

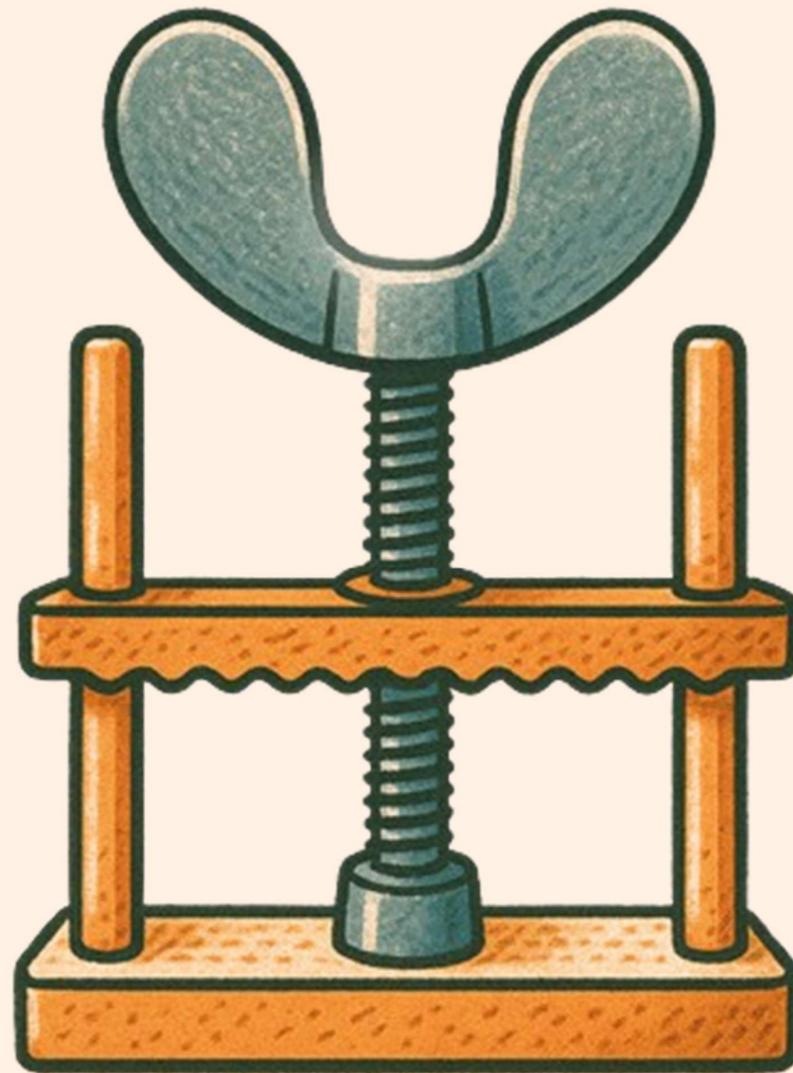
Digital Crafts Day 2025, 4. April 2025, Weiden

Architektur-Governance: Daumenschrauben für Softwareentwickelnde

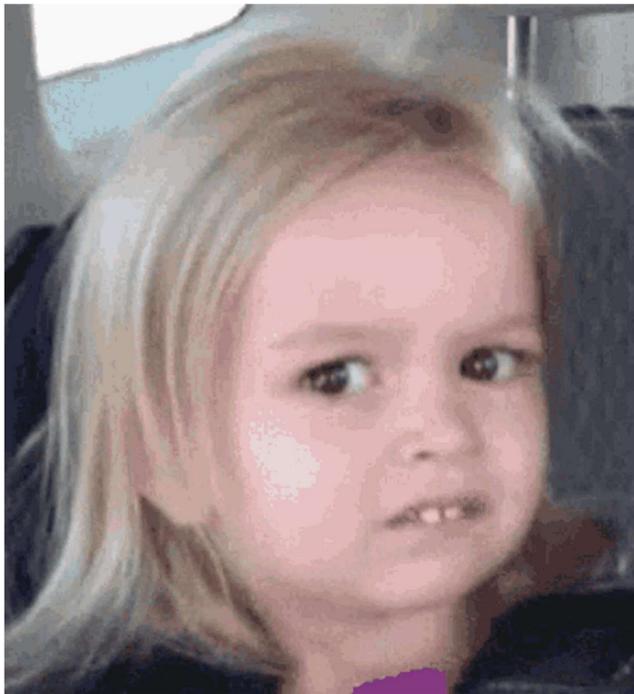
Markus Harrer

Software Evolutionist

INNOQ



Architektur- Governance?



