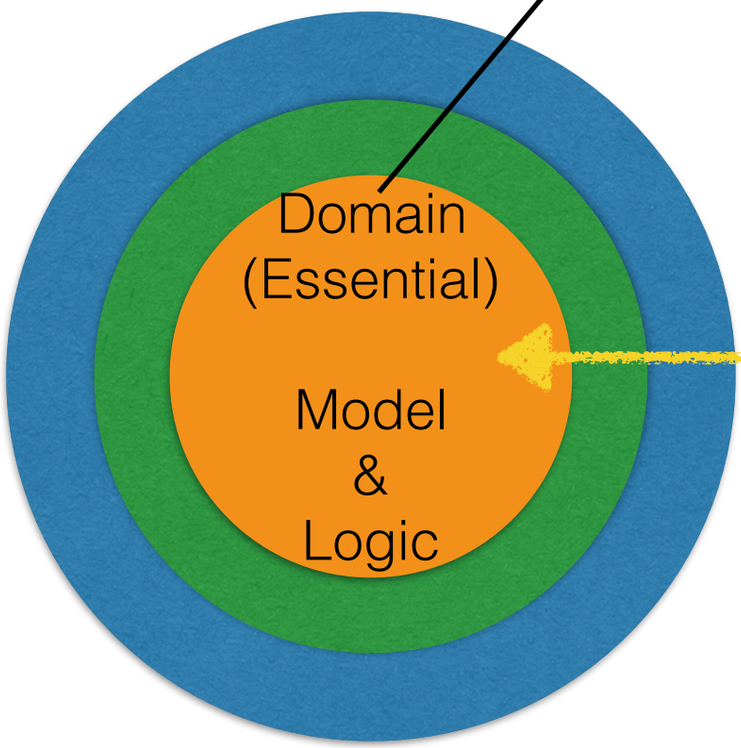
Model & Logic

Fokus auf kreative Arbeit

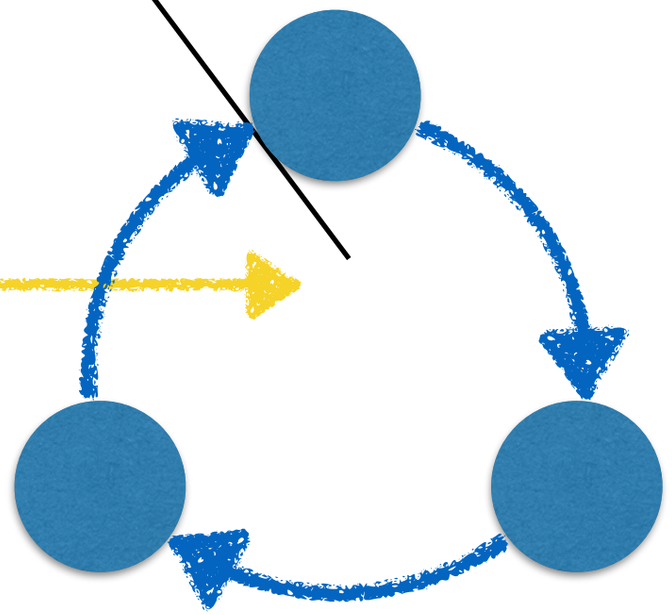


Gemeinsame Vision

Fail > Learn > Improve



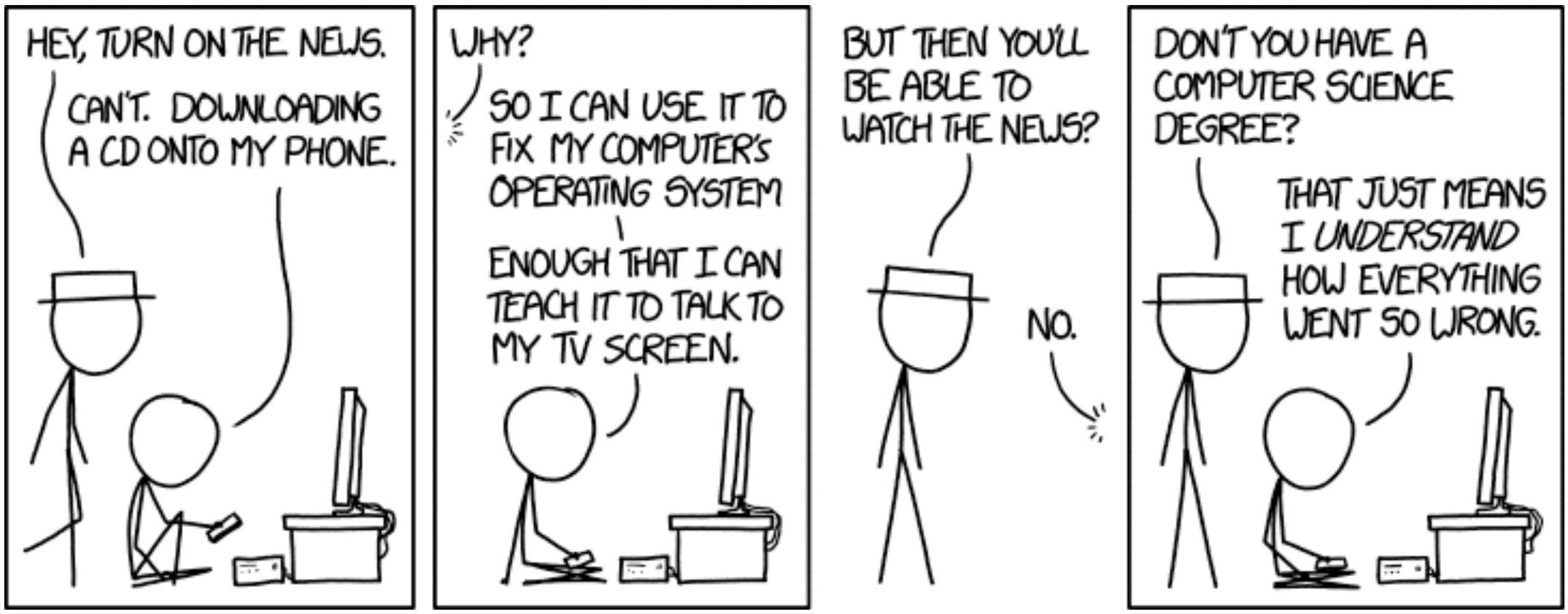
Fokus auf kreative Arbeit



Wahrnehmung von Technologie/Technikern

Vertrauen statt Kontrolle

Let’s talk about the state of **Tools**



TV Problems - https://www.xkcd.com/1760

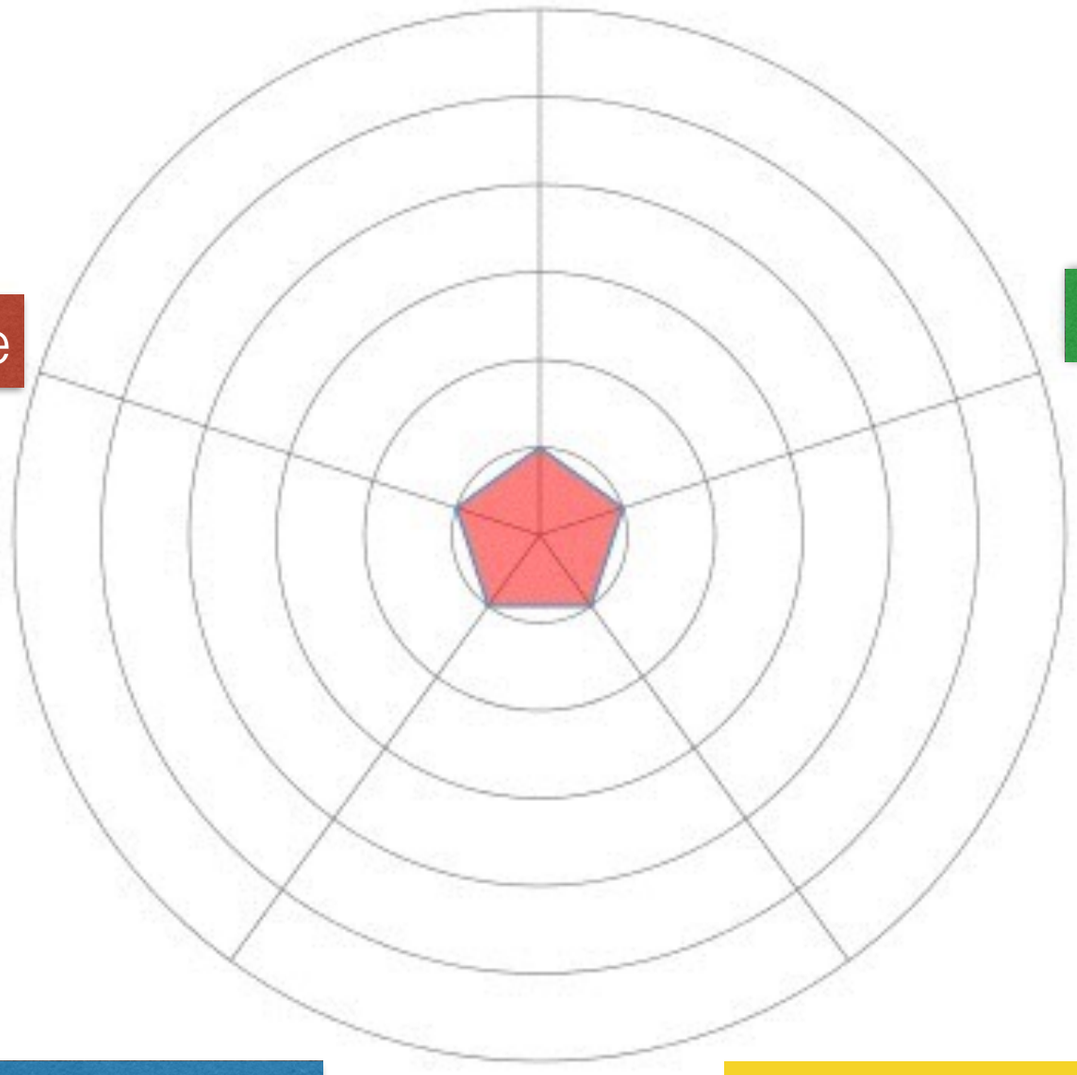
Tools help us reduce accidental complexity

