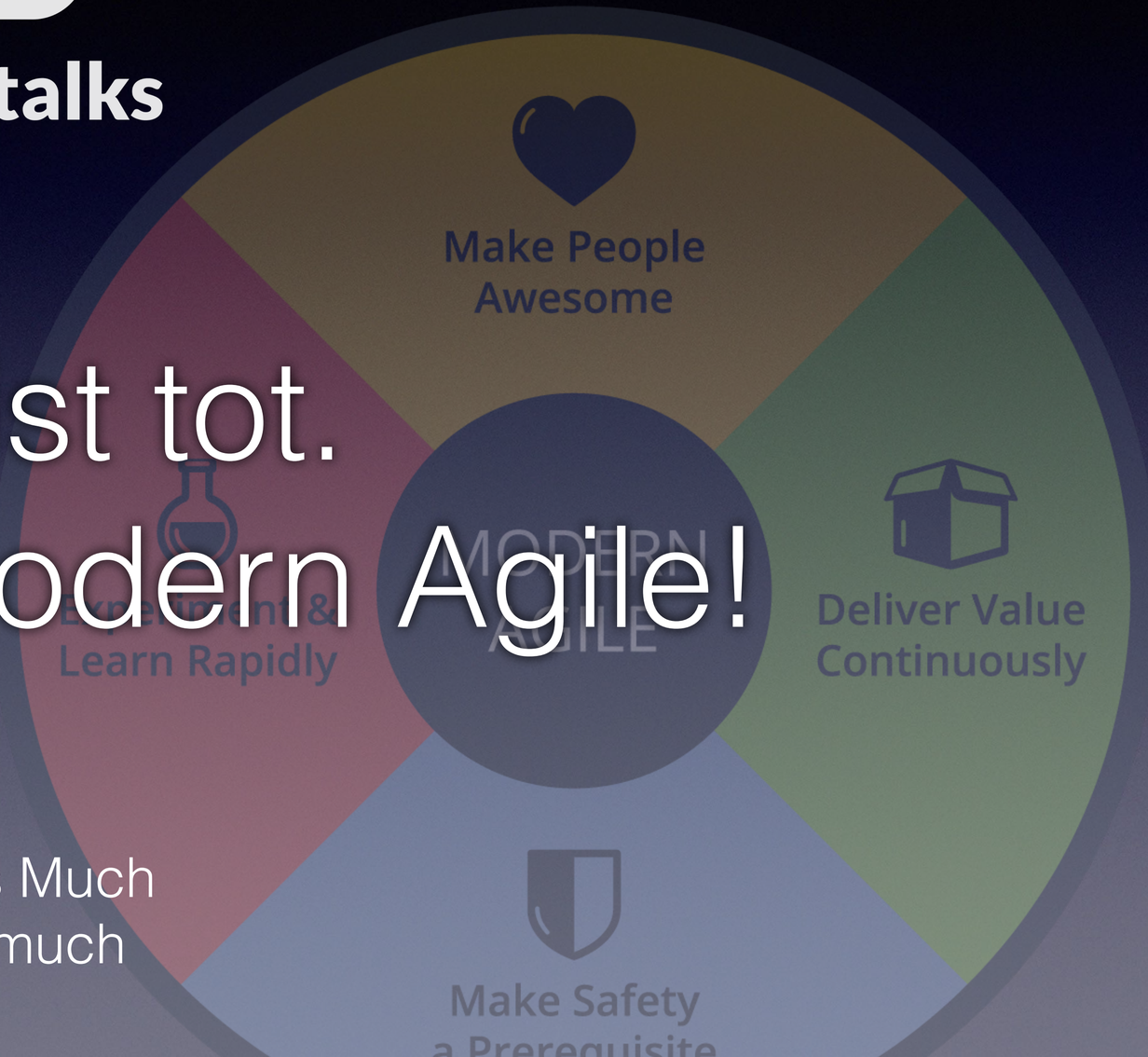
code.talks

Agile ist tot.
Lang lebe Modern Agile!



Thomas Much
@thmuch

Softwareentwickler

Agile Developer Coach

@thmuch

2019

https://pixabay.com/de/photos/deutsches-nationaltheater-weimar-2405690/ https://pixabay.com/de/photos/bauhaus-text-schreiben-schriften-2588026/


70 Jahre
GRUNDGESETZ
für die Bundesrepublik Deutschland

```
IDENTIFICATION DIVISION.
  PROGRAM-ID. HELLO-WORLD.

ENVIRONMENT DIVISION.

DATA DIVISION.
```

60 Jahre

```
PROCEDURE DIVISION.
PARA-1.
  DISPLAY "Hello, world.".

     EXIT PROGRAM.
  END PROGRAM HELLO-WORLD.
```

50 Jahre

https://de.wikipedia.org/wiki/Datei:Boeing_747_Official_Colors.jpg

https://images.nasa.gov/details-as11-40-5878.html

```
-rw-r--r--  1 bin      492 Mar 21 12:07 conf.o
drwxr-xr-x  2 bin      400 Dec  2 18:20 dmr
-rw-r--r--  1 bin      157 Nov 26 18:13 file.h
-rw-r--r--  1 bin      188 Nov 26 18:13 filsys.h
-rw-r--r--  1 bin      581 Nov 26 18:13 inode.h
drwxr-xr-x  2 bin      352 Nov 26 18:13 ken
-rw-r--r--  1 bin       35 Nov 26 18:13 ld
-rw-r--r--  1 bin    48158 Nov 26 18:13 lib1
-rw-r--r--  1 bin    39670 Dec  2 18:20 lib2
-rw-r--r--  1 bin      816 Mar 21 12:06 low.o
-rw-r--r--  1 bin     3744 Mar 21 12:07 mch.o
-rw-r--r--  1 bin      957 Nov 26 18:13 param.h
-rw-r--r--  1 bin      386 Nov 26 18:13 proc.h
-rw-r--r--  1 bin      142 Nov 26 18:13 reg.h
-rw-r--r--  1 bin      217 Nov 26 18:13 seg.h
-rw-r--r--  1 bin      422 Nov 26 18:13 systm.h
-rw-r--r--  1 bin      115 Nov 26 18:13 text.h
-rw-r--r--  1 bin      868 Nov 26 18:13 tty.h
-rw-r--r--  1 bin     1217 Nov 26 18:13 user.h
# ls -l /unix
-rwxrwxrwx  1 bin    25802 Mar 21 12:07 /unix
#
```

50 Jahre

35 Jahre

https://pixabay.com/de/photos/geburtstag-geburtstagstorte-kuchen-1835443/

Manifesto for Agile Software Development

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on
the right, we value the items on the left more.

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler
James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick
Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

https://agilemanifesto.org/

Manifesto for Agile Software Development

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

https://agilemanifesto.org/

I ❤️ Agile 🔥

ABANDON

„Developers Should Abandon Agile“

AGILE

– Ron Jeffries

https://ronjeffries.com/articles/018-01ff/abandon-1/

Was war Agile denn früher?

Extreme Programming

Feature Driven Development

Crystal

Scrum

Dynamic Systems Development Method

Adaptive Software Development

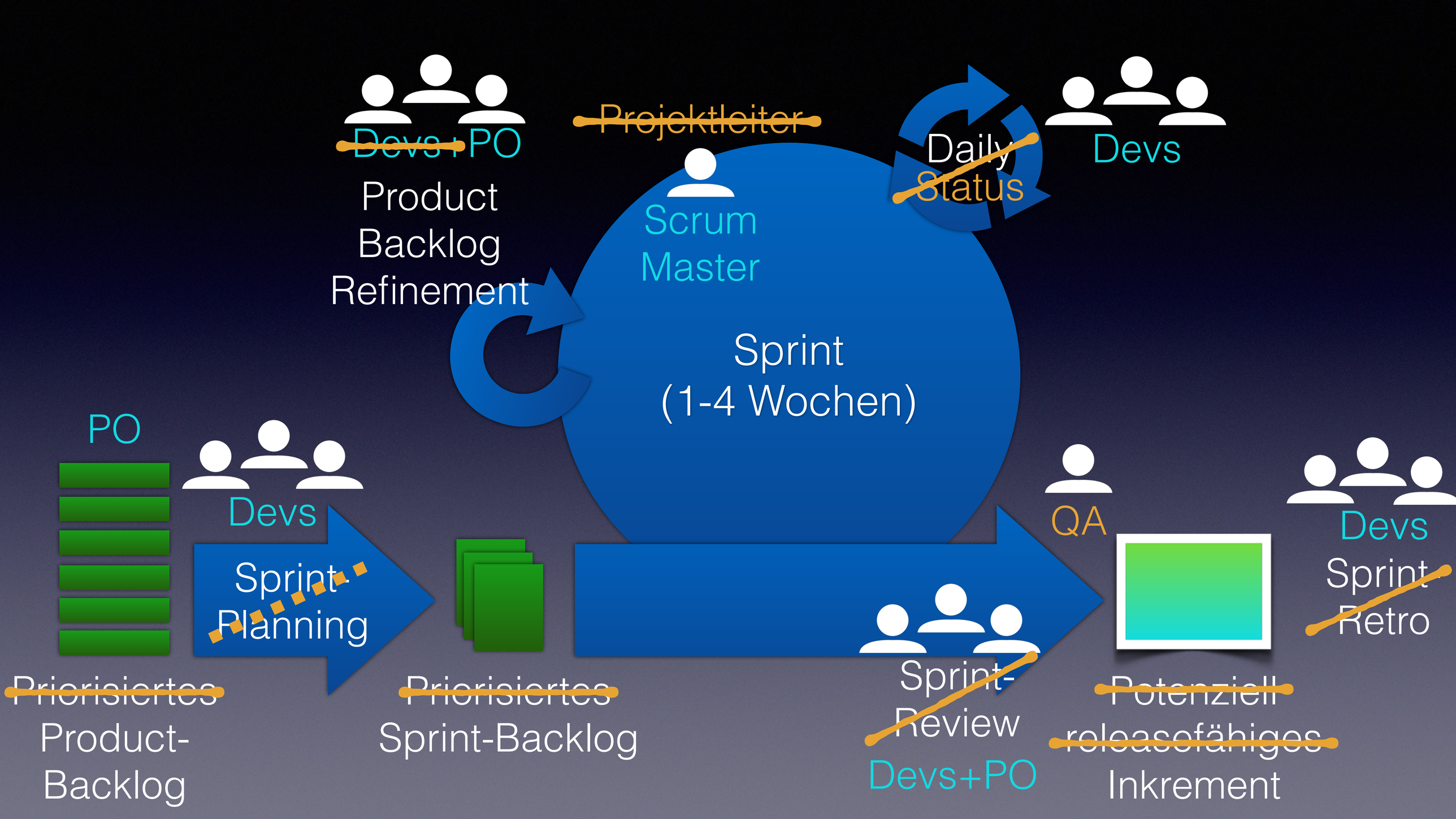
1990er

Und heute?

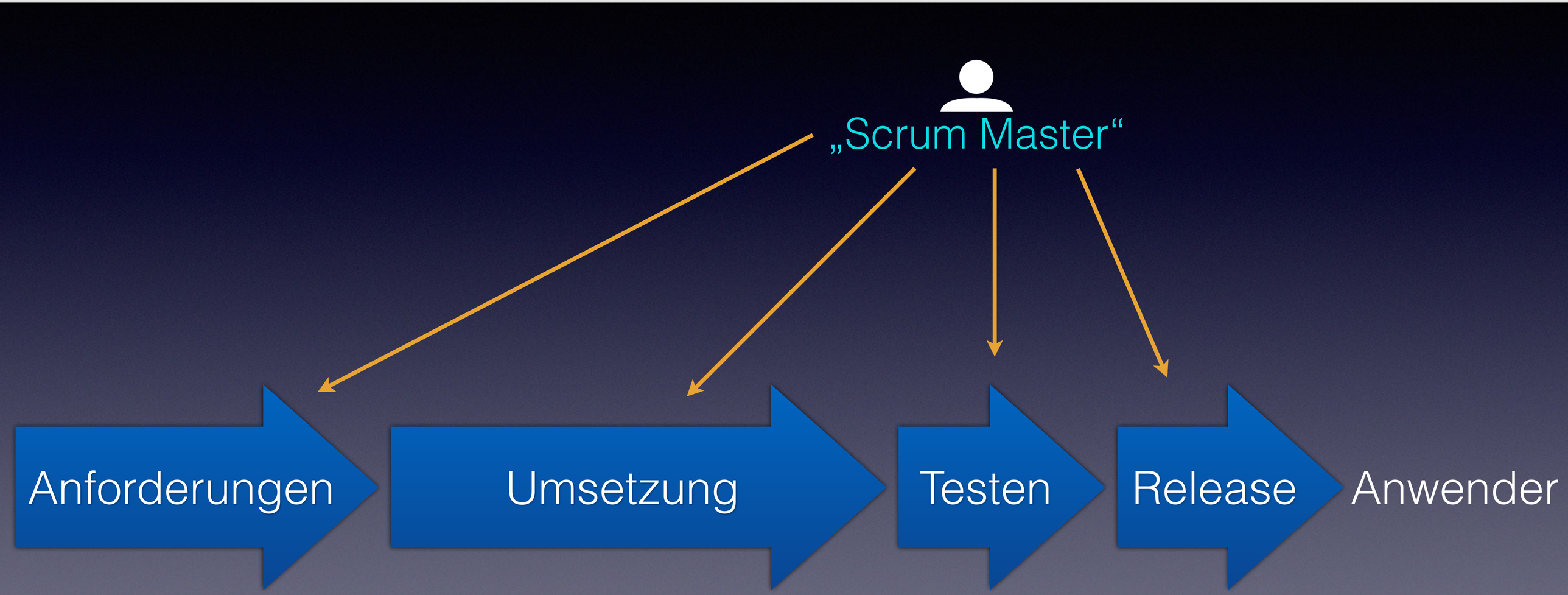
Wir machen Scrum
(das machen die anderen auch)