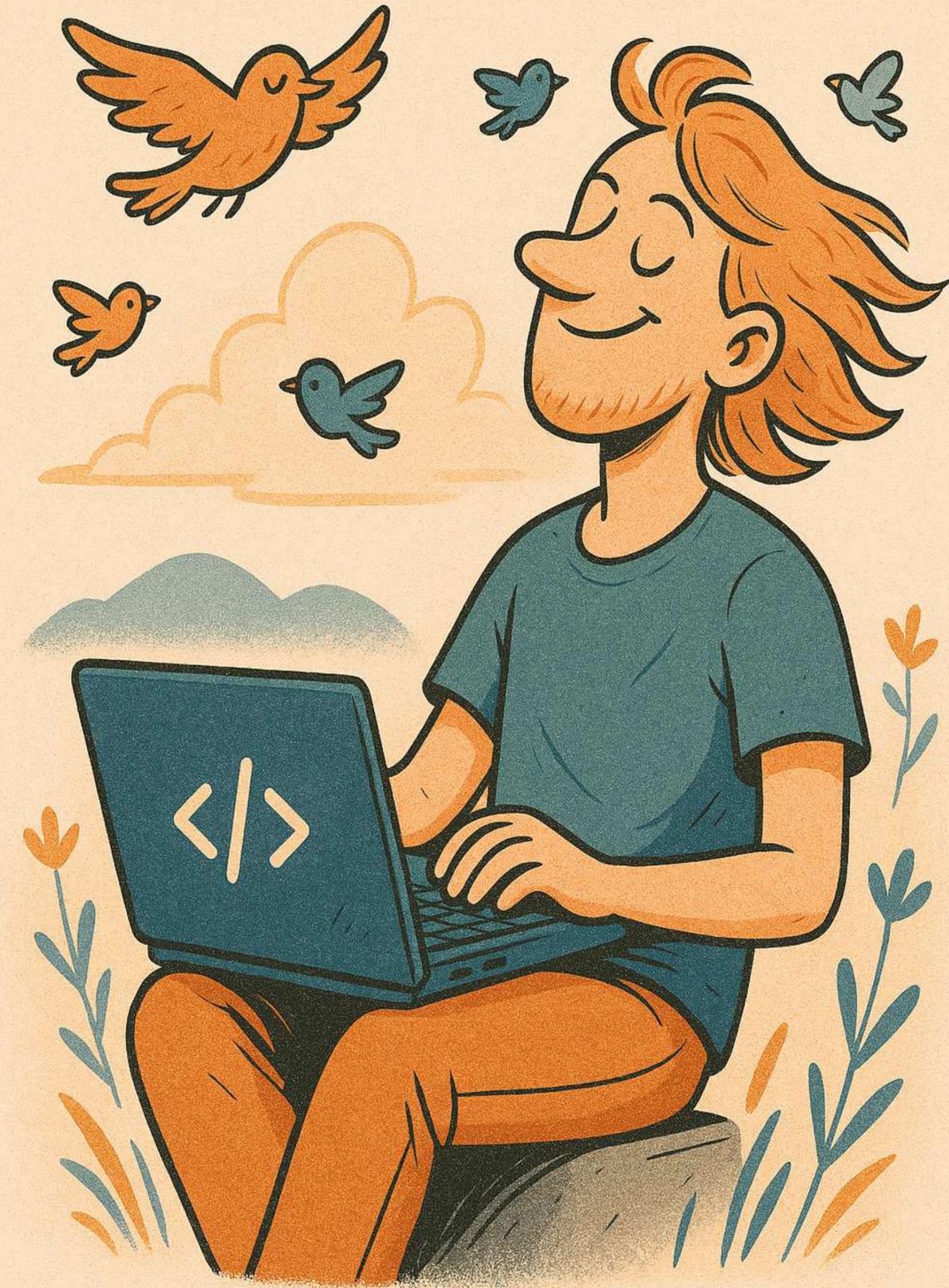
Regeln?
Prozesse?
Strukturen?
Rollen?

Rahmen aus Regeln,
Prozessen, Strukturen
und Rollen, der dafür
sorgt, dass Software-
Architekturentschei-
dungen konsistent,
nachvollziehbar und
strategiekonform
getroffen und umgesetzt
werden.