…but often it goes (terribly) wrong

E-Mail

Wahrnehmung von Technologie/Technikern

Fail > Learn > Improve



Gemeinsame Vision

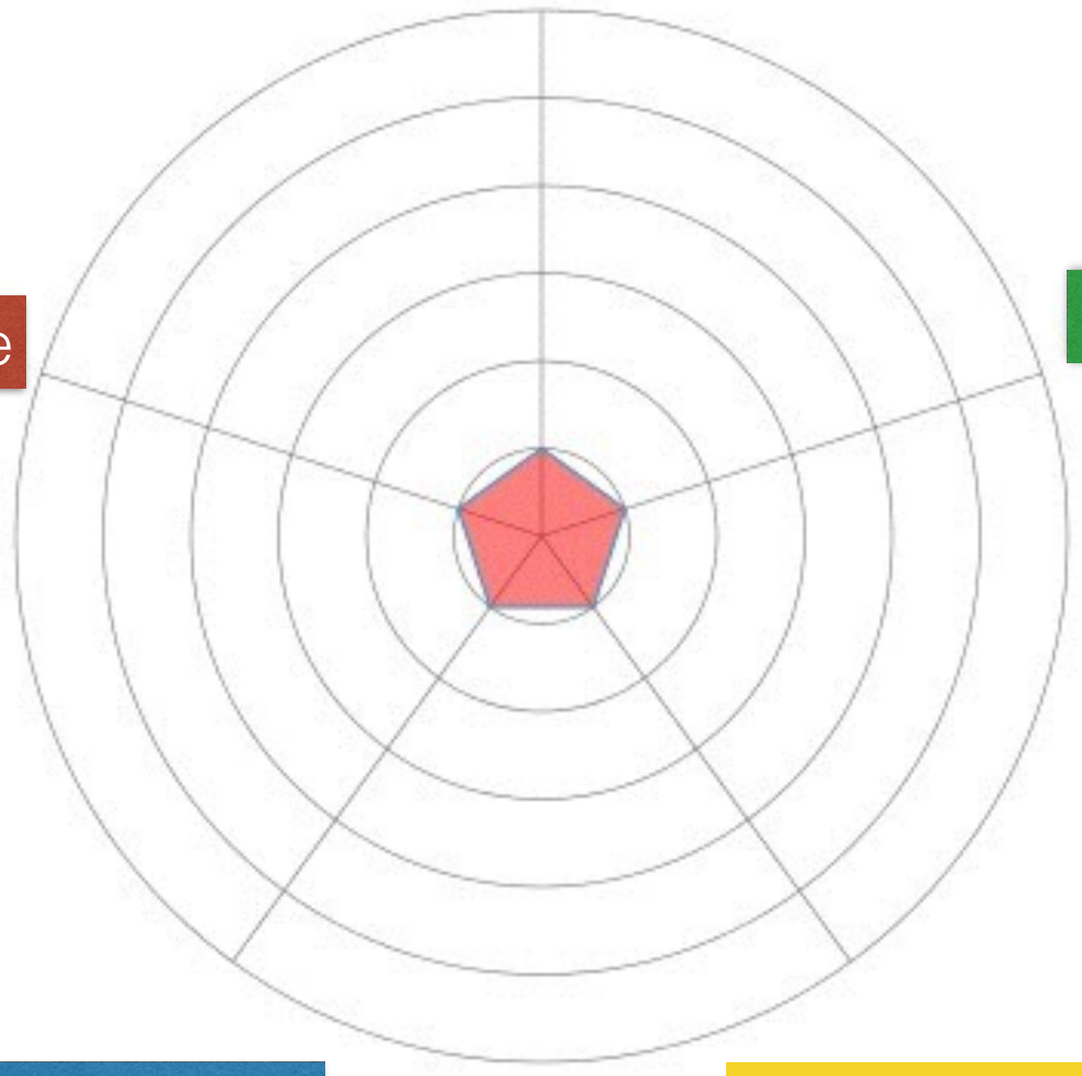
Fokus auf kreative Arbeit

Transparenz: Vertrauen statt Kontrolle

Meetings

Wahrnehmung von Technologie/Technikern

Fail > Learn > Improve



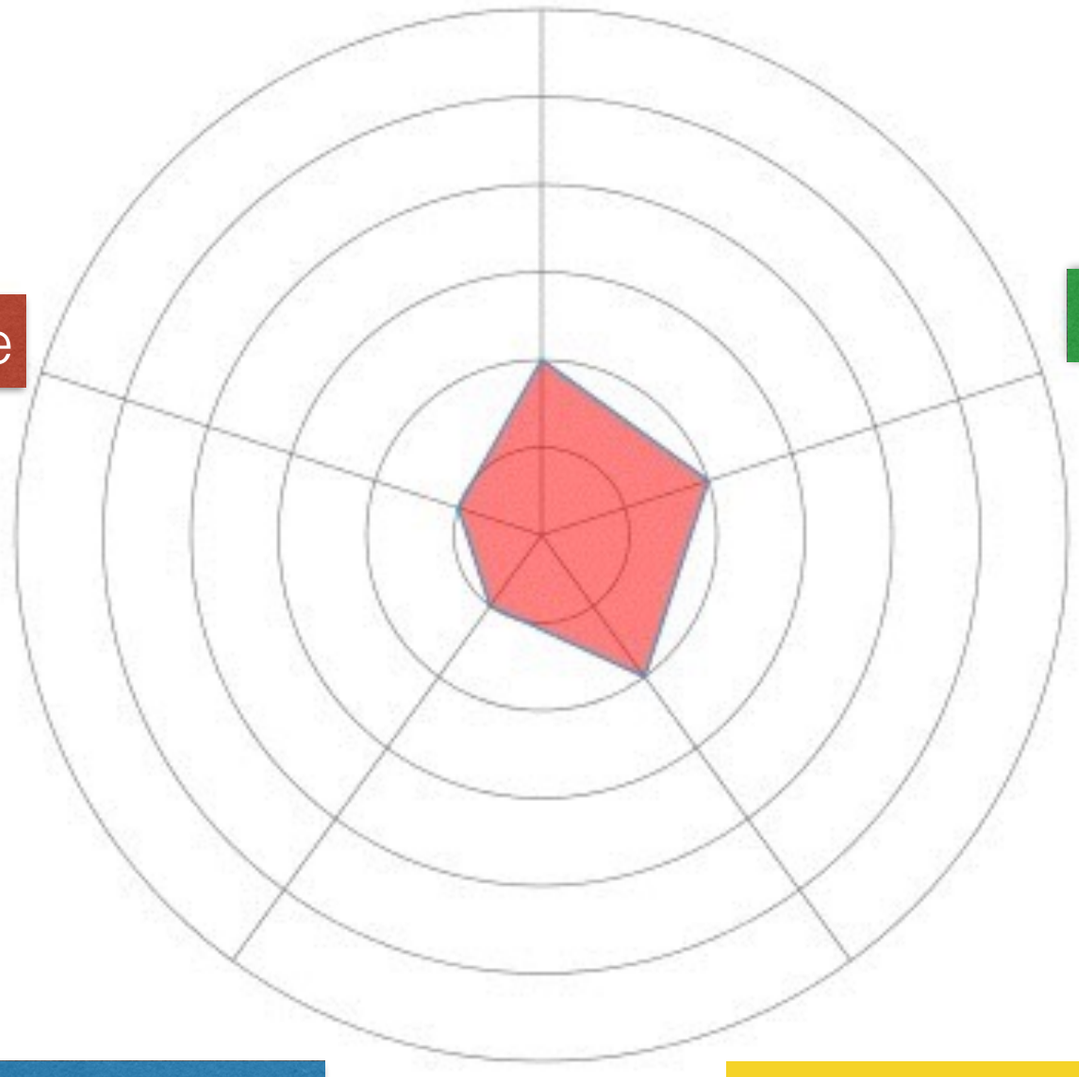
Gemeinsame Vision

Fokus auf kreative Arbeit

Transparenz: Vertrauen statt Kontrolle

JIRA (Ticket-System)