Wir kaufen JIRA
(das haben die anderen auch)

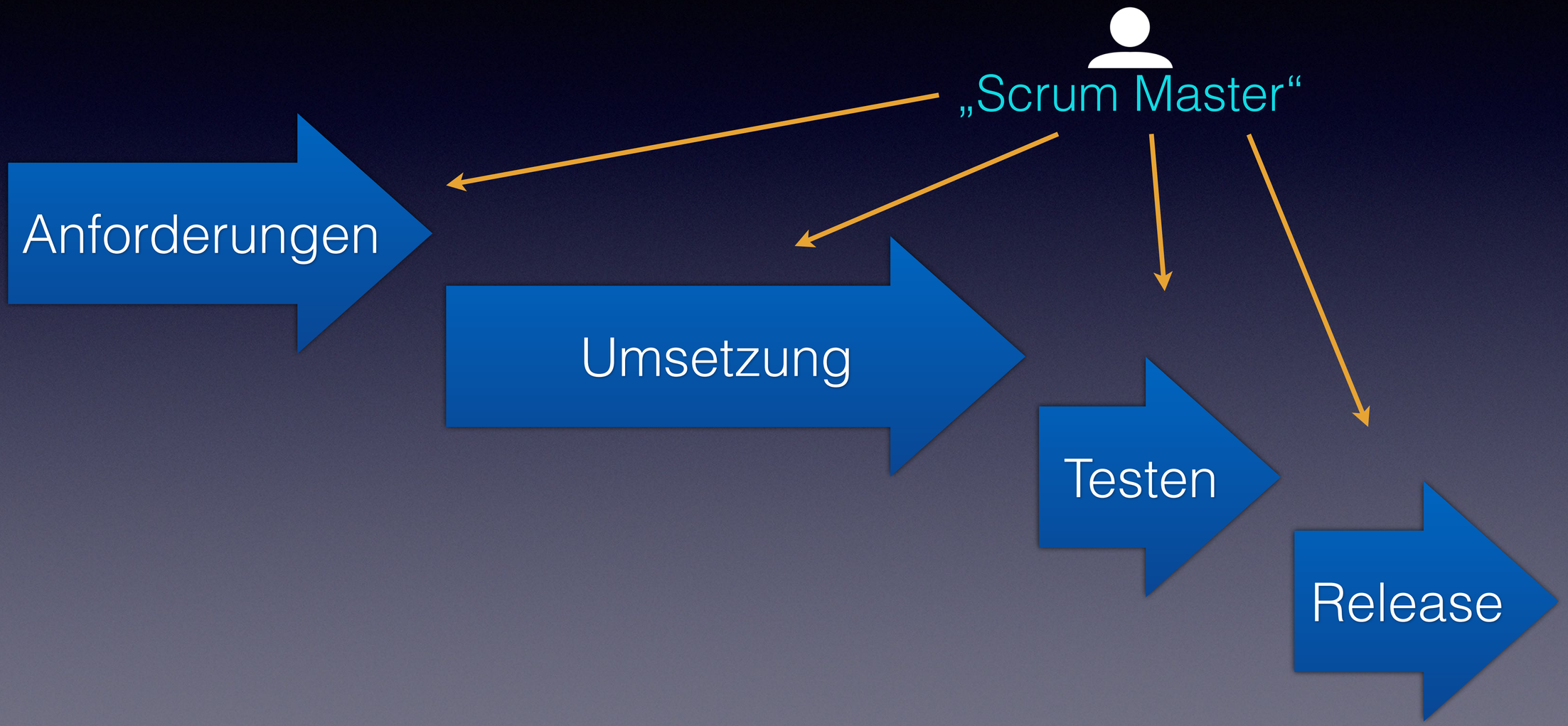
Wir skalieren Scrum über alle unsere Teams
(damit unsere KPIs passen)


~~Devs+PO~~
Product
Backlog
Refinement
~~Projektleiter~~
Scrum
Master
Daily
~~Status~~
Devs
Sprint
(1-4 Wochen)
PO
Devs
~~Priorisiertes~~
Product-
Backlog
Sprint-
Planning
~~Priorisiertes~~
Sprint-Backlog
Sprint-
~~Review~~
Devs+PO
QA
~~Potenziell~~
~~releasefähiges~~
Inkrement
Devs
Sprint
~~Retro~~

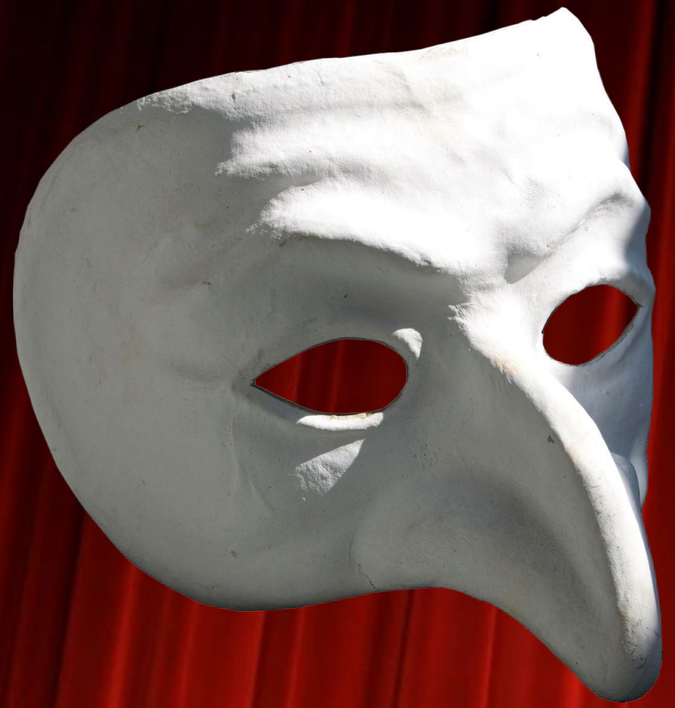
Prozess statt Agilität+Flexibilität



Prozess statt Agilität+Flexibilität



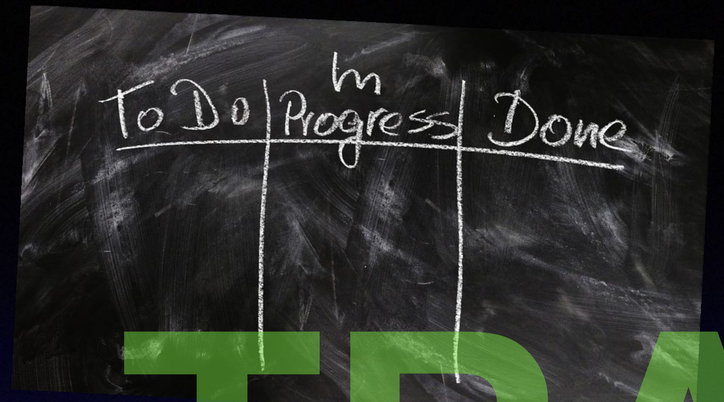
Anwende



Fake Agile

Agiles Theater

https://pixabay.com/de/photos/maske-pulcinella-pulcinella-maske-1641264/ https://pixabay.com/de/photos/vorhang-theater-las-vegas-rot-269920/



TRANS-PARENZ

Cool!

Noch mehr Kontrolle

Noch mehr Auslastung

Noch mehr Druck

https://pixabay.com/de/vectors/gui-interface-internet-programm-2311261/
https://pixabay.com/de/illustrations/arbeit-arbeitsprozess-organisieren-4096927/
https://pixabay.com/de/illustrations/postit-notiz-zettel-hinweis-3838102/

Dark Agile

https://pixabay.com/de/photos/star-wars-darth-vader-dunkel-darth-1386790/

„Agile“ hat den Fokus verloren

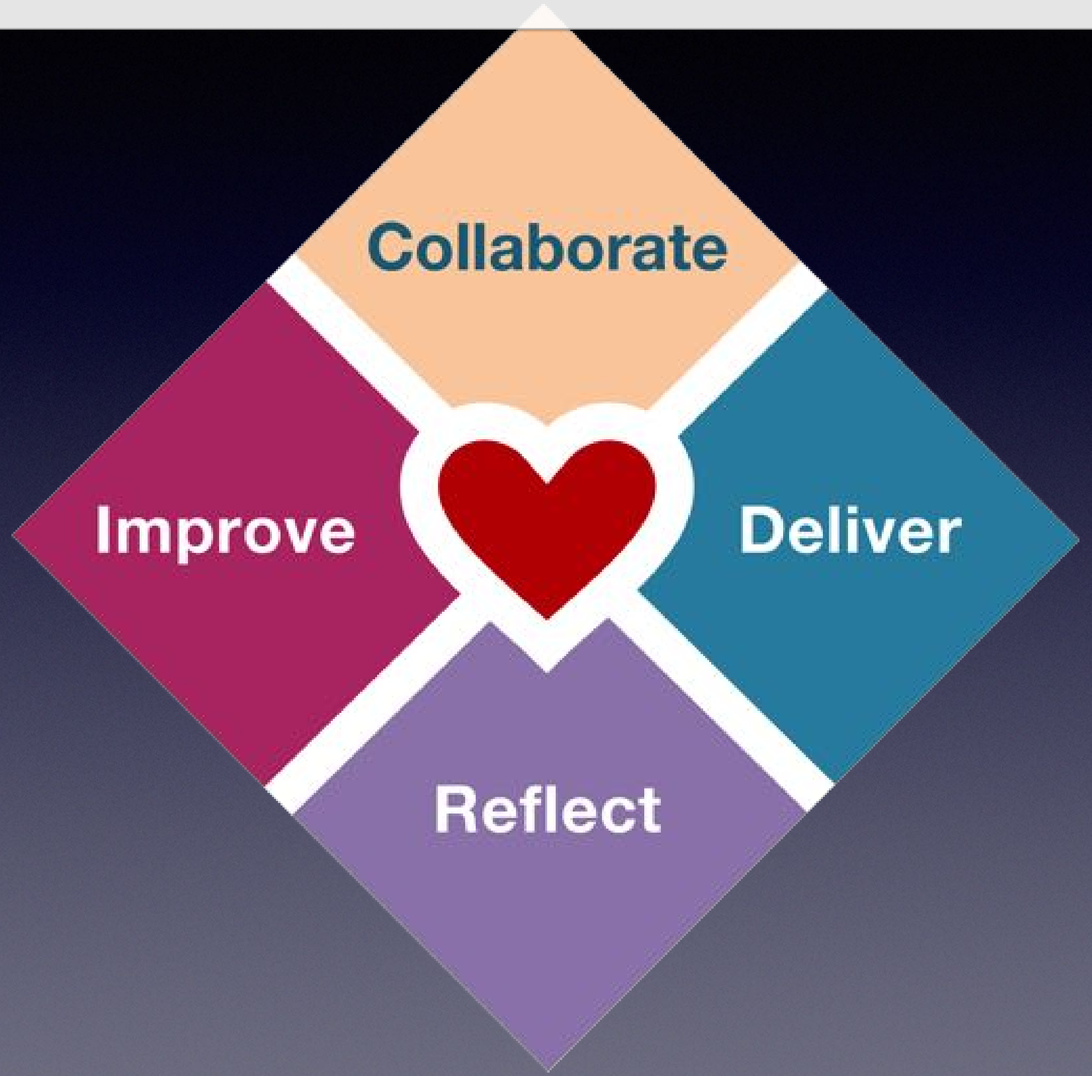
So fühlt sich „Agile“ nicht gut an.
(Und liefert keine guten Ergebnisse.)

Weichgespült, nicht mehr konsequent.