Regeln? WTF!

Als Entwickler:in möchte ich doch kreativ, frei und selbstbestimmt arbeiten

WTF: Welch' törichte Frage



Hallo, agil?

Manifest für Agile Softwareentwicklung

Wir erschließen bessere Wege, Software zu entwickeln,
indem wir es selbst tun und anderen dabei helfen.
Durch diese Tätigkeit haben wir diese Werte zu schätzen gelernt:

Individuen und Interaktionen **mehr als** Prozesse und Werkzeuge
Funktionierende Software **mehr als** umfassende Dokumentation
Zusammenarbeit mit dem Kunden **mehr als** Vertragsverhandlung
Reagieren auf Veränderung **mehr als** das Befolgen eines Plans

Das heißt, obwohl wir die **Werte auf der rechten Seite wichtig finden,**
schätzen wir die Werte auf der linken Seite höher ein.

Regel A

**Ach ja, noch
ne Regel**

Regel B

IT

= Department Of No

naysayers →

Information Security Officer

Enterprise Architects

Softwarearchitekten

Regel dies

Regel das

Regel bla

Authentifizierung
Continuous Delivery **Gitflow**
Monitoring **Domain Driven Design**
Code Reviews **API-Dokumentation**
Microservices **Alerting**
AI **Code-Doku**
Input Validation
Commit Messages **RESTful APIs**
Logging **Datenverschlüsselung**
Performance-Tests



Architektur-Governance ist wichtig

Wir hatten eine State of the Art Cloud Native App geschrieben, konnten sie aber dann nicht produktiv nehmen, da der Produktivbetrieb in der Cloud von der IT untersagt war.

Wir haben uns Kafka auf AWS geklickt, aber die Admins wollten uns dann keine Firewall-Freischaltung dafür geben.

Wir können jetzt innerhalb von 30 Minuten auf Prod deployen, aber die Release-Manager tagen nur alle 6 Monate.

damit sowas nicht passiert ...

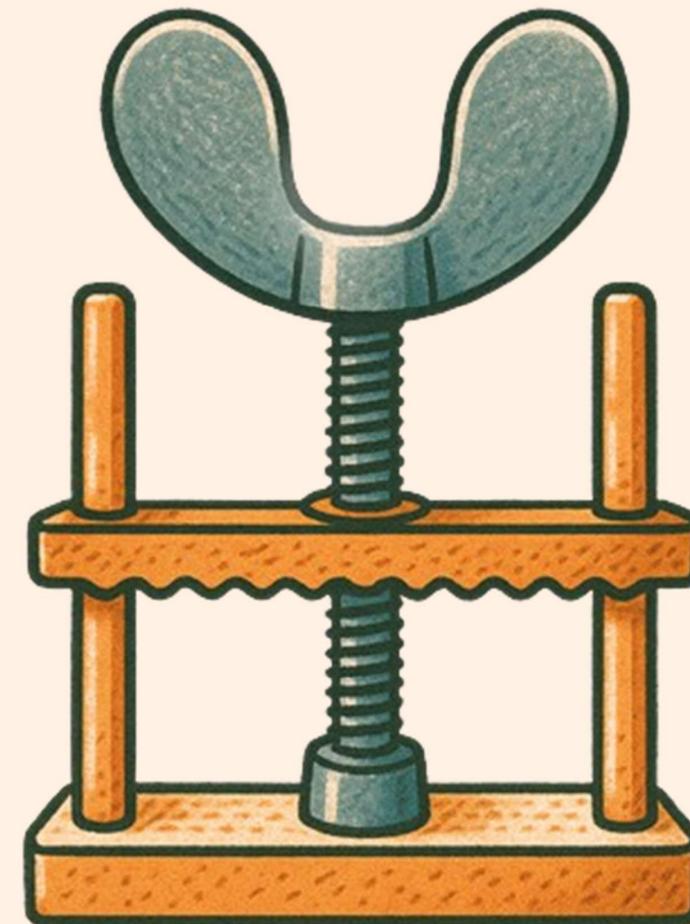
Wo dürfen wir regieren?