Wahrnehmung von Technologie/Technikern



Fail > Learn > Improve

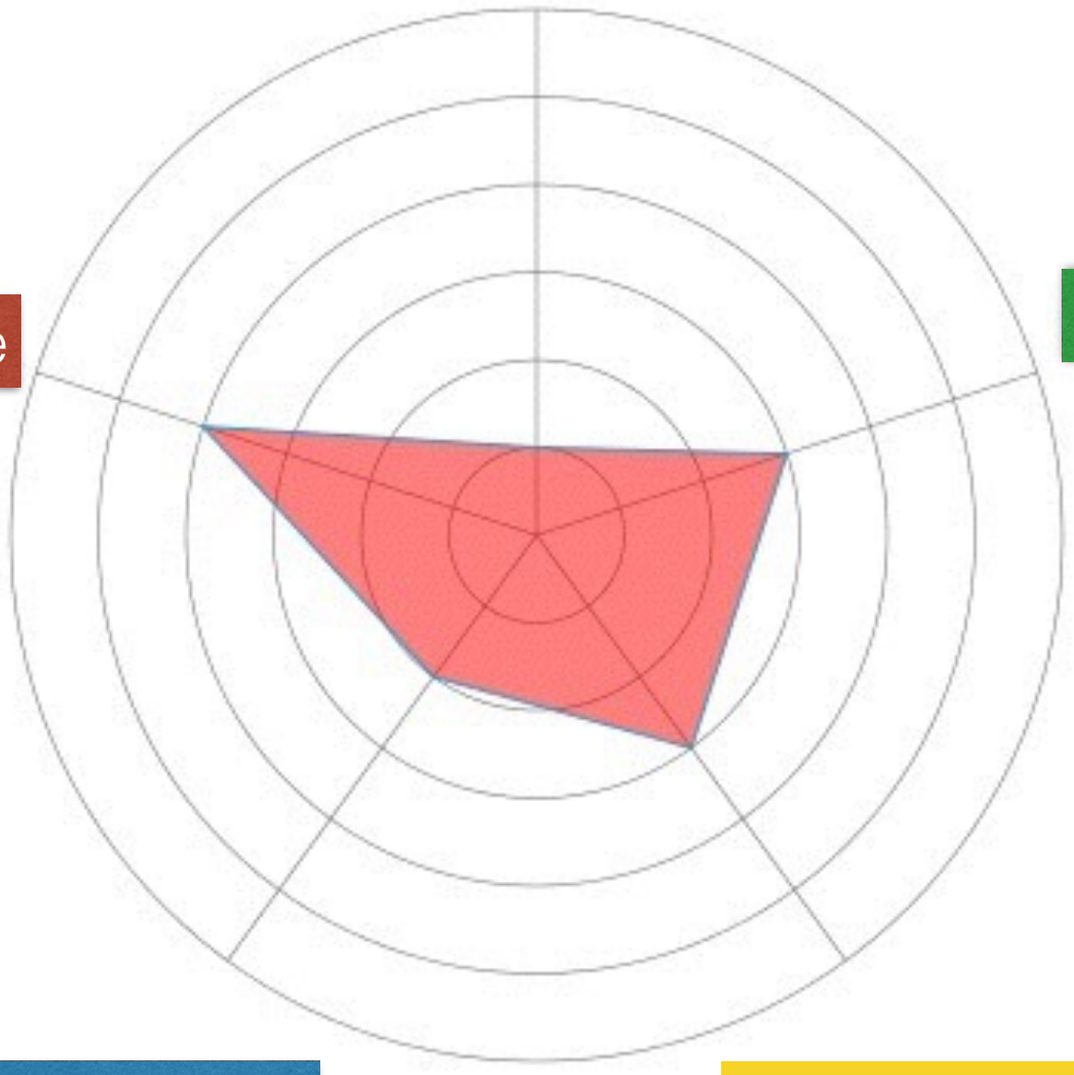
Gemeinsame Vision

Fokus auf kreative Arbeit

Transparenz: Vertrauen statt Kontrolle

Git

Wahrnehmung von Technologie/Technikern



Fail > Learn > Improve

Gemeinsame Vision

Fokus auf kreative Arbeit

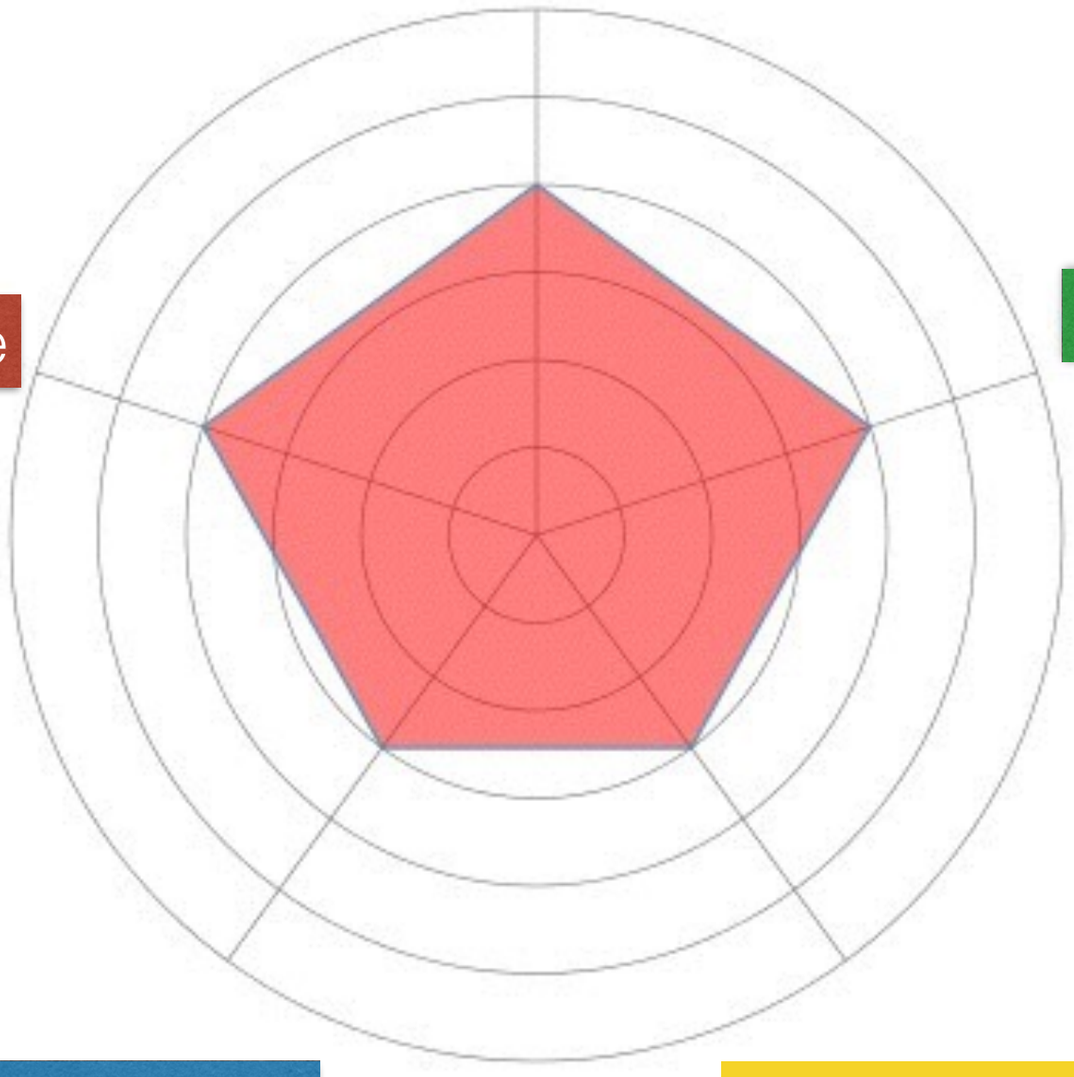
Transparenz: Vertrauen statt Kontrolle

GitHub

(verbindet Git und Planungs-Tool)