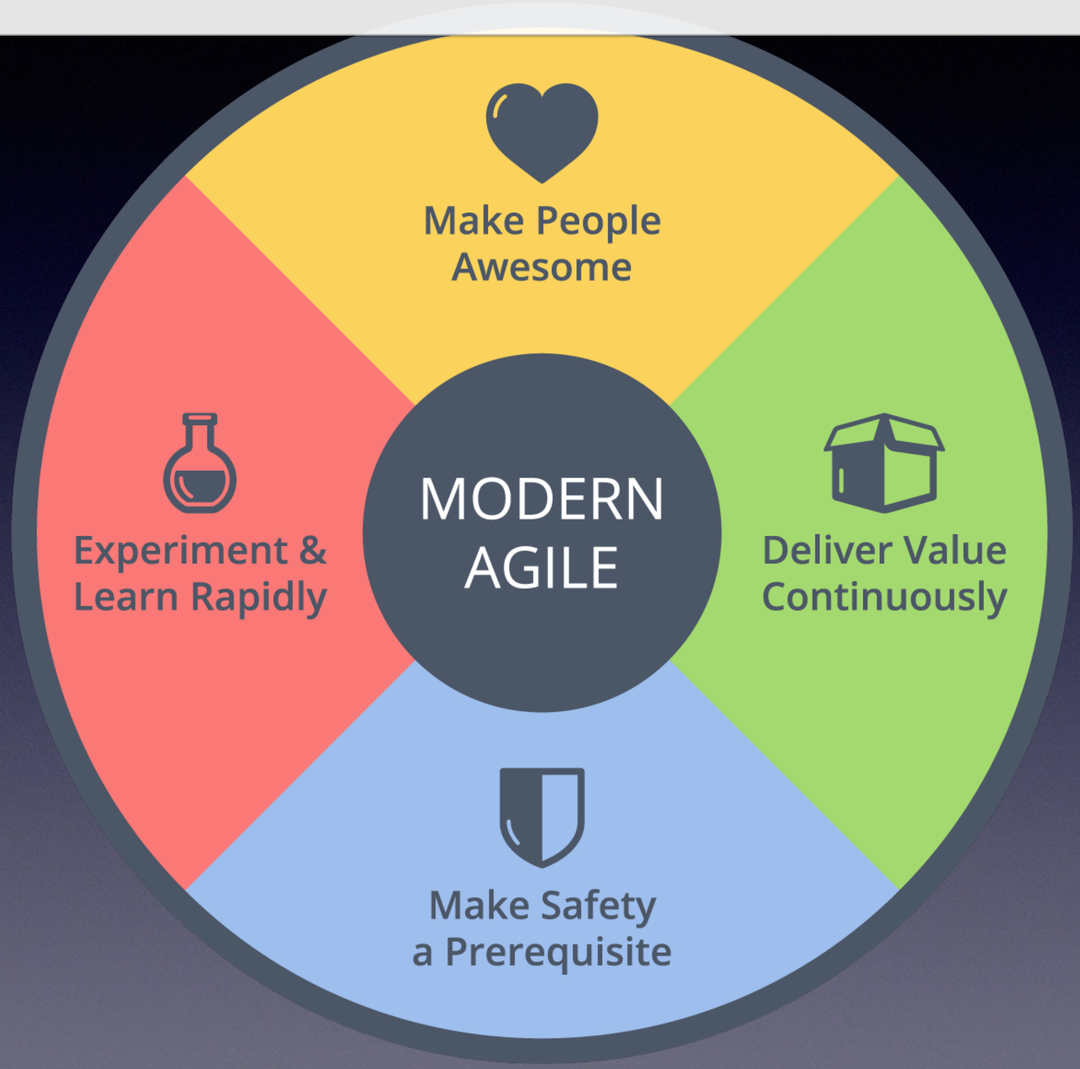
Soll zu bisherigen Vorgehensweisen & Organisationsformen passen.

Frameworks bringen Komplexität & neue Prozesse.

Fokus wieder auf den Kern!

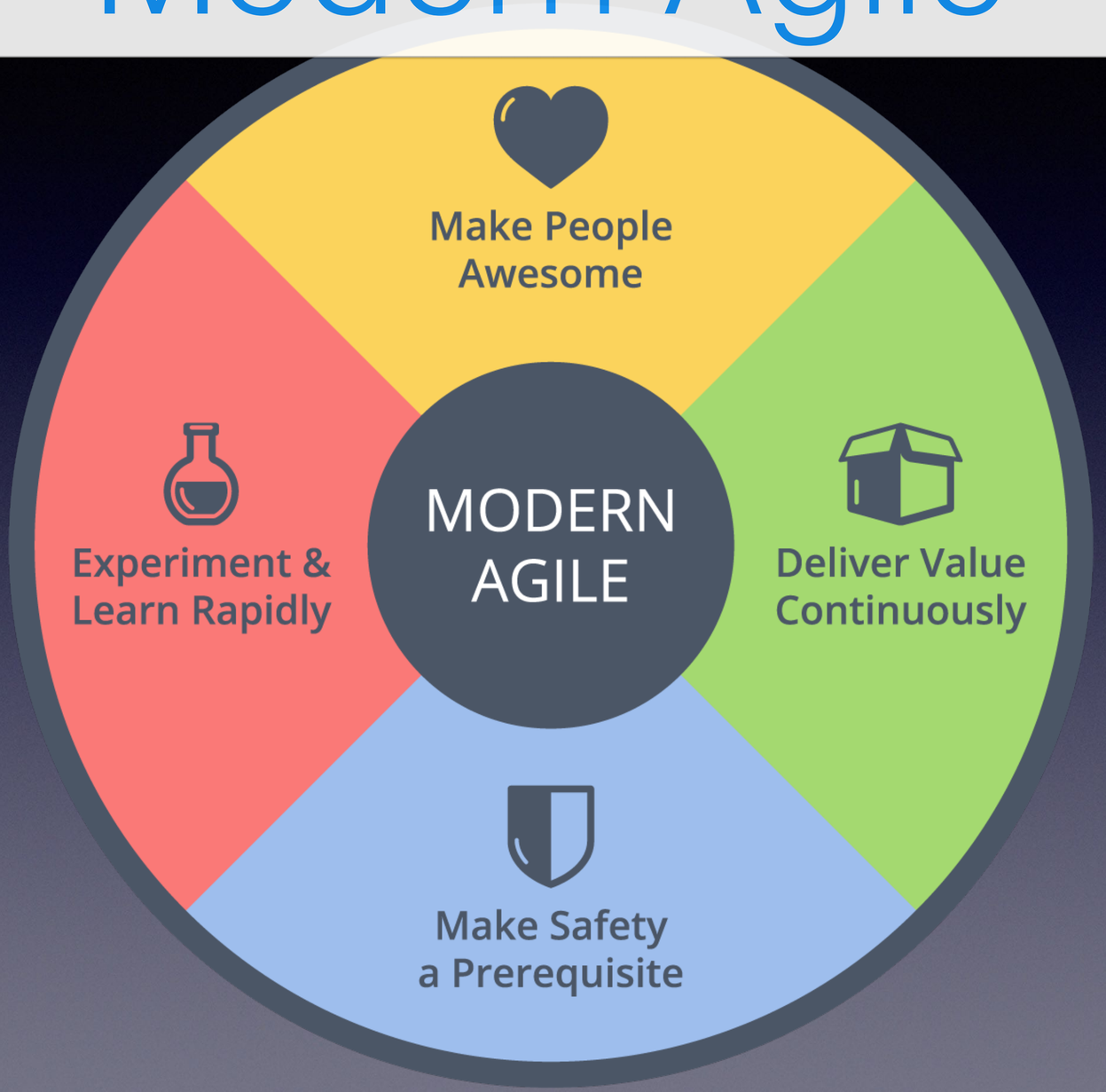


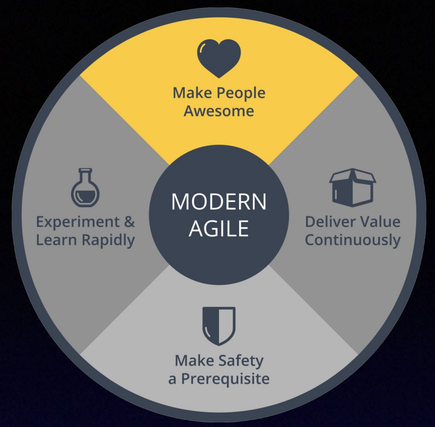
https://heartofagile.com/



http://modernagile.org/

Modern Agile

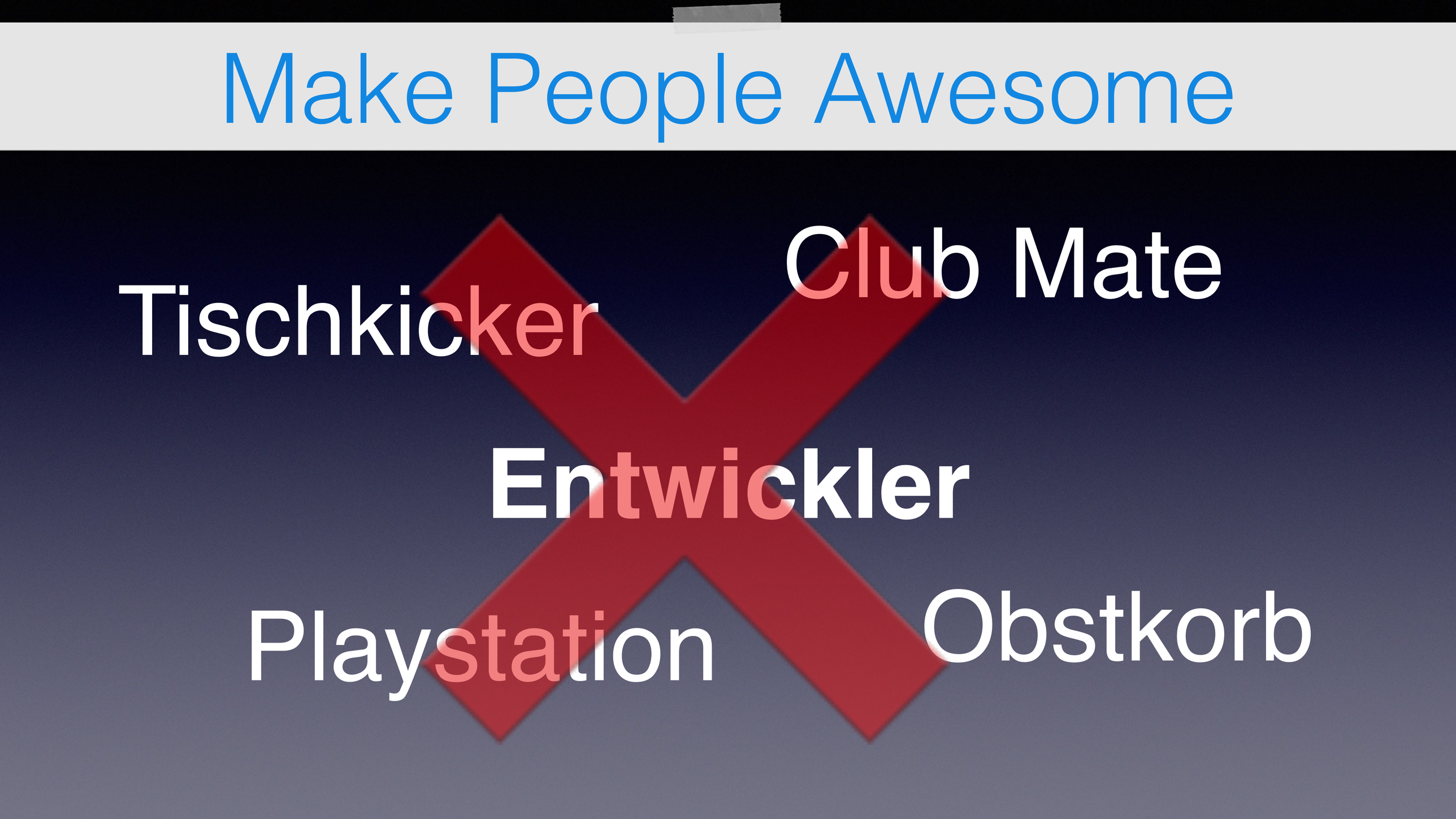




Make People Awesome

Awesome?

Make People Awesome



Make People Awesome

Entwickler **Fachbereiche**

PEOPLE

Management **Anwender**

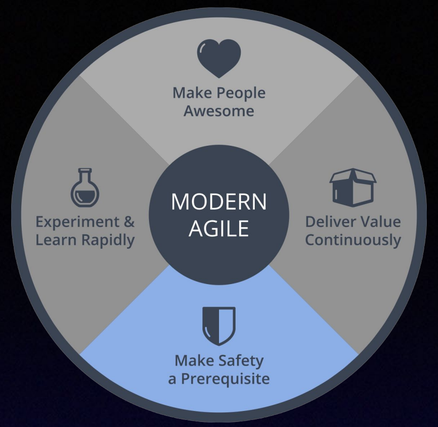
Wann fühle ich mich *awesome*?

großartig

brilliant

Was bewegen können

Sich ernst genommen fühlen



Make Safety a Prerequisite

Psychologische Sicherheit

cozy

keine Kontroversen

warm

lieb & nett

kuschelig

kein Streit

Tischkicker

gemütlich

hygge

Komfortzone

ANGST

Emotionale Sicherheit

Offene, ehrliche Gespräche
(auch wenn der Chef anwesend ist)

Fehler-Ursachen-Suche ohne Schuldzuweisungen
(kein Finger-Pointing!)

Streit über die Sache, nicht persönlich

Respekt & Wertschätzung

Ohne Angst raus aus der Komfortzone

Technische Sicherheit

Nicht mit 1 Klick eine Katastrophe auslösen können.