Ziel dieses Vortrags

Das richtige Drehmoment für die Daumenschrauben finden

„Geregeltes Quälen“



Spoiler: kann weh tun

Mehr Infos: <https://wissen.schloesserland-sachsen.de/blog/detail/sprichwoertlich-die-daumenschrauben-anlegen/>

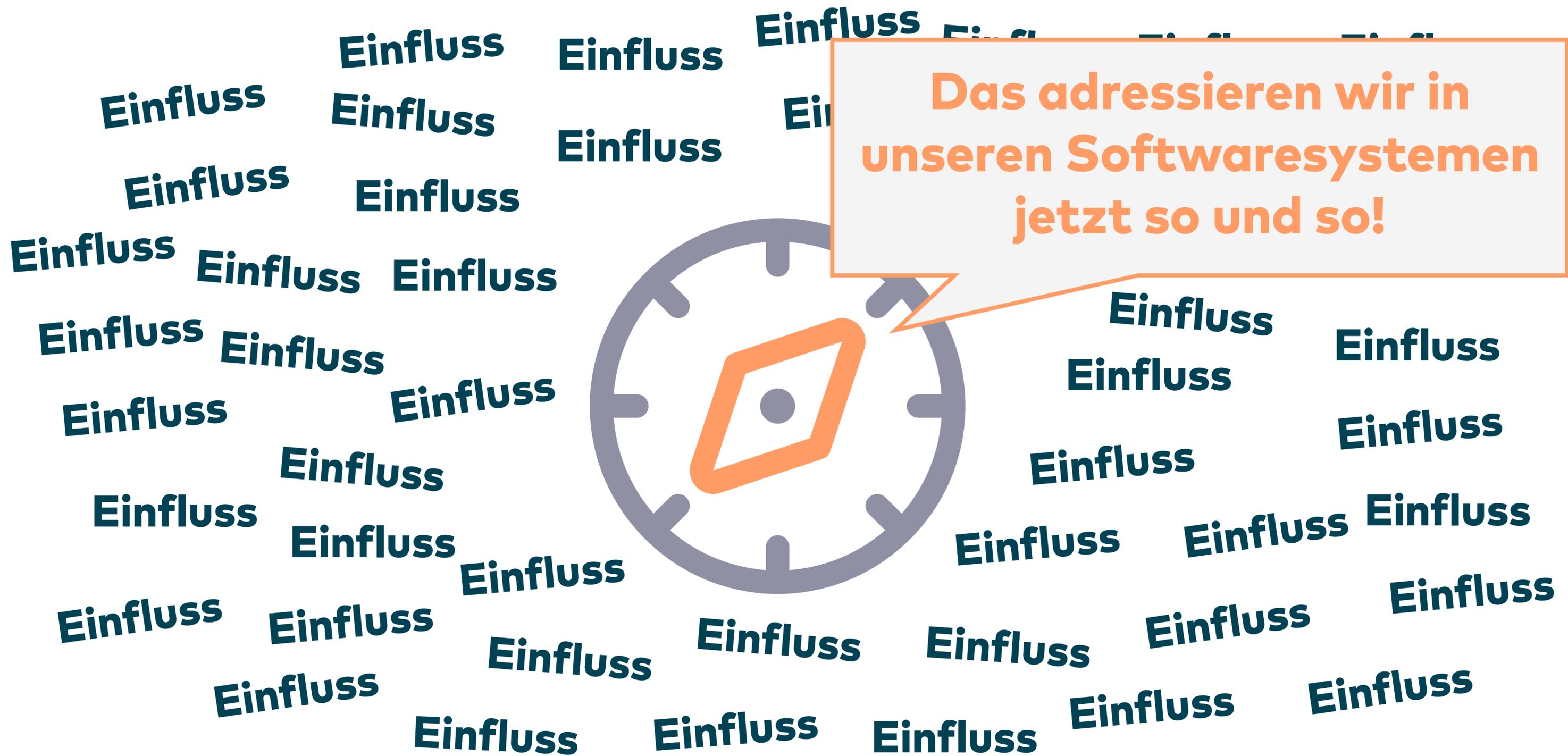
**Woher kommen denn
diese ganzen Regeln?**