Wahrnehmung von Technologie/Technikern

Fail > Learn > Improve



Gemeinsame Vision

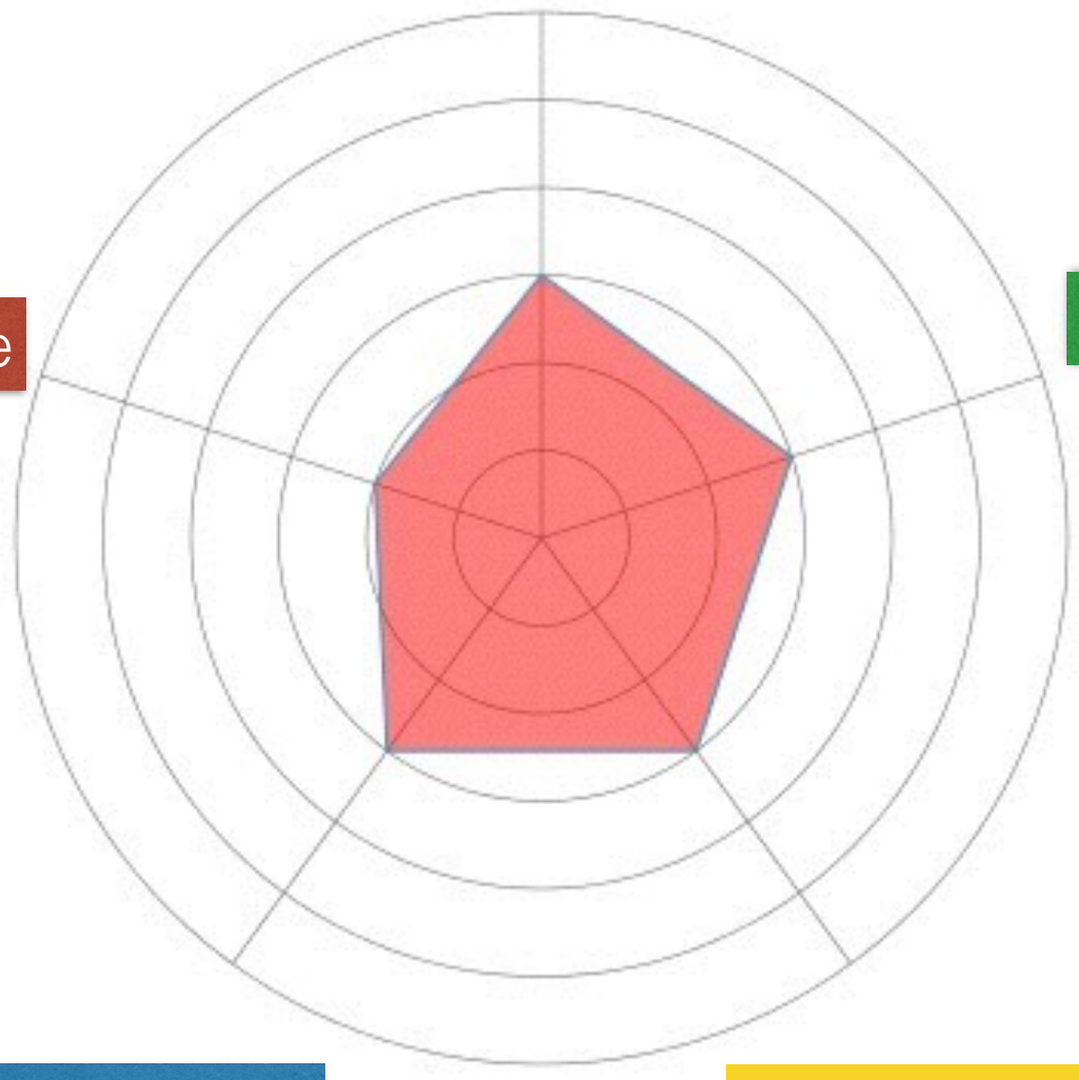
Fokus auf kreative Arbeit

Transparenz: Vertrauen statt Kontrolle

Slack

Wahrnehmung von Technologie/Technikern

Fail > Learn > Improve



Gemeinsame Vision

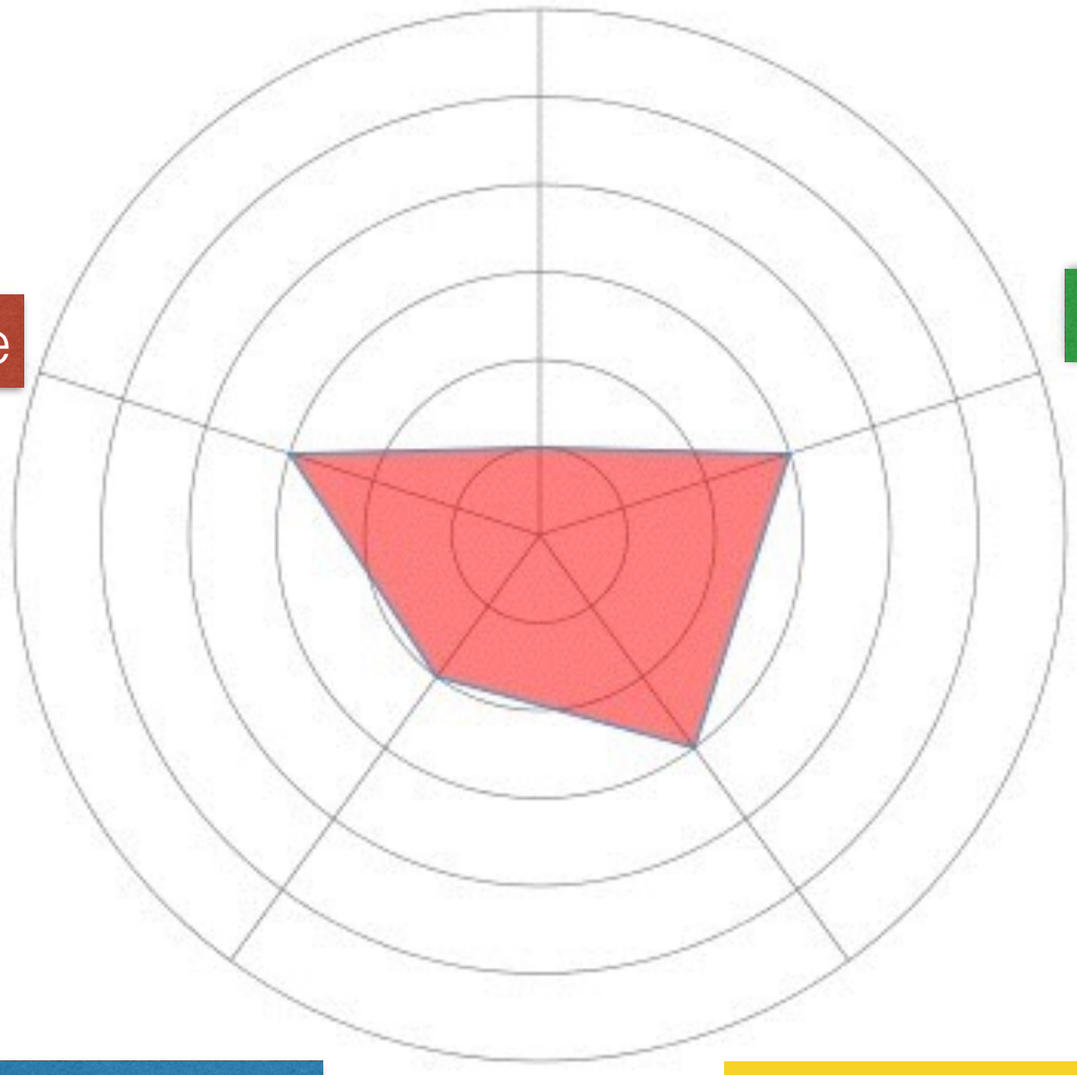
Fokus auf kreative Arbeit

Transparenz: Vertrauen statt Kontrolle

Scrum (or Kanban, or Scrumban, or Scrum of Scrums or *<fill in agile buzzword here>*)