Datenschutz
Verschlüsselung
Geheimhaltung
Absicherung gegen Angriffe

Security by Design, DevSecOps
Autom. Pentests
Chaos Engineering

https://pixabay.com/de/photos/netz-schutz-arbeiten-dacharbeiten-3304098/

„The five keys to a successful Google team“

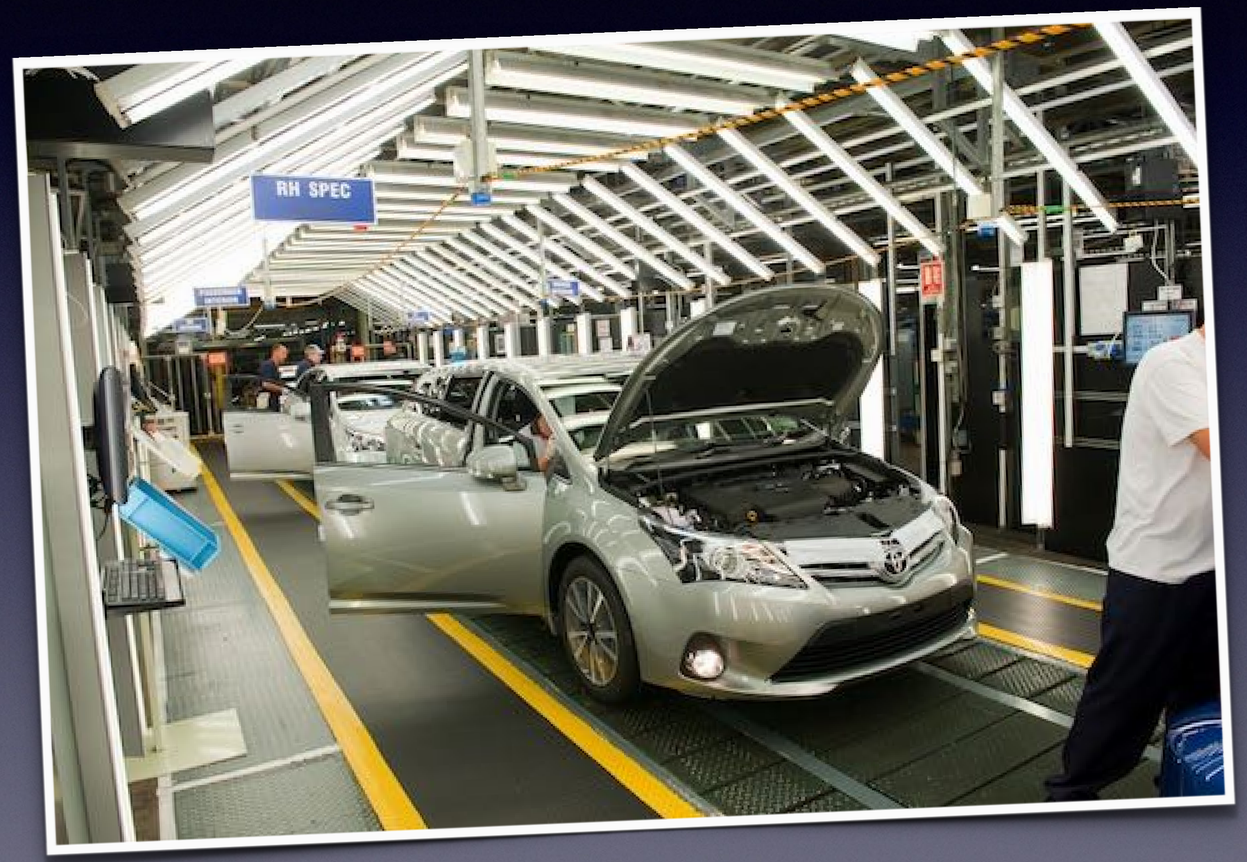
https://rework.withgoogle.com/blog/five-keys-to-a-successful-google-team/

Wie schaffen wir Sicherheit?

- Bei den Meetings starten (wie Google)
- Siehe die 5 „Meeting Agreements“:

https://github.com/modernagile/modernagile.github.io/blob/master/goodies/ma-meetingAgreements.zip

Wie schaffen wir Sicherheit?



Stop Work Authority

Stop any work or behavior you deem unsafe to yourself or others.

STOP

Be safe.
Please help protect our:

Health
Time
Money
Information
Relationships
Reputation

Safety unlocks high performance. You will never be penalized for stopping unsafe work or speaking up about hazards or injuries.

MODERNAGILE.ORG

https://blog.toyota.co.uk/andon-toyota-production-system

https://github.com/modernagile/modernagile.github.io/blob/master/goodies/swa.zip

Sicherheit

VERTRAUEN

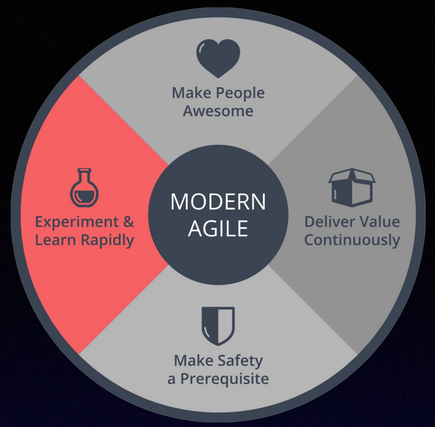
ANGST

ZUTRAUEN

Was hat das mit Agile zu tun?

Gesunder Menschenverstand?!

Auch für „normale“ Projekte gut!?!



Experiment & Learn Rapidly

Experimente?

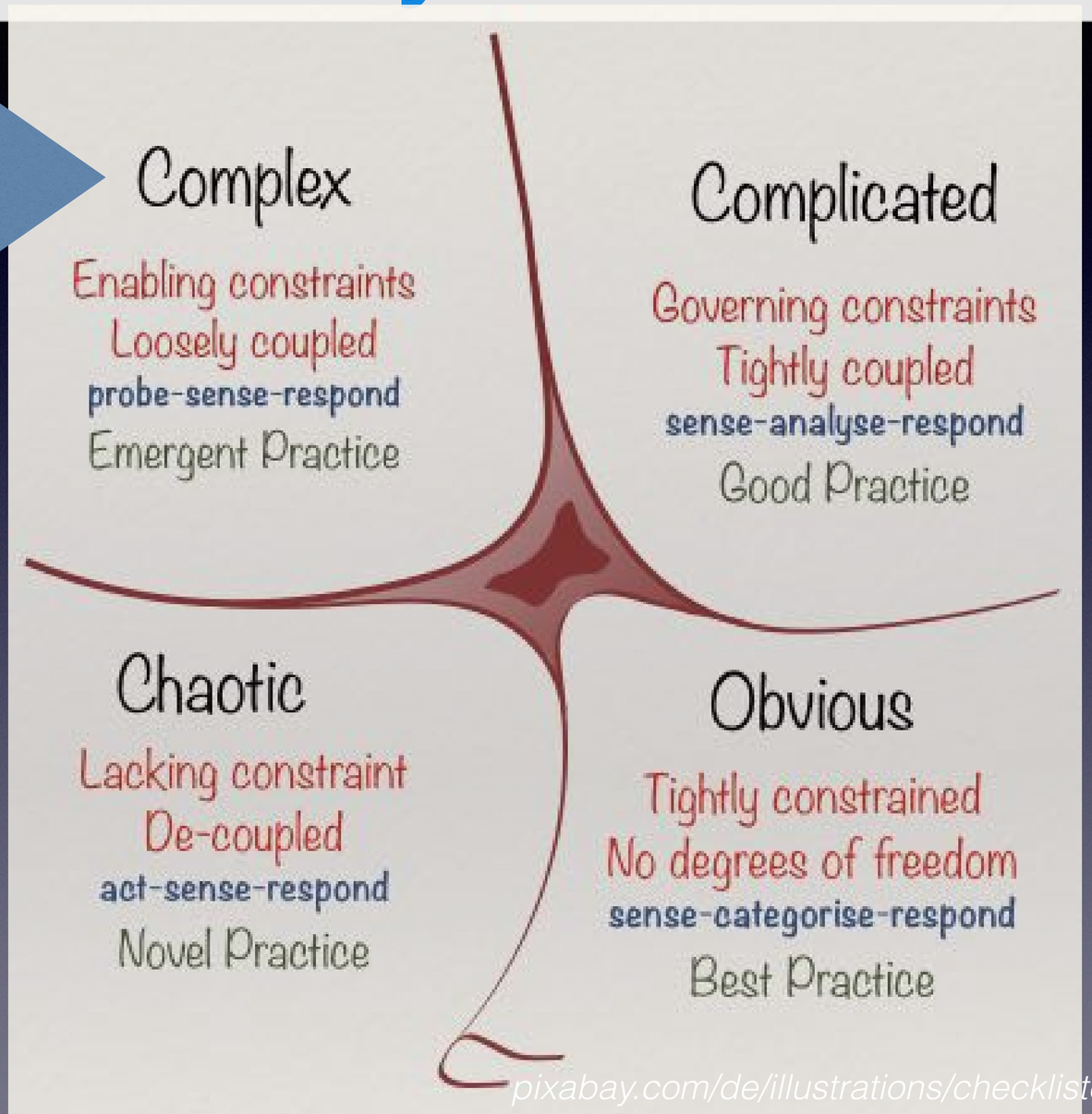
Wissen wir denn nicht, was wir bauen wollen?

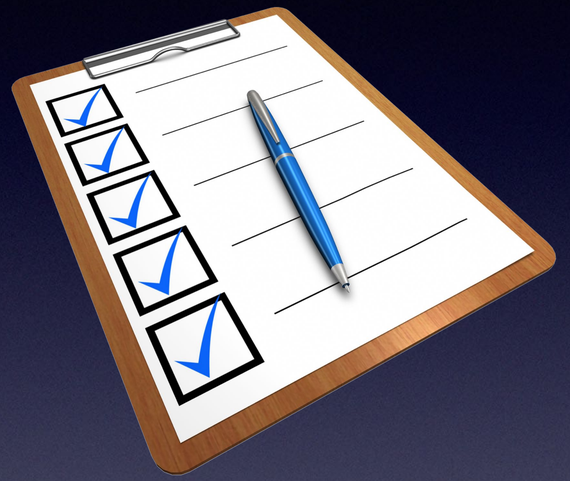
Wissen wir, was wir bauen *sollten?*

Cynefin

Wir befinden uns häufig hier!

Ursache / Wirkung erst im Nachhinein zu erkennen





The Cynefin Framework
by Dave Snowden.
CC BY-SA 3.0

pixabay.com/de/illustrations/checkliste-zwischenablage-fragebogen-1622517/

Experimente!

Auf Experimente vorbereitet sein:

80%-Lösungen
A/B-Testing

Hypothese & Experiment & Auswertung
(statt Anforderung & Umsetzung & Hoffen / Hotfixes)

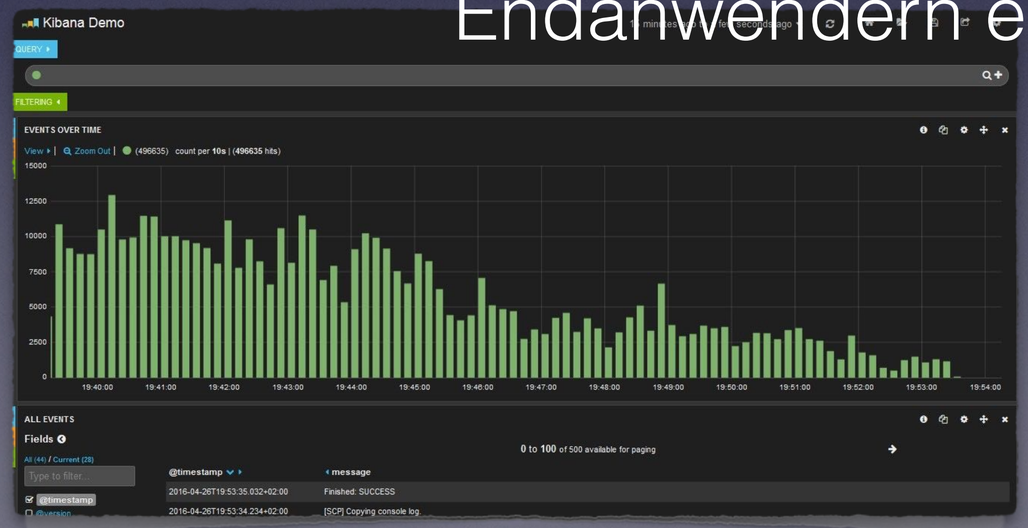
Beobachtbarkeit gibt Sicherheit.