Mögliche Einflüsse von außen

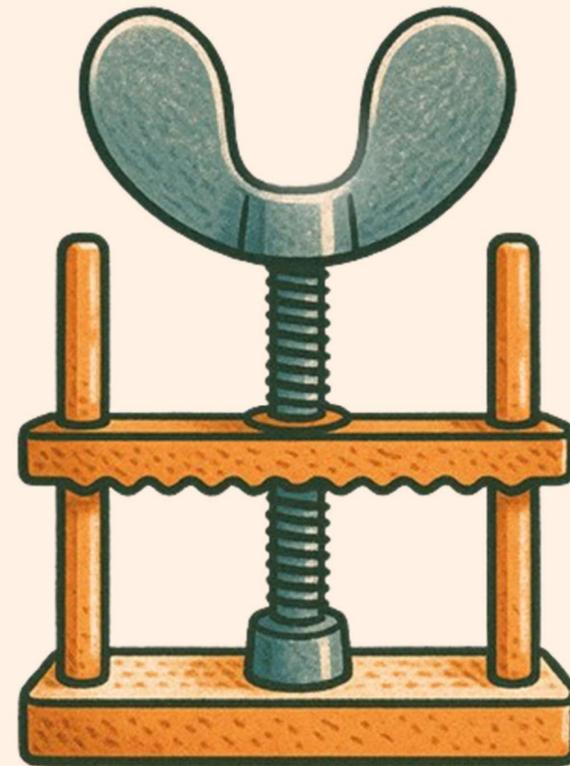
- Gesetzgebung
- Normen / Standards
- Regulatorische Anforderungen (z. B. DSGVO, BaFin, HIPAA)
- Sicherheitsrichtlinien (z. B. ISO/IEC 27001, OWASP, BSI)
- Branchenspezifische Vorschriften (z. B. BAIT/VAIT, Medizinproduktegesetz, Luftfahrtstandards)
- Industrie-Konsortien und -Allianzen (z. B. W3C, GAIA-X)
- ...

Mögliche Einflüsse von innen

- Wachstumspläne (z. B. Internationalisierung, neue Märkte)
- Digitalisierungsstrategie / Cloud Transformation
- Erfahrung und Skills im Team
- Verfügbare Kapazitäten (FTEs, Zeit, Geld, ...)
- Erwartungen von Fachabteilungen, Controlling, Legal, ...
- Vorhandene andere Systeme, Plattformen und Infrastruktur
- ...



Was bedeutet das für uns?



Wir müssen uns damit leider auseinandersetzen!

Stile der Architektur-Governance

bestimmend



begleitend

Verschiedene Auslegungen

"to **rule** without sovereign power and usually **without having** the **authority** to determine basic policy"

"to **manipulate**"

"to **control**"

"to **direct**, or **strongly influence** the **actions** and conduct of"

"to **serve** as a precedent or **deciding principle** for"

bestimmend



begleitend

Verschiedene Motivatoren

**Extrinsisch
motiviert**



Ehre



Ruhm



Geld



Mastery



Purpose



Autonomy

**Intrinsisch
motiviert**

bestimmend



begleitend

Strafen

Verfahren

Verschiedene Umsetzungen

Feedback

Praktiken

Prinzipien

Regeln

Vorgaben

Werte

**hart &
detailliert**

**Weich &
offen**

Stile der Architektur-Governance

bestimmend

Tayloristisch

Zerlegung von Arbeit in kleinste, standardisierte Einzelschritte

Trennung von Denken (Planung) und Handeln (Ausführung)

Vorgaben und Kontrolle durch Management

Ziel: **Maximale Effizienz und Produktivität**



begleitend

Agile / Lean

Ganzheitliche Aufgaben im Team, Verantwortung über Features oder User Stories

Integrierte **Verantwortung** (Team denkt und handelt selbst)