Wahrnehmung von Technologie/Technikern

Fail > Learn > Improve



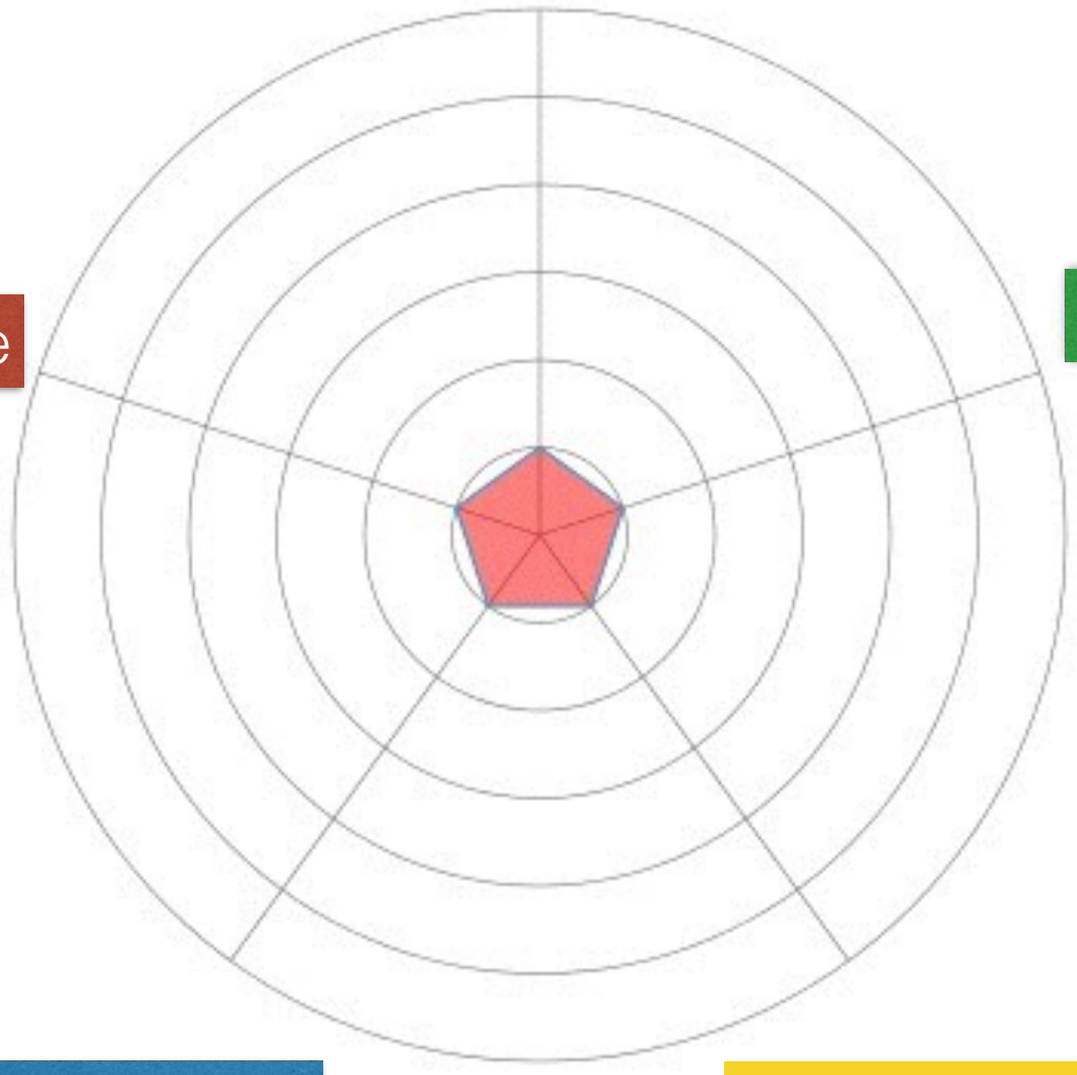
Gemeinsame Vision

Fokus auf kreative Arbeit

Transparenz: Vertrauen statt Kontrolle

Code > Compile > Run

Wahrnehmung von Technologie/Technikern



Gemeinsame Vision

Fail > Learn > Improve

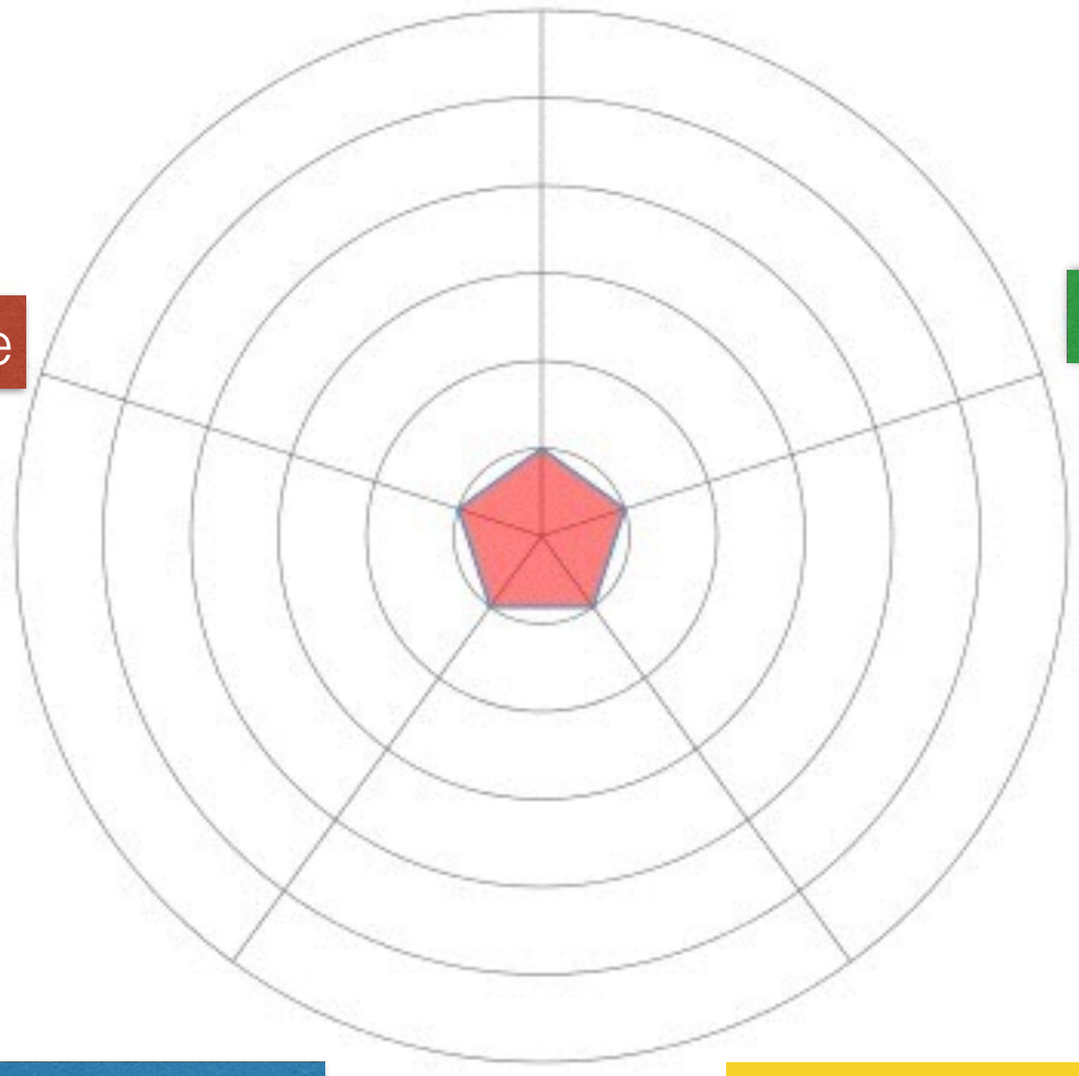
Fokus auf kreative Arbeit

Transparenz: Vertrauen statt Kontrolle

Object-Oriented Programming

Wahrnehmung von Technologie/Technikern

Fail > Learn > Improve



Gemeinsame Vision

Fokus auf kreative Arbeit

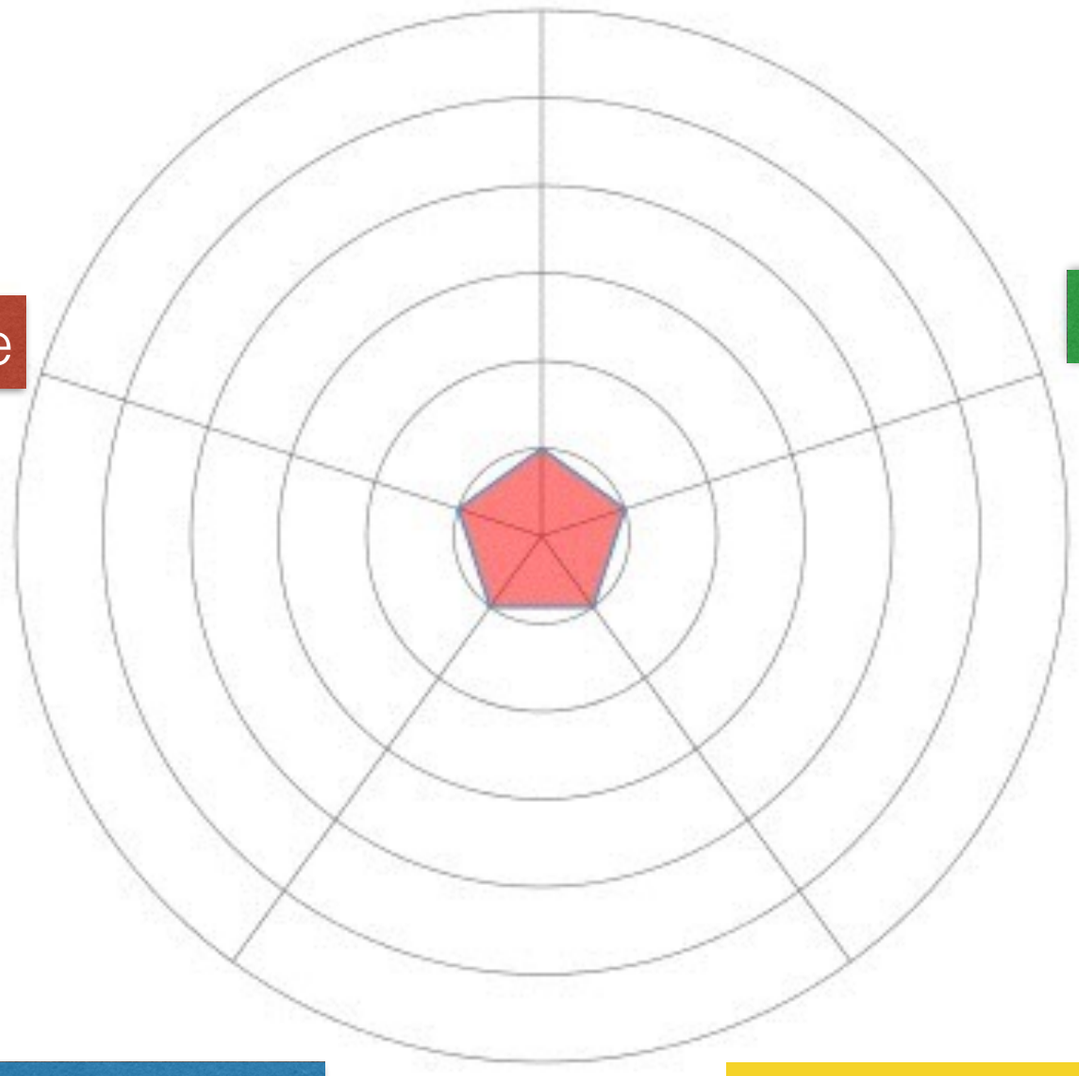
Transparenz: Vertrauen statt Kontrolle

Enterprise Architectures/Frameworks

(maybe not all of them ;))

Wahrnehmung von Technologie/Technikern

Fail > Learn > Improve



Gemeinsame Vision