Beobachtbarkeit

Laufzeit-Monitoring auch durchs Dev-Team
(technisch *und fachlich!*)

Real User Monitoring

Endanwendern einfaches Feedback ermöglichen



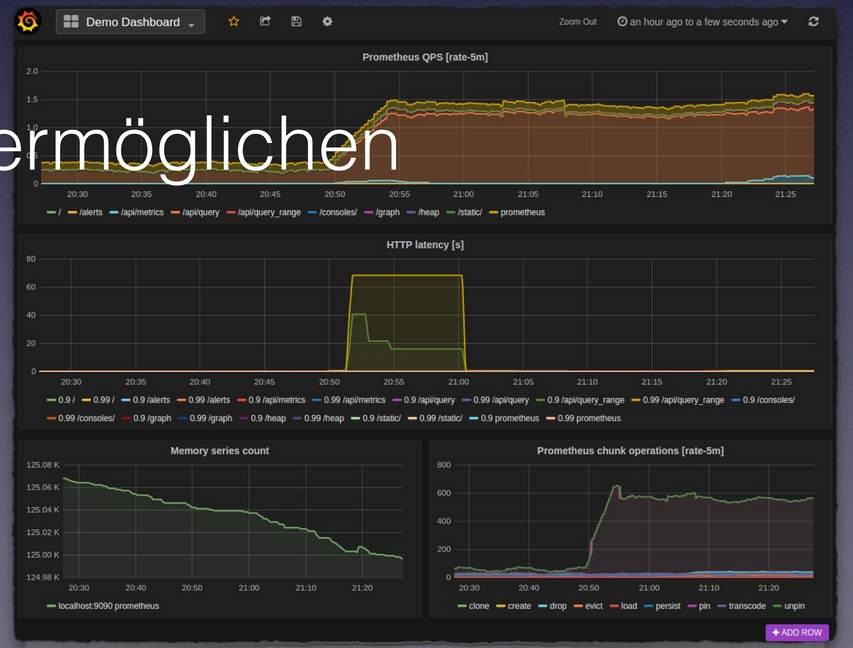
Klar strukturiert, sehr übersicht... 11. Mai
★★★★★ Schusseli

Klasse App, sehr übersichtlich und schnell.

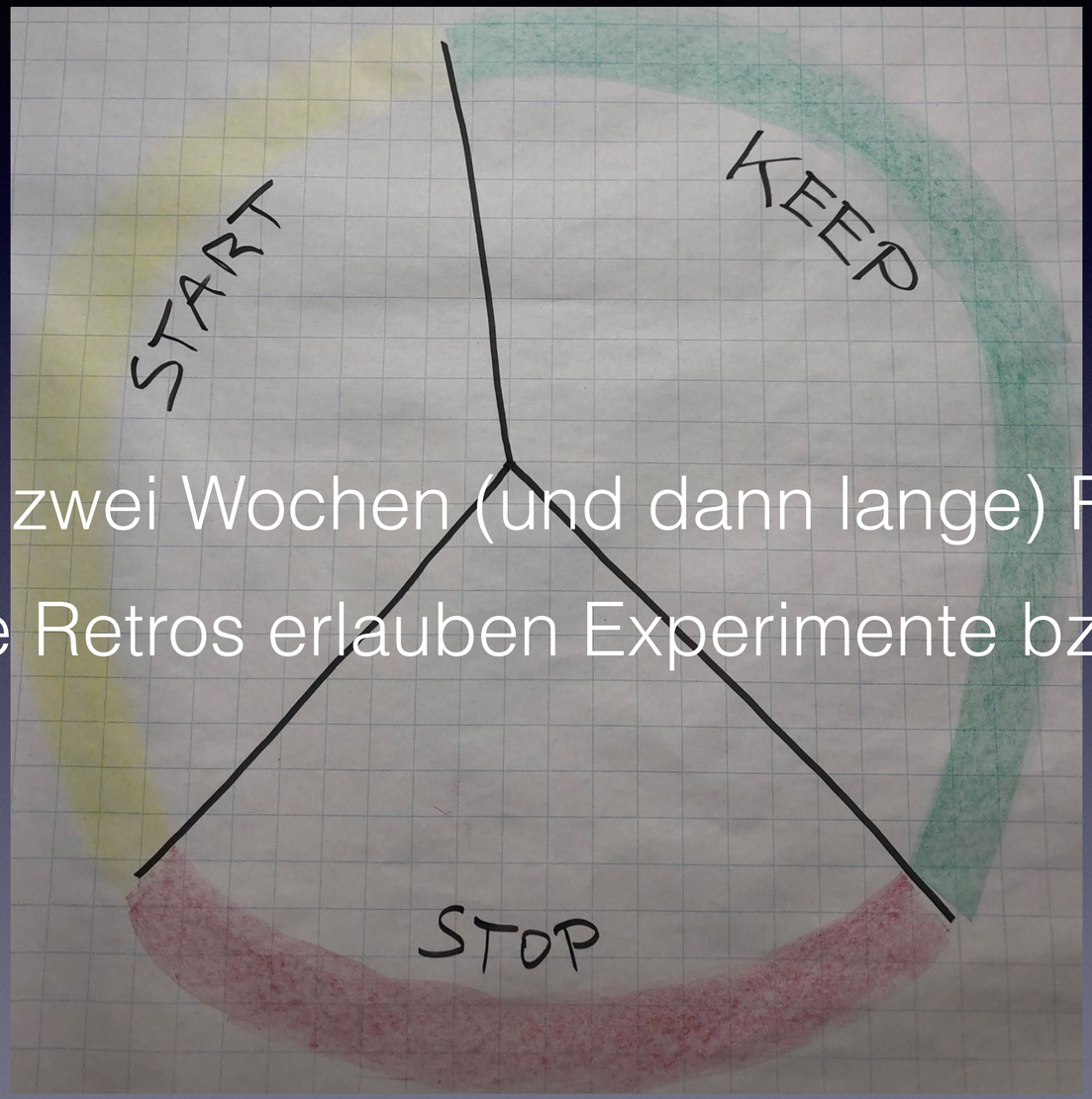
Tolle Shopping-App 27. Okt.
★★★★★ Miss Marple72

Übersichtlich aufgebaut und sehr gut zu bedienen, auch die Kundenbetreuung ist immer freundlich und gut zu erreichen, was will man mehr!

Preiswecker funktioniert nicht... Vor 1 J.
★☆☆☆☆ mndjdkvvdje

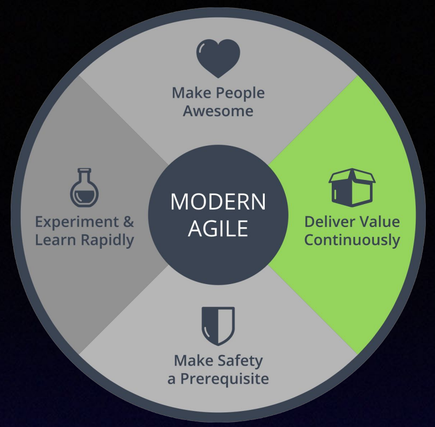


Schnelleres Lernen lernen



Warum nur alle zwei Wochen (und dann lange) Retrospektiven?

Häufigere, kürzere Retros erlauben Experimente bzgl. Arbeitsweisen.



Deliver Value Continuously

Was blockiert uns?

Technik

Einfaches, sicheres Deployment möglich?

Angst, Code zu ändern?

Prozess

Sprints?

Estimations?

DoR, DoD?

Fehlendes Wissen bzw. Können

Deliver Value Continuously

Automatisierung

Pipelines – Continuous Delivery – Continuous Deployment

Gemeinsamer Wissensaufbau

Gemeinsames Lernen schafft wertvolles Können

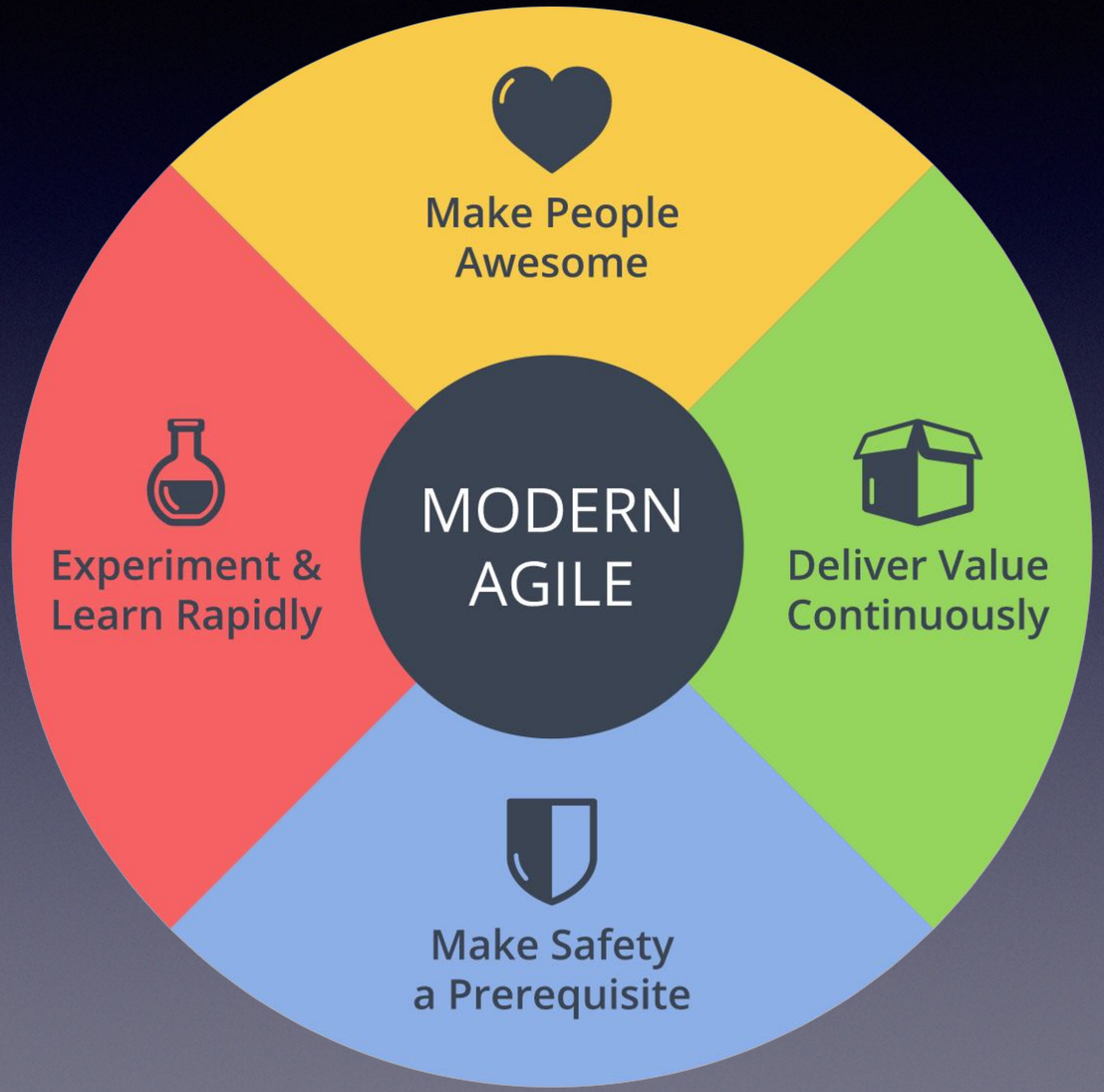
Pull statt Push

Gemeinsam Dinge fertig bekommen (statt viel parallel anfangen)