Selbstorganisation des Teams, über Prinzipien, Ziele, Vertrauen

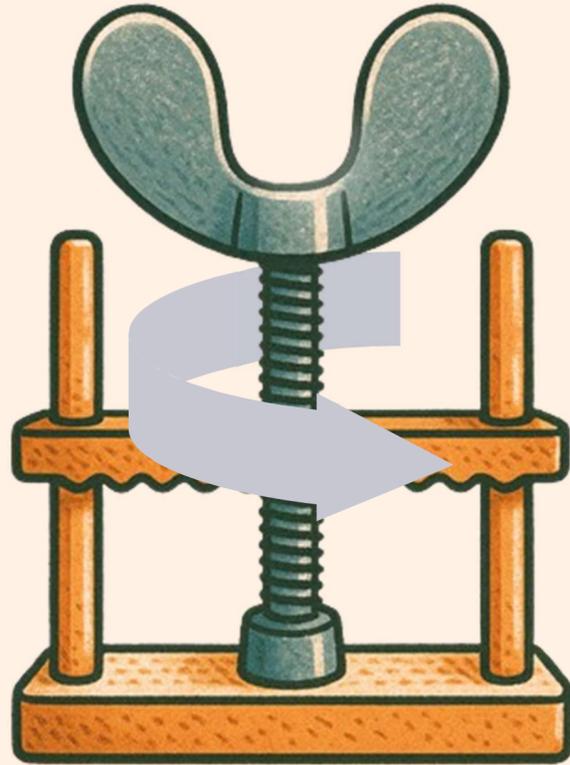


Dreyfus-Modell / Kompetenzstufen

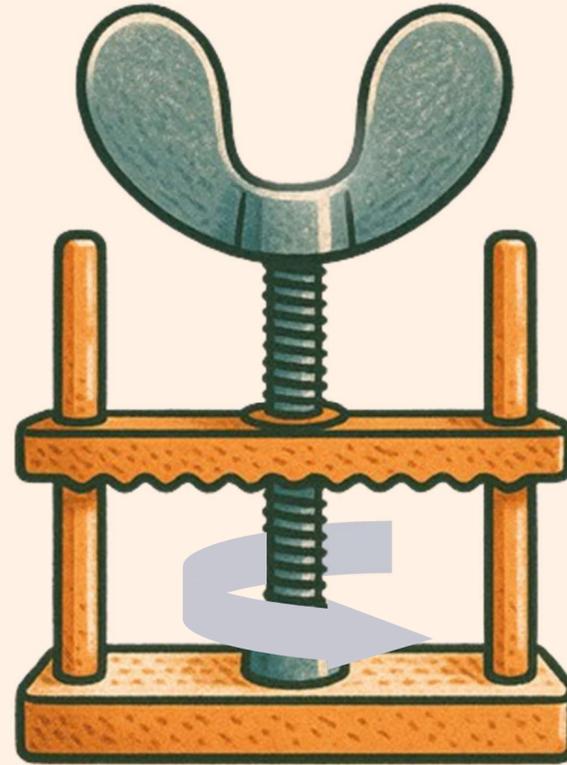


Angelehnt an <https://www.brainbok.com/guide/pm-fundamentals/interpersonal-and-team-skills/dreyfus-model-of-skill-acquisition/>

Was bedeutet das für uns?

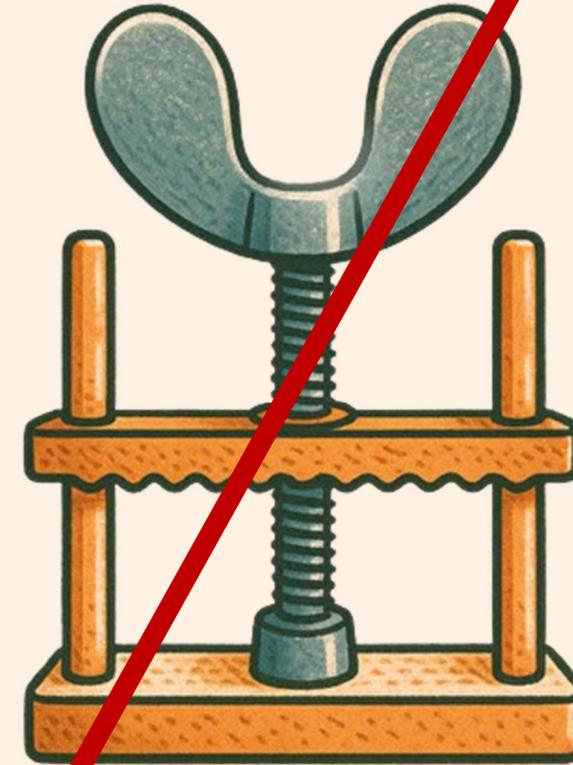


Neuling



fortgeschrittener
Anfänger

Gewandter



Kompetenter

Experte

Das richtige Drehmoment hängt von Fähigkeiten ab!

Evolution der Governance

Falls Software erfolgreich ist ...