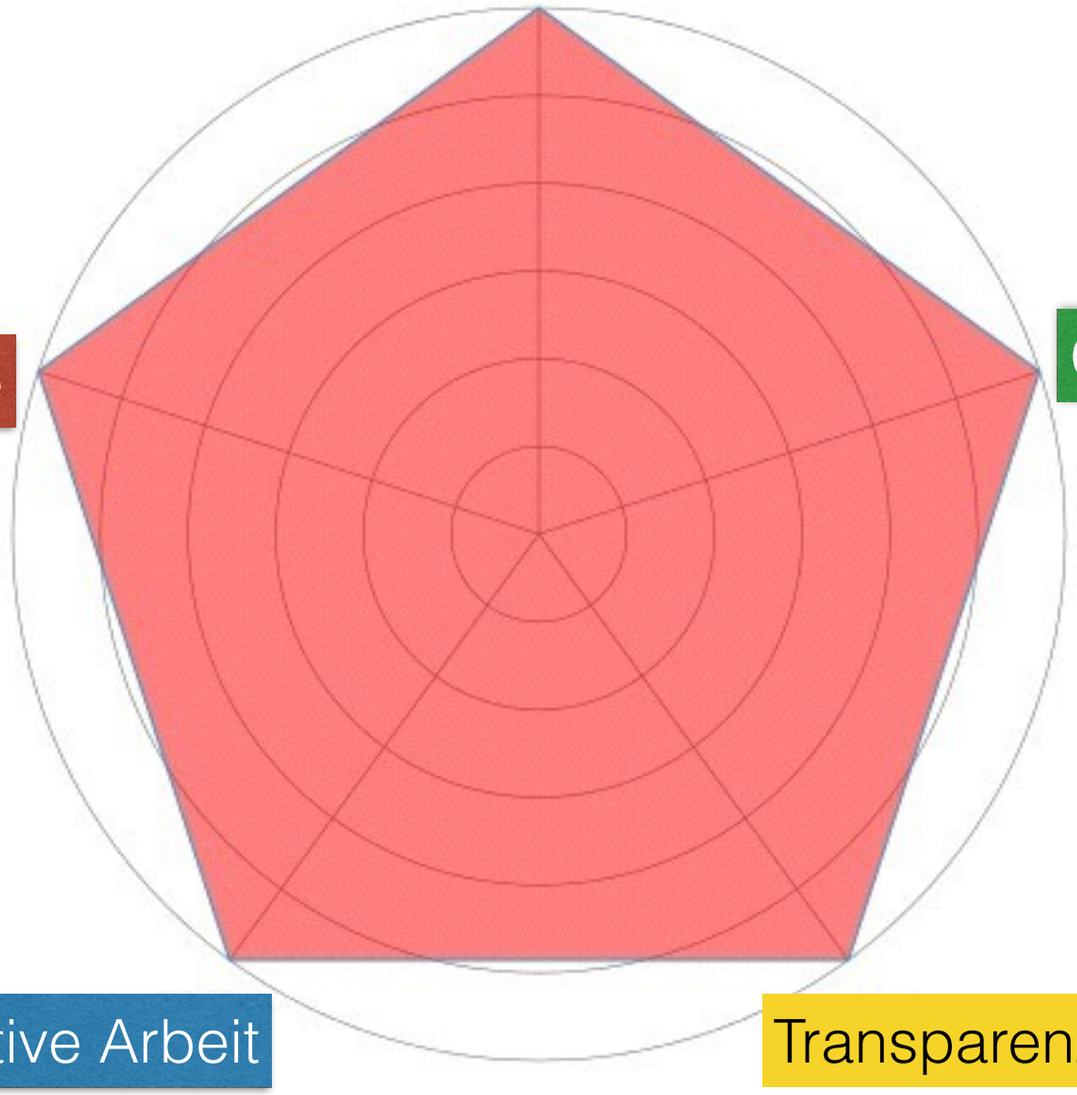
Fokus auf kreative Arbeit

Transparenz: Vertrauen statt Kontrolle

What do we need?

“The Super Tool”?

Wahrnehmung von Technologie/Technikern



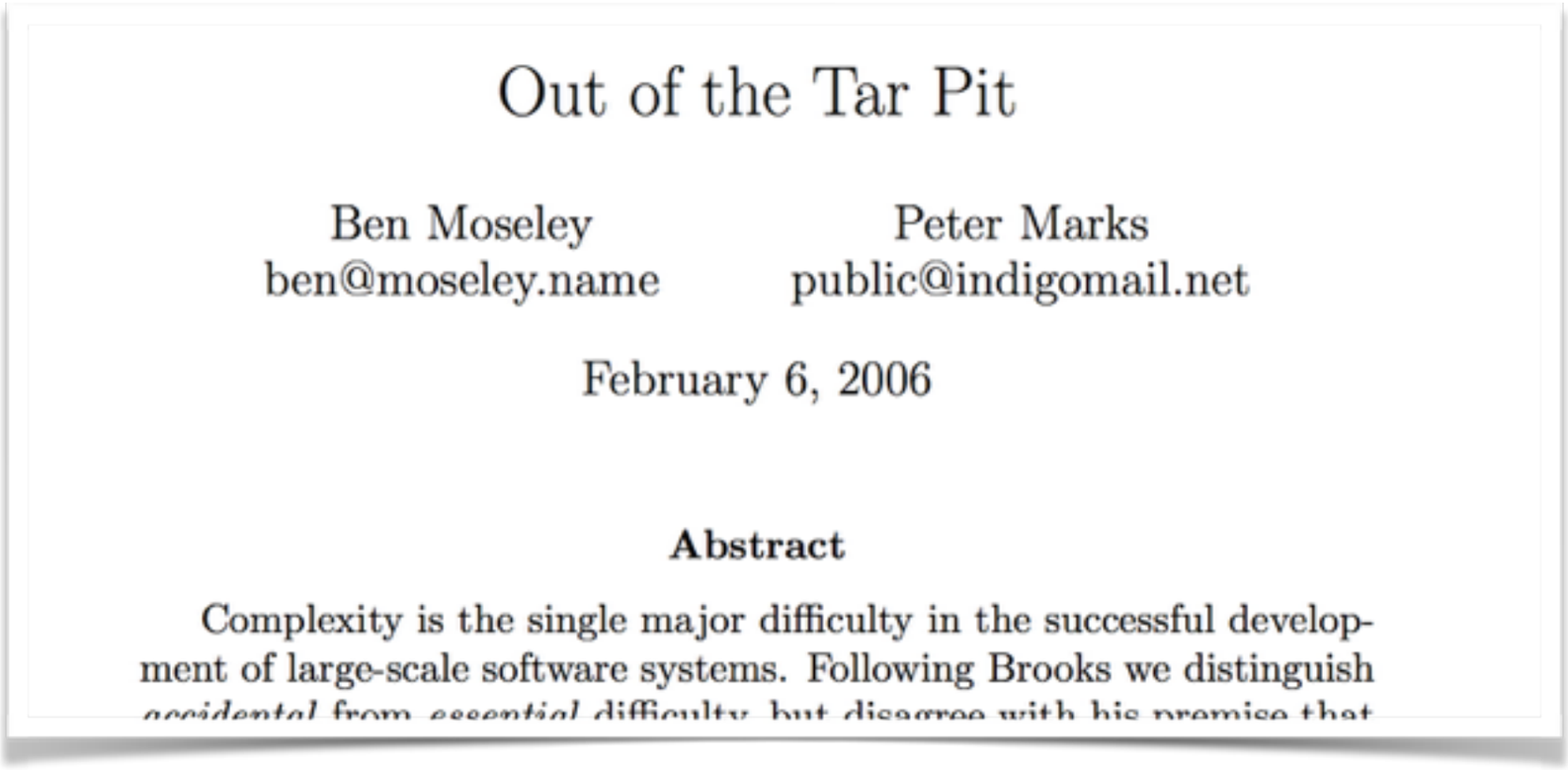
Gemeinsame Vision

Fail > Learn > Improve

Fokus auf kreative Arbeit

Transparenz: Vertrauen statt Kontrolle

Komplex? und **Kompliziert?**



Out of the Tar Pit

Ben Moseley
ben@moseley.name

Peter Marks
public@indigomail.net

February 6, 2006

Abstract

Complexity is the single major difficulty in the successful development of large-scale software systems. Following Brooks we distinguish *accidental* from *essential* difficulty, but disagree with his premise that

Complexity is *the* root cause of the vast majority of problems with software today.