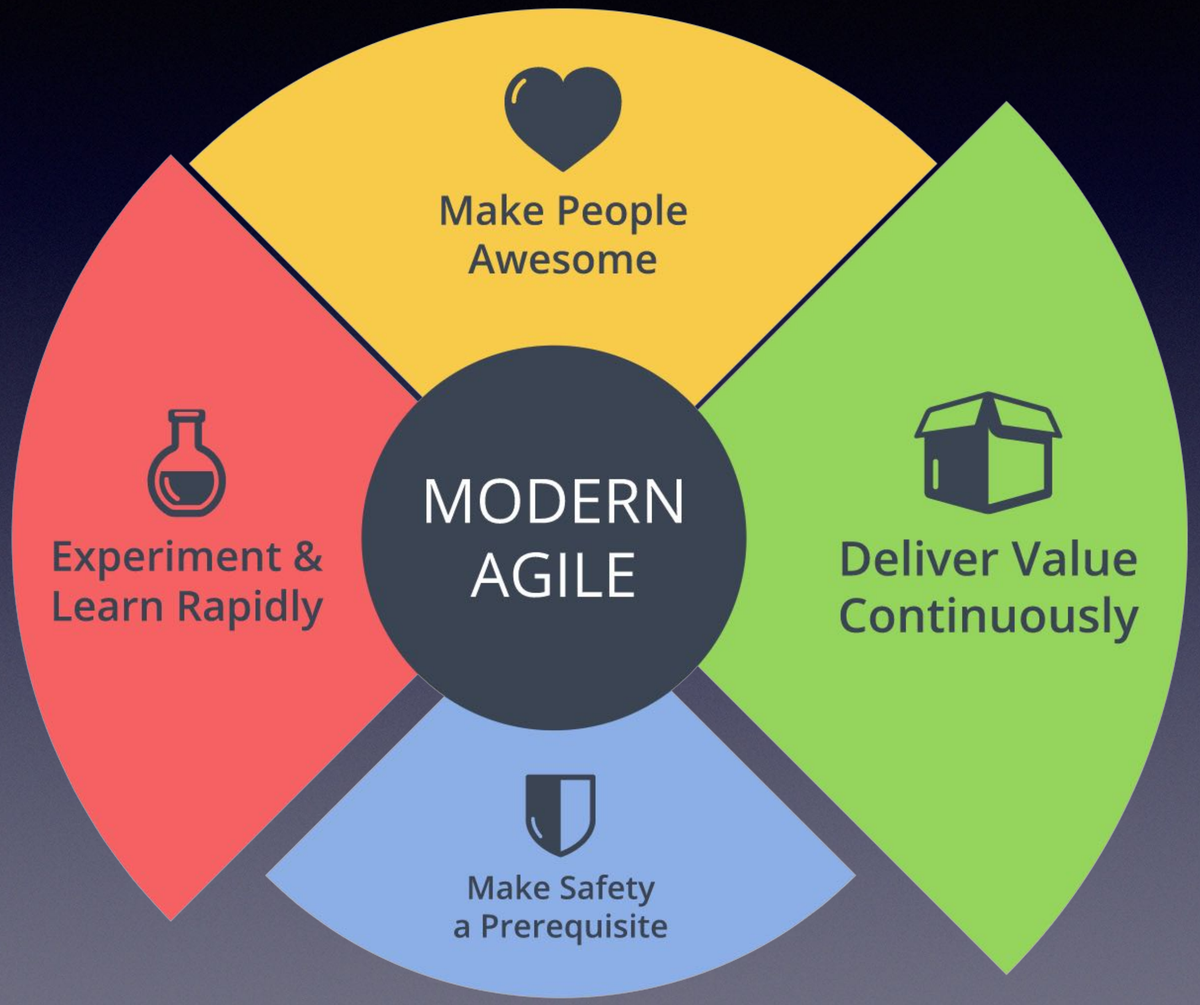
Flow

„Das machen wir doch alles schon…“

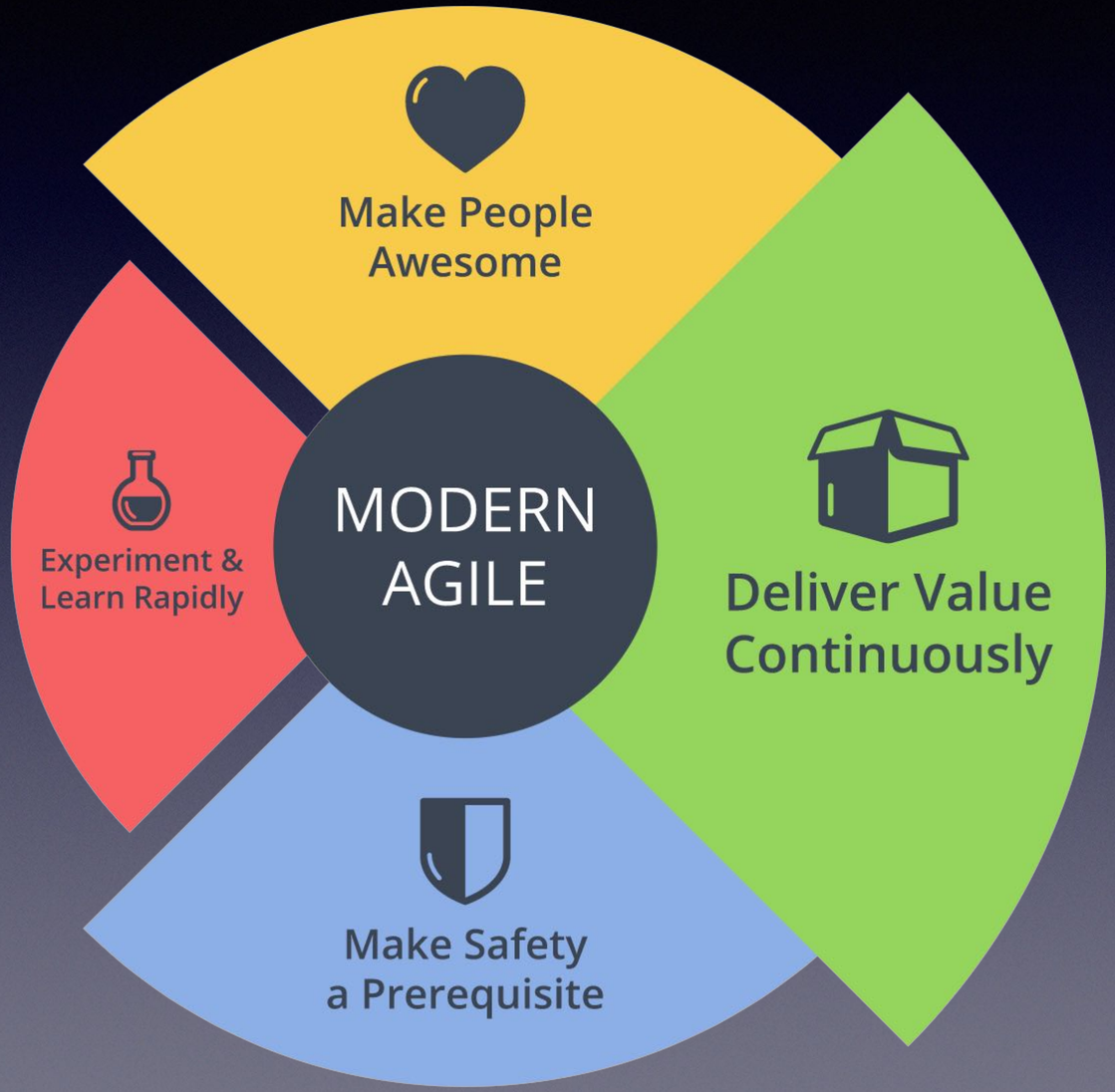
Läuft das Rad rund?



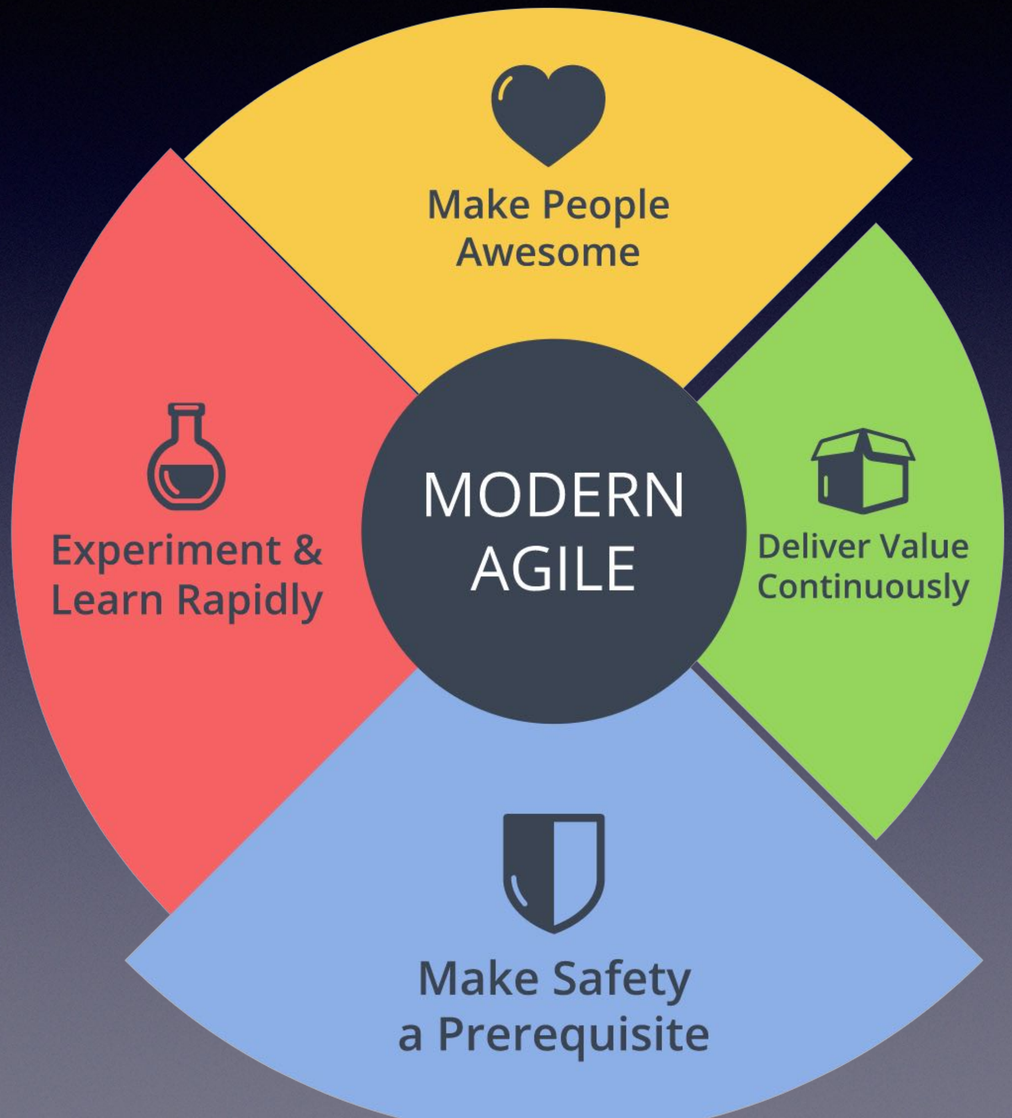
Läuft das Rad rund?



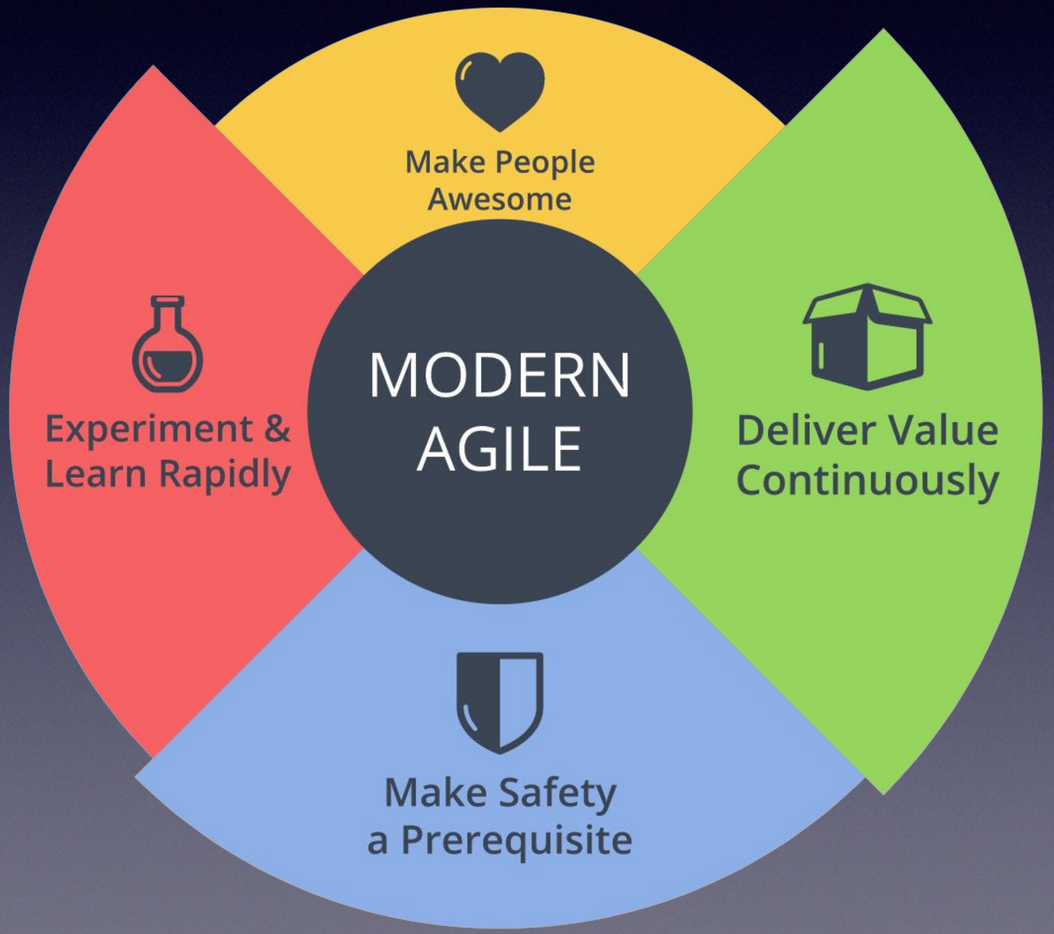
Läuft das Rad rund?



Läuft das Rad rund?



Läuft das Rad rund?



Sensor zum Kalibrieren

Agile und Modern Agile sind Sensoren, Messinstrumente.

Helfen, das eigene Vorgehen einzuschätzen und zu kalibrieren.

Machen vorhandene Probleme offensichtlich.

Don’t shoot the messenger!

Alles altbekannt?

Ja, genau!

Moderne agile Praktiken?

Keine festgeschrieben!

Werte/Prinzipien bleiben, Praktiken verändern sich.

Derzeit hilfreiche Praktiken:

Continuous Deployments

#MobProgramming

#NoEstimates

#NoProjects

Modern Agile statt Agile?