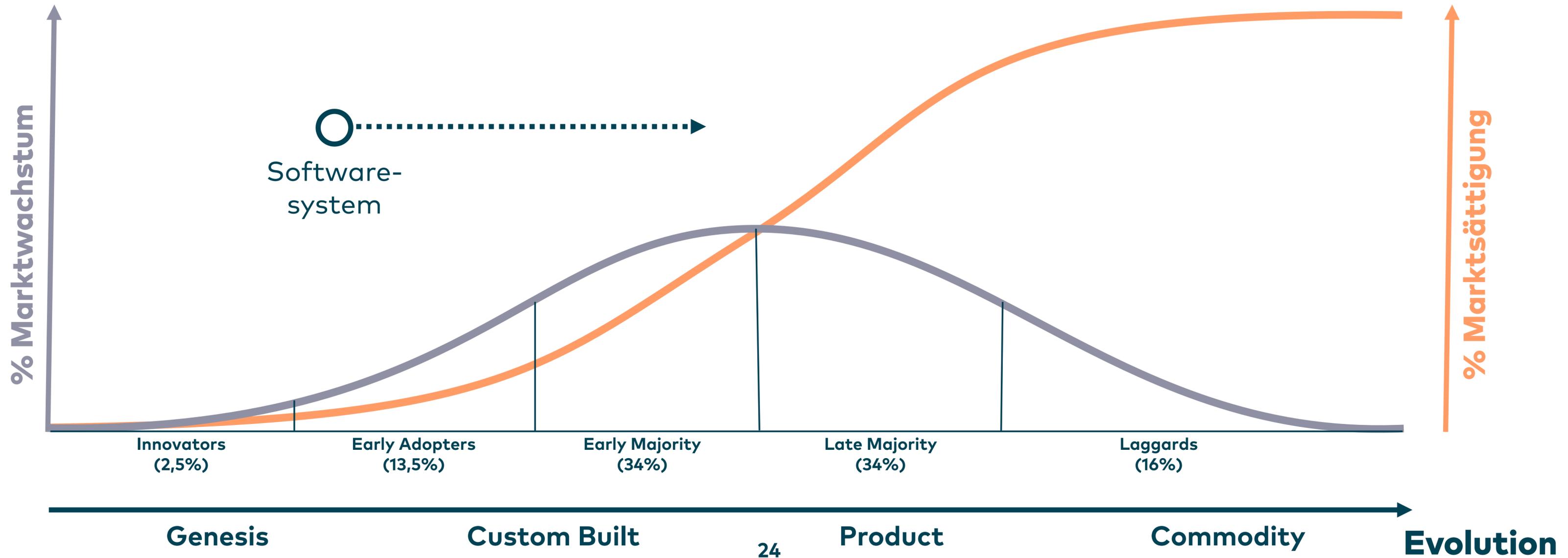


„Alles entwickelt sich durch den Wettbewerb von Angebot und Nachfrage“ – Simon Wardley



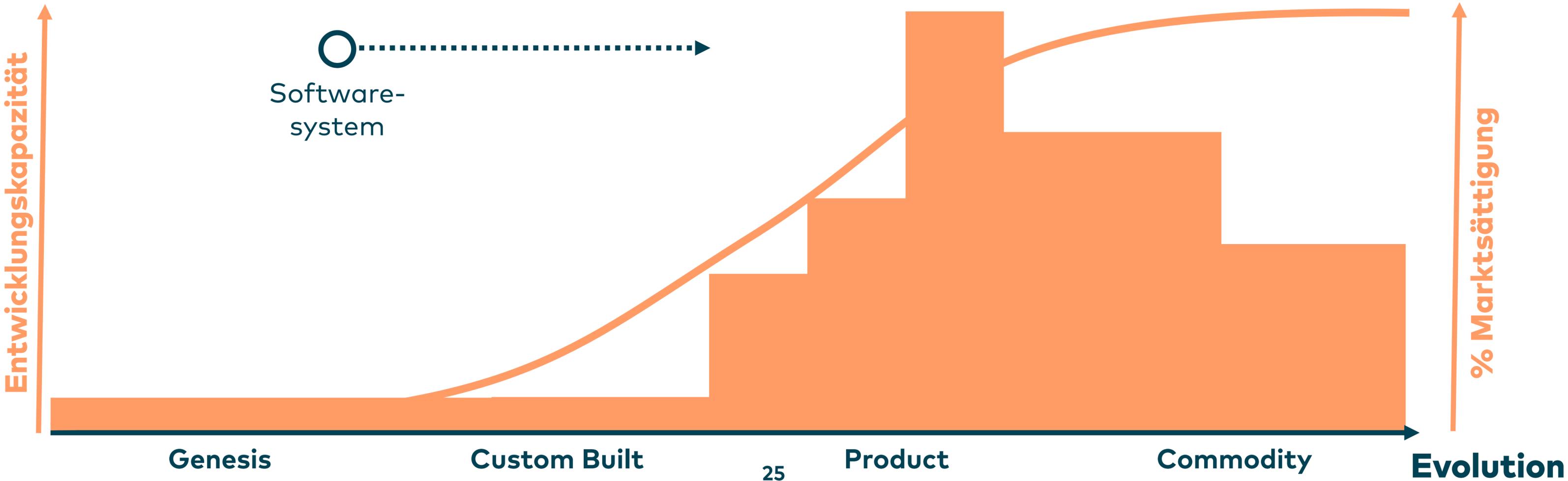
Wer will unsere Software?