Moseley, Marks; - http://shaffner.us/cs/papers/tarpit.pdf
summary by Adrian Colyer's *the morning paper* - https://blog.acolyer.org/2015/03/20/out-of-the-tar-pit

We need a
Software Development System

…that takes away
accidental complexity*
by using the right
criteria for decomposition

and enables **us**** to focus on
solving complex problems of our users

*<u>*technical*</u> and <u>*human interaction*</u>
accidental complexity

****us*** = everyone (=every creative)
involved

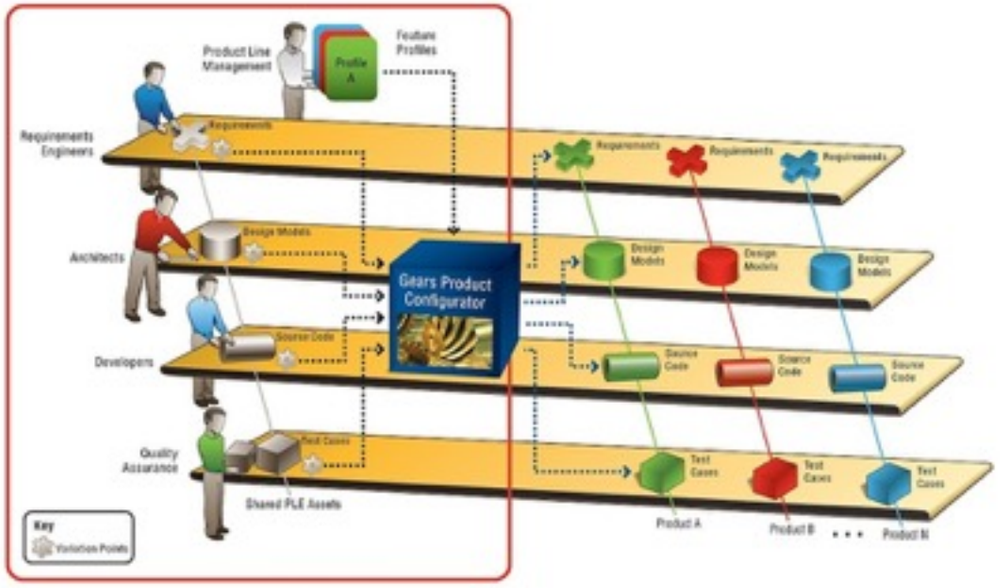
How do we get there?

Where do we start?

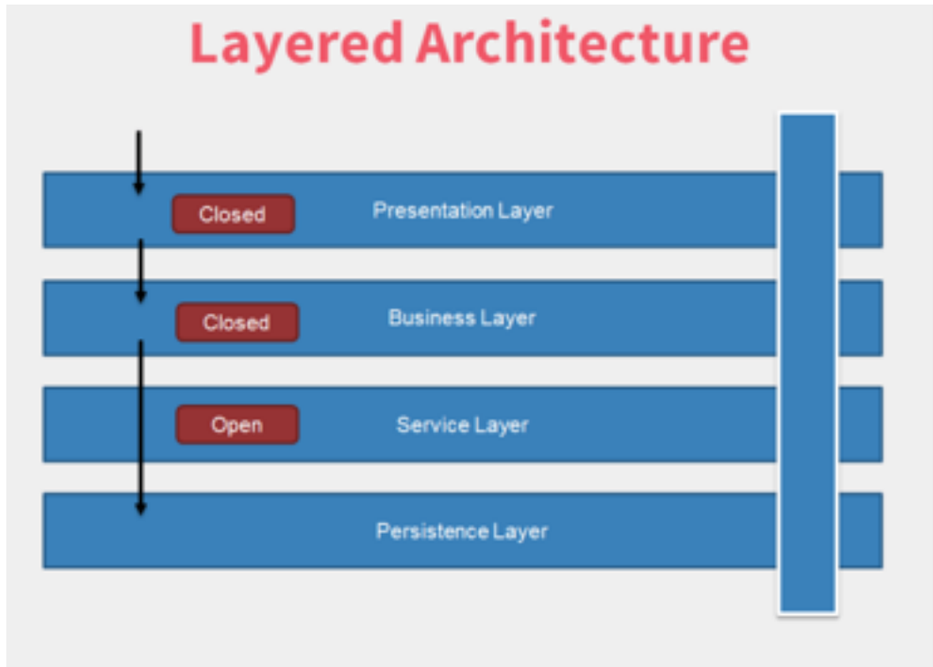
Fail > Learn > Improve Was muss sich ändern?

- Fähigkeit zu Variation nicht auf die Software draufbauen, sondern als Grundlage etablieren