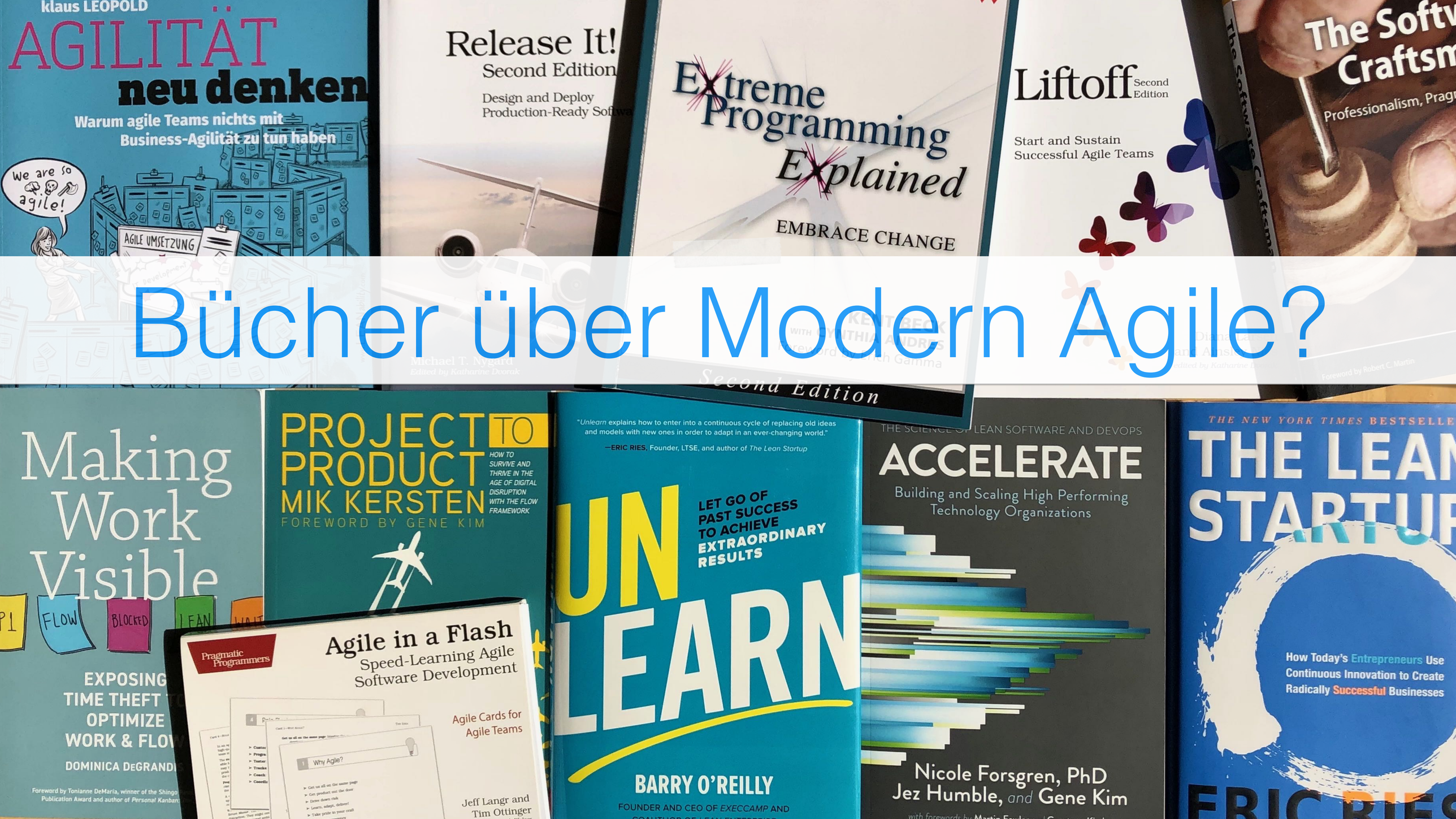
Modern loslegen oder lieber traditionell (z.B. Scrum)?

Sind 2-Wochen-Iterationen eine Verbesserung?

Auch verglichen mit Wettbewerbern?

Kunden-Erwartungen?

pixabay.com/de/photos/ansager-audio-neon-kassette-316585/
pixabay.com/de/photos/telefon-alt-1955-telefonh%C3%B6rer-2524268/
pixabay.com/de/photos/smartphone-bildschirm-horizontale-1957742/
pixabay.com/de/photos/musik-auf-ihrem-smartphone-spotify-1796117/



Bücher über Modern Agile?

Lernen & mitmachen!

Learn More

THE #MODERNAGILE SHOW

In addition to the many great article, video and book links below, you can learn a lot about Modern Agile by watching or listening to The #ModernAgileShow.

Watch on Youtube

Audio-only Podcast

An Introduction to Modern Agile

Make People Awesome | Make Safety a Prerequisite | Experiment & Learn Rapidly | Deliver Value Continuously

Höchstleistung durch Psychologische (emotionale) Sicherheit

Erstellt von @HeidiHelfand and @JoshuaKerievsky –
Übersetzt von Reiner Ritter und Holger Dierssen – Version 0.5

Psychologische Sicherheit existiert, w
keine Angst hast du selbst zu sein, Pro
benennen, Risiken zu nehmen, Frager
und nicht mit andern Einverstanden z
- Joshua Kerievsky & Heidi Helfand

Niemand kann seine beste Leistung ei
bis er sich sicher fühlt. Vertreibe die A
müssen angstfrei sein, um Ideen ausz
und Fragen zu stellen.
- W. Edwards Deming, Out Of The Cri

Führungskräfte **fördern Teamsicherhe**
- Deutung von Arbeit als Lernproble
nicht als Ausführungsproblem.
- Zugeben ihrer eigenen Fehlbarkeit
zeigen von Verletzbarkeit vor dem
- Neugierig sein und Fragen stellen.
- Ermutigen verschiedene Standpun
offenzulegen.

- Angepasst von Amy Edmondson und

Chartering Cheat Sheet

Created by Joshua Kerievsky & Matthew
Version 1.0

Chartering is a continual practice of defining and aligning on desired outcomes, the community needed to achieve those outcomes and the agreements by which people in the community will collaborate harmoniously.
-- Joshua Kerievsky

The Charter is the official container for the community's shared understanding and agreements. *--Joshua Kerievsky & III*

Evolutionary Design Cheat Sheet

Created by Joshua Kerievsky, Ashley Johnson, Mike Rieser, Bill Wake, Tim Ottinger
Version 1.0

MODERN AGILE

"A seed is an embryonic plant." *–Wikipedia*

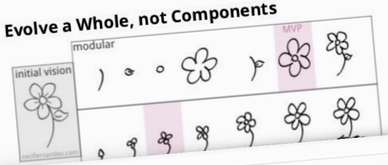
Evolutionary Design Is...
A strategy for rapidly providing value by growing something from a primitive, yet complete, whole to a higher level of sophistication over time.
–Joshua Kerievsky

Think Primitive and Essential

"A complex system that works is inevitably found to have evolved from a simple system that works." *–John Gall*

What Is a Primitive Whole?
- "Primitive" - a "rough cut" version, lacking details and precision.
- "Whole" - an integration of all major parts.
- A valuable and useful starting point.

Evolve a Whole, not Components



What Do You Evolve First?
Pursue high-value and high-risk items early. Hi
value is the essence of what the end solution
does. High risk is anything new or untested tha
you'd do well to learn about sooner by using it
early evolutions.

Accelerating Flow Cheat Sheet

Created by Ashley Johnson, Mike Rieser & Joshua Kerievsky
Version 1.0.1

Lean Thinking
1. *Identify Value* - Specify value from the standpoint of the end customer.
2. *Map the Value Stream* - Identify steps in the value stream, removing all unnecessary steps.
3. *Create Flow* - Make value-creating steps occur in tight sequence so product flows smoothly to customers.
4. *Establish Pull* - As flow is introduced, let customers pull value from the next upstream activity.
5. *Seek Perfection* - As value is specified, value streams are identified, wasted steps are removed, and flow & pull are introduced, begin the process again and continue until a state of perfection is reached in which perfect value is created with no waste.

- *Type 2 Muda* - This is "pure waste" or things that could be eliminated immediately without damaging our ability to deliver value.
- *Type 1 Muda* - This is "Supposedly Necessary Waste" and refers to things which do not directly create value, but eliminating them all at once would damage our ability to deliver value. An example of this might be a validation process which doesn't actually create value, but omitting it could be catastrophic while at our current level of producing defects.

Kanban Boards

How to Sabotage Flow
"The hard part about developing eyes for waste is that most waste is caused by doing things right within the conventional system."
--Allen Ward
Easy ways to sabotage flow include:
- *Allow High WIP.* WIP is a leading indicator of

Mob Programming Cheat Sheet

Created by Coaches and Developers @IndustrialLogic
Version 0.1

What is Mob Programming?
"All the brilliant people working on the same thing, at the same time, in the same space, on the same computer." *- Woody Zuill*

"Mob Programming is continuous integration for ideas." *- Joshua Kerievsky*

There are many ways to successfully mob. In general, there is one computer, a keyboard and mouse, one or more monitors, a whiteboard, one Driver and one or more Navigators sharing a work environment like the one below:

Image by Mark Pearl

Ideal Mob Size The whole team, although teams of 3-6 people is ideal. Beyond 6 people, a Mob may have difficulties keeping everyone engaged. You can counter this with quicker rotation times.

Driver's Role - The Driver operates the computer to input/implement ideas made by the Navigator(s).

Navigator(s) Role - Navigator
Driver. Usually it works to allow
Mob to interact directly with th
causes too much chaos, have
give directions to the Driver a
serves as the voice of the Mo
mobbing can ask for help on

Driver Dos & Don'ts
- Drivers don't navigate. If t
only one who knows wha
relinquish their role as D
person in the rotation.
- Each Driver can share h
directed, including askin
intent, location and deta
navigators must commu
allows the Driver to un
action.
- If no one is navigating,
typing.

Navigator(s) Dos & Don
- Navigator ideas can o
going through the Dri
- Navigators must pay
skill level. If the Drive
instructions, the Mob
what the Driver nee
more advanced, Nav
instructions, like "co
method to the pare

Agile

Customer collaboration
over contract negotiation

Responding to change
over following a plan

Working software
over comprehensive documentation

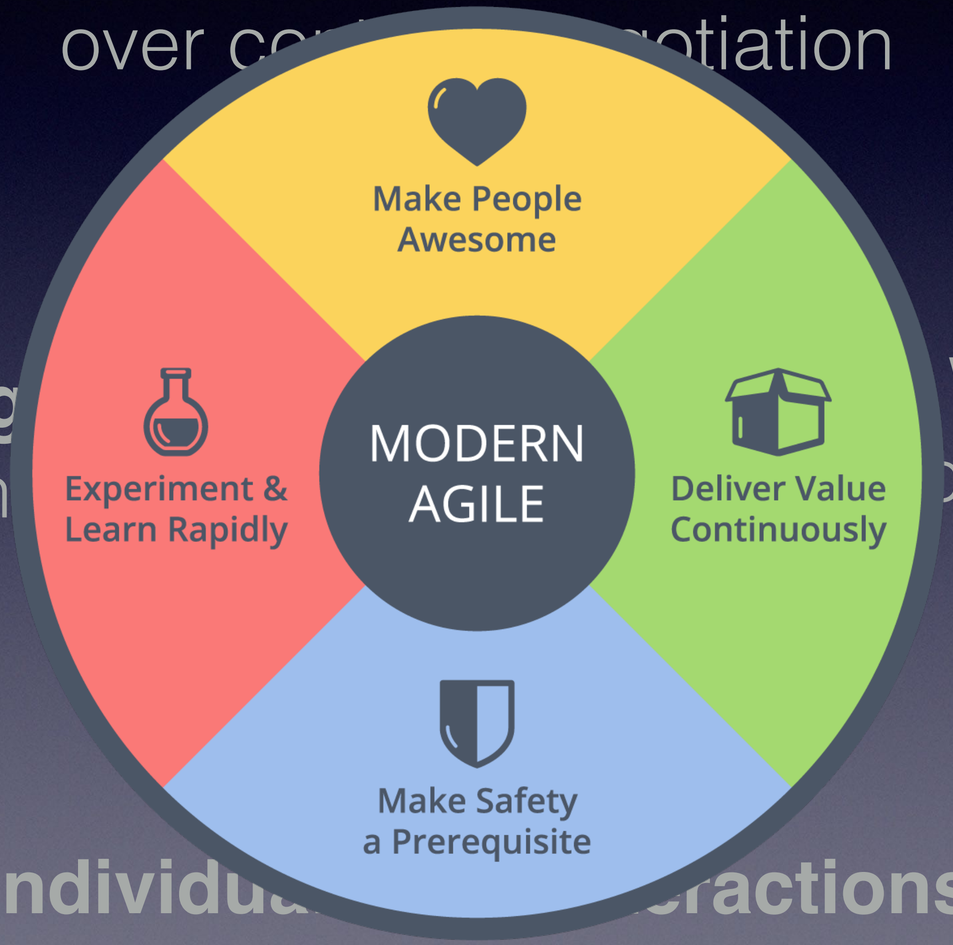
Individuals and interactions
over processes and tools

Modern Agile

Customer collaboration
over co[illegible]otiation

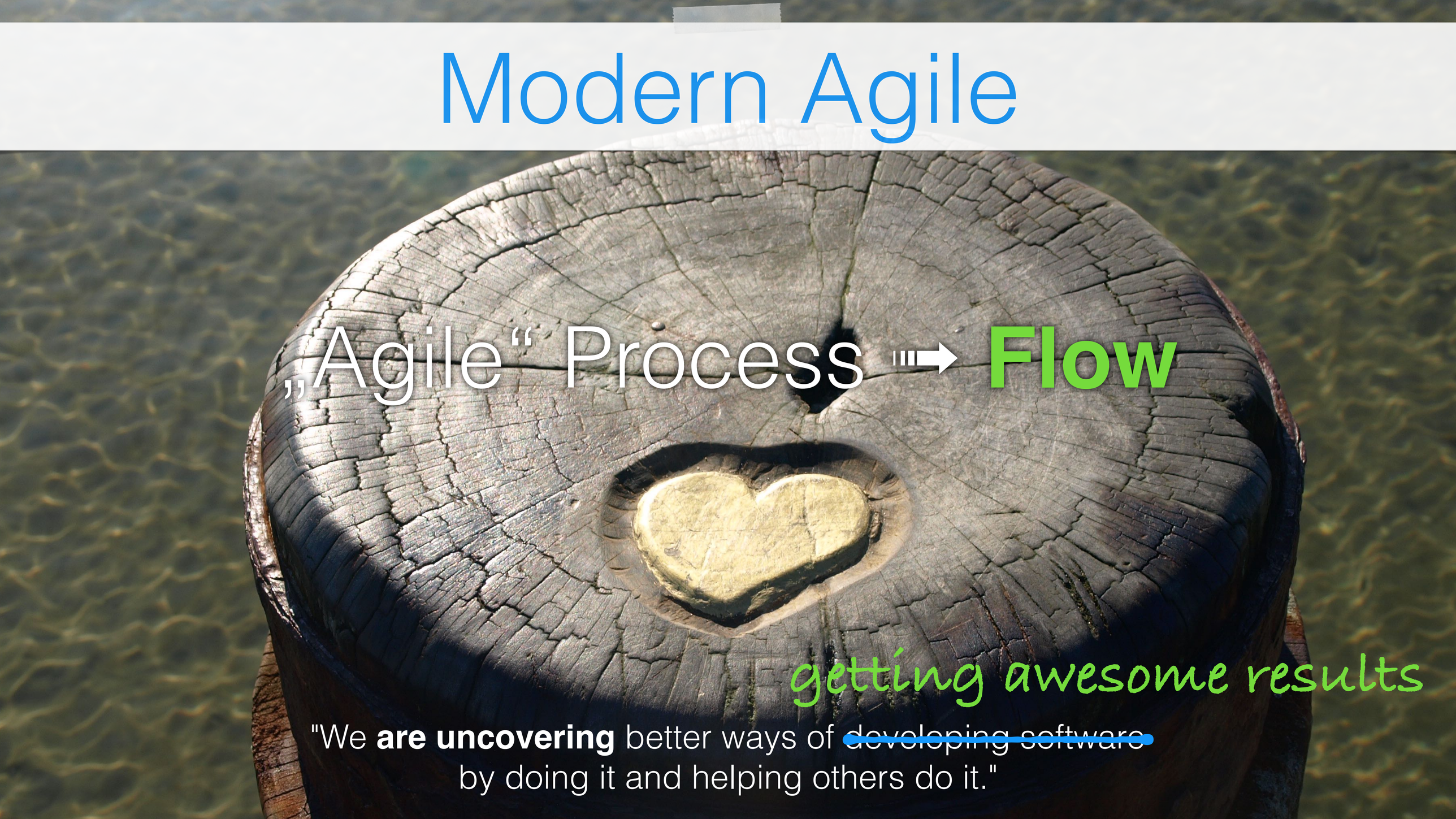
Responding to chang
over following a plan

Working software
over comprehensive documentation



Individua[illegible]eractions
over processes and tools

Modern Agile

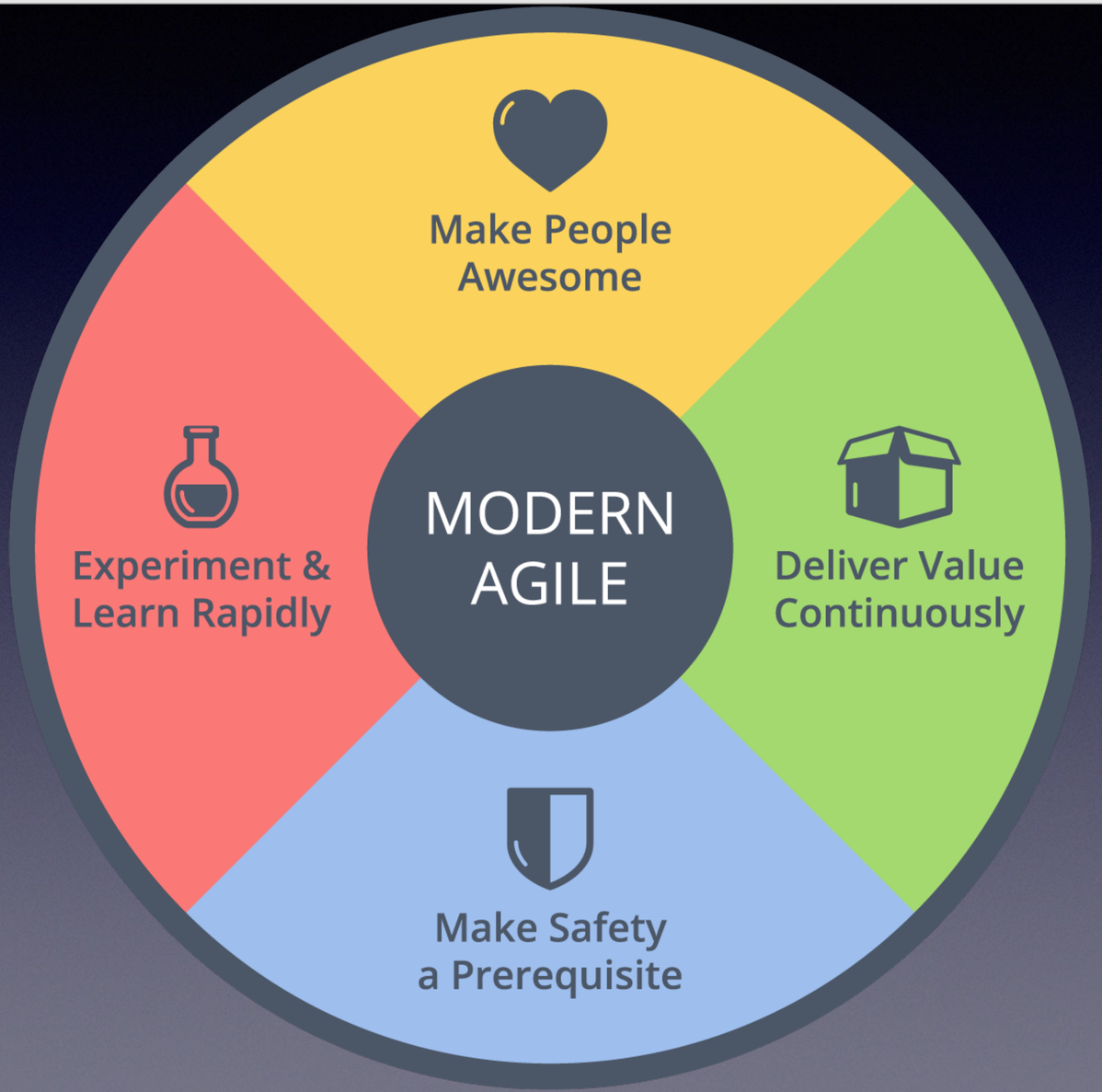


„Agile“ Process ➠ **Flow**

getting awesome results

"We **are uncovering** better ways of ~~developing software~~
by doing it and helping others do it."

Fragen? Fragen!

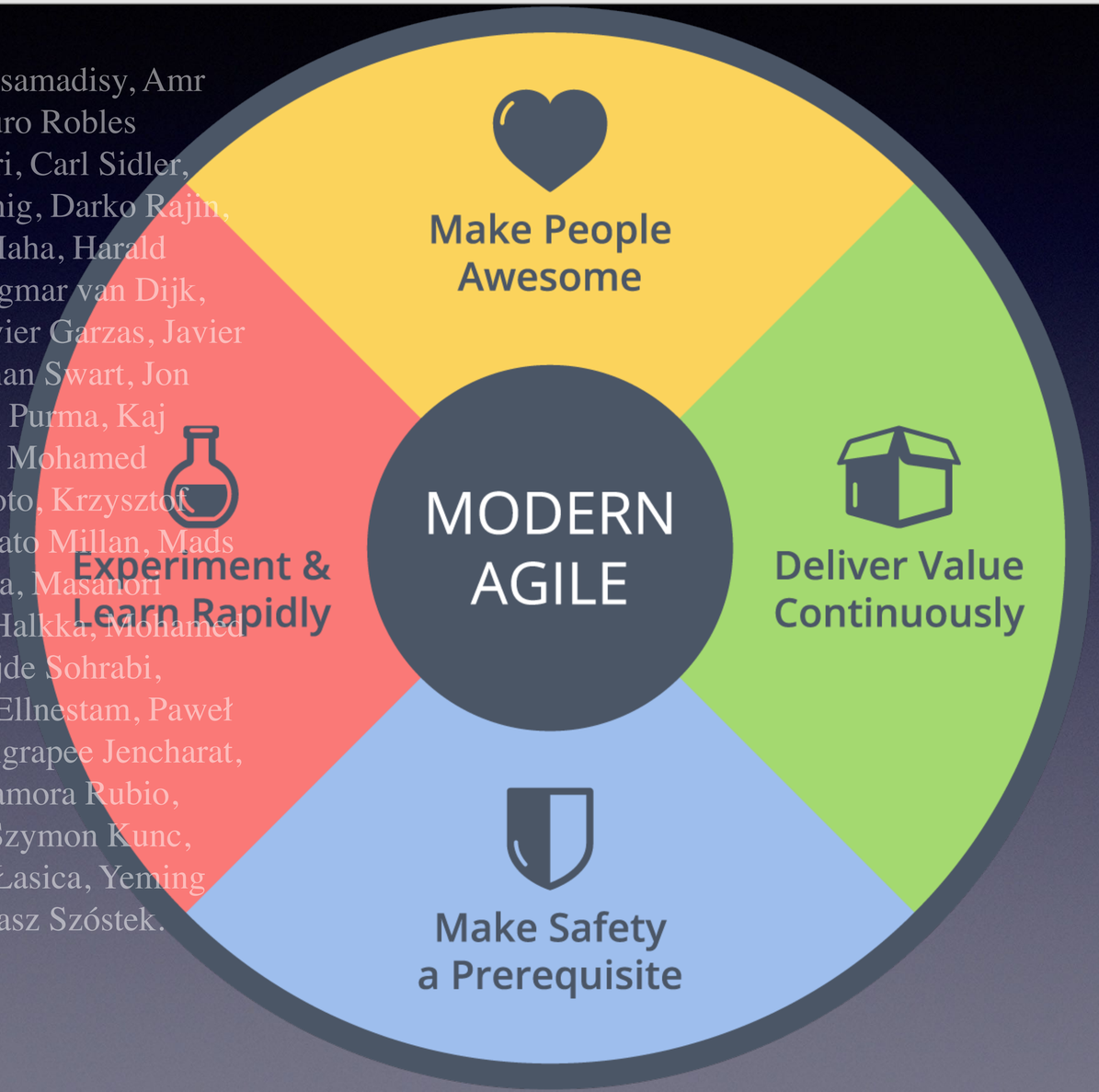


code.talks

Thomas Much
@thmuch

Vielen Dank!

Alexandre Freire, Alexey Krivitsky, Amr Elssamadisy, Amr Noaman, Anders Breivik, Anne Landro, Arturo Robles Maloof, Arunthep Sangvareethip, Asad Safari, Carl Sidler, Carmen Diaz Guadarrama, Cristiano Schwenig, Darko Rajin, Dejana Šćuric, Di Cao, Evelyn Tian, Frieh Maha, Harald Koebler, Helmut Pienaar, Hossein Ansari, Ingmar van Dijk, Jahan Zinedine, Janice Qian, Jason Qian, Javier Garzas, Javier Tenorio Martínez, Jenjira Sangvareethip, Johan Swart, Jon Brownstein, Mario, Joshua Kerievsky, Jukka Purma, Kaj Mustikkamäki, Kamil Berdychowski, Karim Mohamed Elsayed, Karin-Lerich Deyzel, Karoliina Luoto, Krzysztof Jelski, Lucas Duarte Silveira, Luis Raúl Mulato Millan, Mads Opheim, Marek Włodkowski, Marta Smyrska, Masanori Kado, Melissa Pienaar, Miguel Peres, Miili Halkka, Mohamed Ragab, Mohsen Ghafoori(@EmJiHash), Mojde Sohrabi, Molood N. Alavijeh, Nenad Maljković, Ola Ellnestam, Paweł Polewicz, Paweł Wehr, Phil Rautenberg, Pongrapee Jencharat, Rafael Rodrigues, Ruud Wijnands, Sergio Zamora Rubio, Shayan Salehian(@shayan72), Steve Shen, Szymon Kunc, Tanat Kitcharoen, Tomasz Fortuna, Tomasz Łasica, Yeming Yang, Zuzi Sochova, Łukasz Krupa and Łukasz Szóstek.



http://modernagile.org/#mediaKit

code.talks

Thomas Much
@thmuch