- Verfügbarkeit von Daten wichtiger einstufen als Datenreduktion



http://www.biglever.com/overview/software_product_lines.html

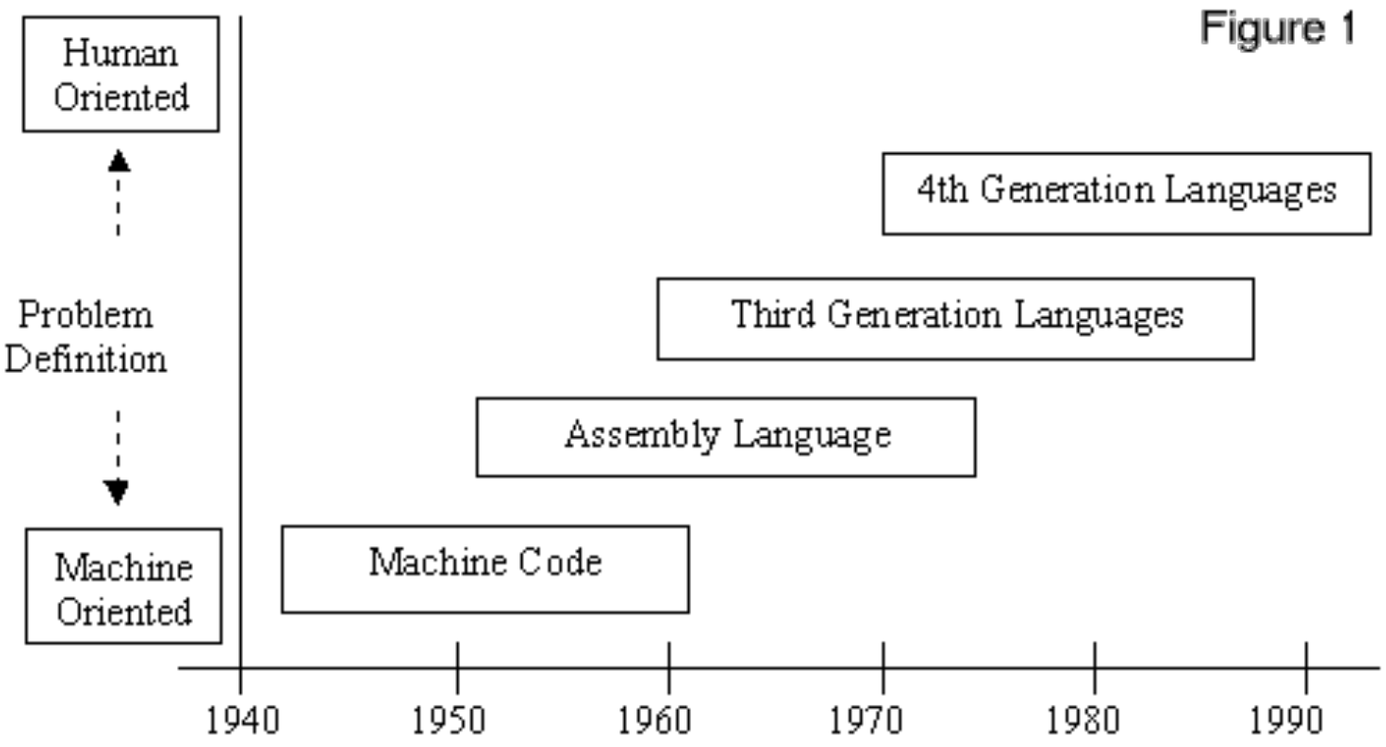


Gemeinsame Vision

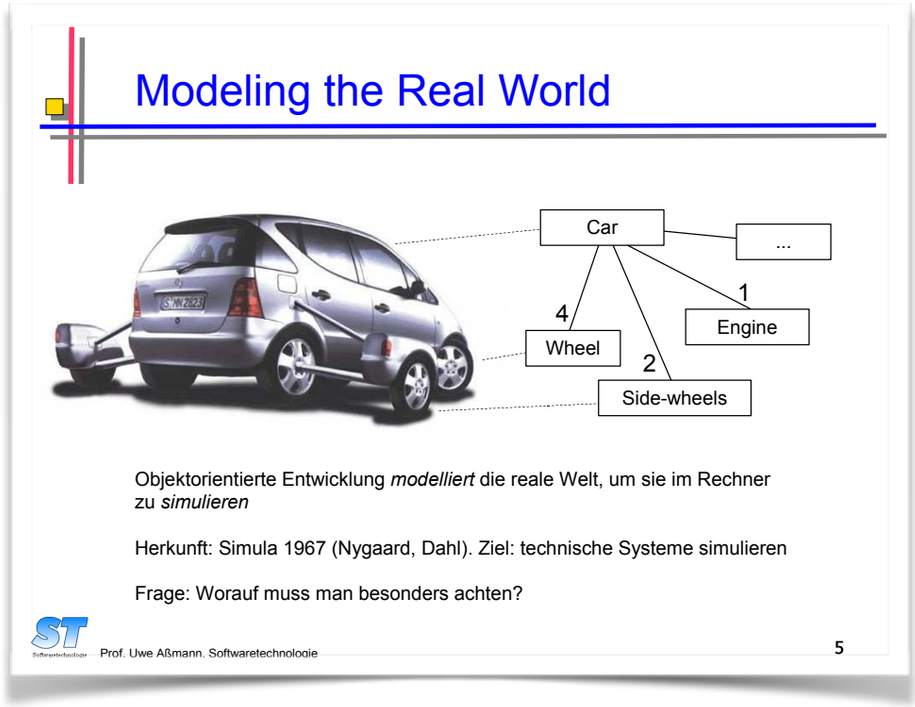
Was muss sich ändern?

- Wir brauchen “Human-Oriented”, aber komplette, Sprachen für Domänen-Modell und Logik (um gemeinsam dran zu arbeiten)

- Wir sollten Object-Orientiertheit als **die** Lösung die Welt zu modellieren hinterfragen



http://users.evtek.fi/~jaanah/IntroC/DBeech/3gl_intro.htm



Fokus auf kreative Arbeit

Was muss sich ändern?

- Wir müssen aufhören Entwickler (und als Entwickler uns selber) als nicht-kreative zu betrachten

http://beyondthecubicle.org/wp-content/uploads/2014/12/cubiclesA.jpg

Literatur

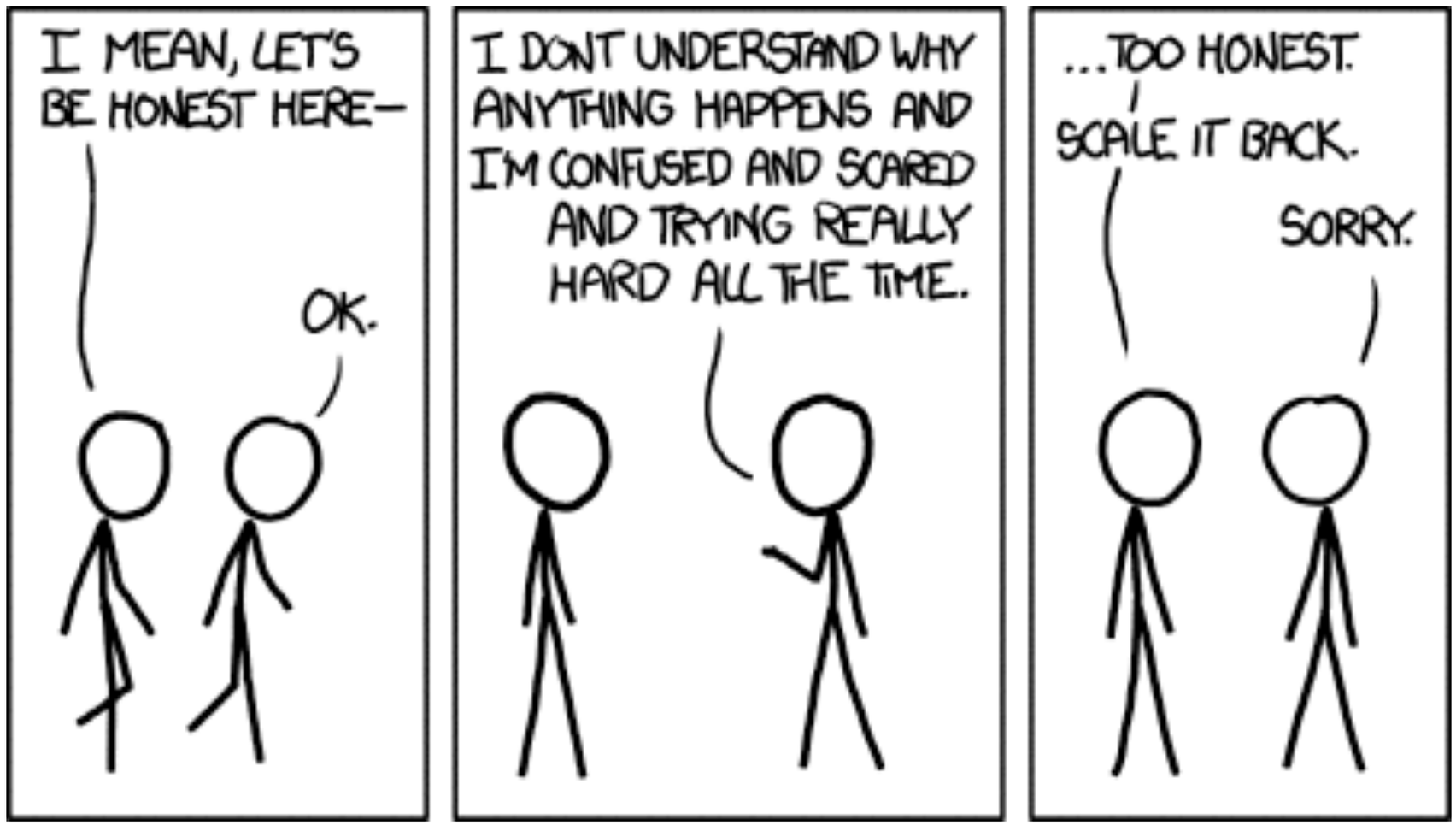
- **Out of the Tar Pit (Moseley, Marks)**
  https://blog.acolyer.org/2015/03/20/out-of-the-tar-pit
- **On the Criteria to be used in decomposing a System into Modules (Parnas)**
  blog.acolyer.org/2016/09/05/on-the-criteria-to-be-used-in-decomposing-systems-into-modules
- **The Clean Architecture (Uncle Bob)**
  https://8thlight.com/blog/uncle-bob/2012/08/13/the-clean-architecture.html
  Talk-Video: https://www.youtube.com/watch?v=Nsjsiz2A9mg
- **Spotify Engineering Culture (Kniberg)**
  https://www.youtube.com/watch?v=Mpsn3WaI_4k
- **Inventing on Principle (Victor)**
  http://worrydream.com/#!/InventingOnPrinciple

Weiterführende Literatur

- **Karl Popper und agile Produktentwicklung** (Patrick Breitenbach, Tobias Baier)
  Podcast Agiles Produktmanagement: http://www.agilesproduktmanagement.de/2016/03/apm-15-karl-popper-und-agile-produktentwicklung-mit-patrick-breitenbach
- **Spiel als Arbeit** (Fabian Hoose, Daniel Raumer)
  Podcast Insert Moin Ep. 1524: http://insertmoin.de/im1524-buch-spiel-als-arbeit
- **Was macht eigentlich Market Analytics?** (Michael Lenz, Christian Schmidt, Daniel Raumer)
  Podcast Insert Moin Ep. 1260: http://insertmoin.de/im1260-was-macht-eigentlich-market-analytics
- **Work to Publish** (Jack Conte)
  PatreCon: https://www.youtube.com/watch?v=5r5VIlMhftM
- **Die Rückkehr der Vernunft** (Lars Jankowfsky, Johann-Peter Hartmann)
  Konferenzvortrag, CodeTalks 2016: https://www.youtube.com/watch?v=XXPl1UgDwh8
- **Eve: Programming designed for humans**
  http://witheve.com

Thanks!

jendrik.johannes@gmail.com | @jeoj | jjohannes.de



Honest - https://xkcd.com/1146