Nachfragewettbewerb (oder: Technology Adoption Curve FTW)

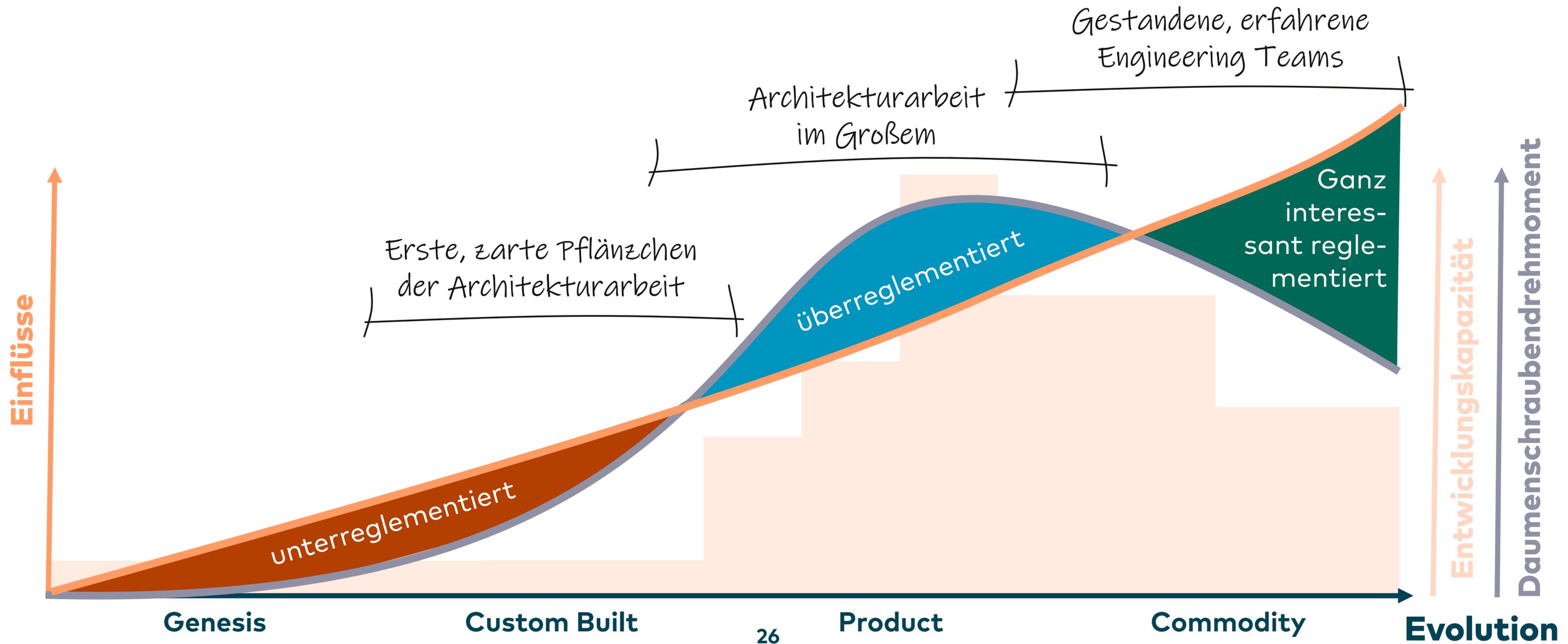


Wer muss mitarbeiten?

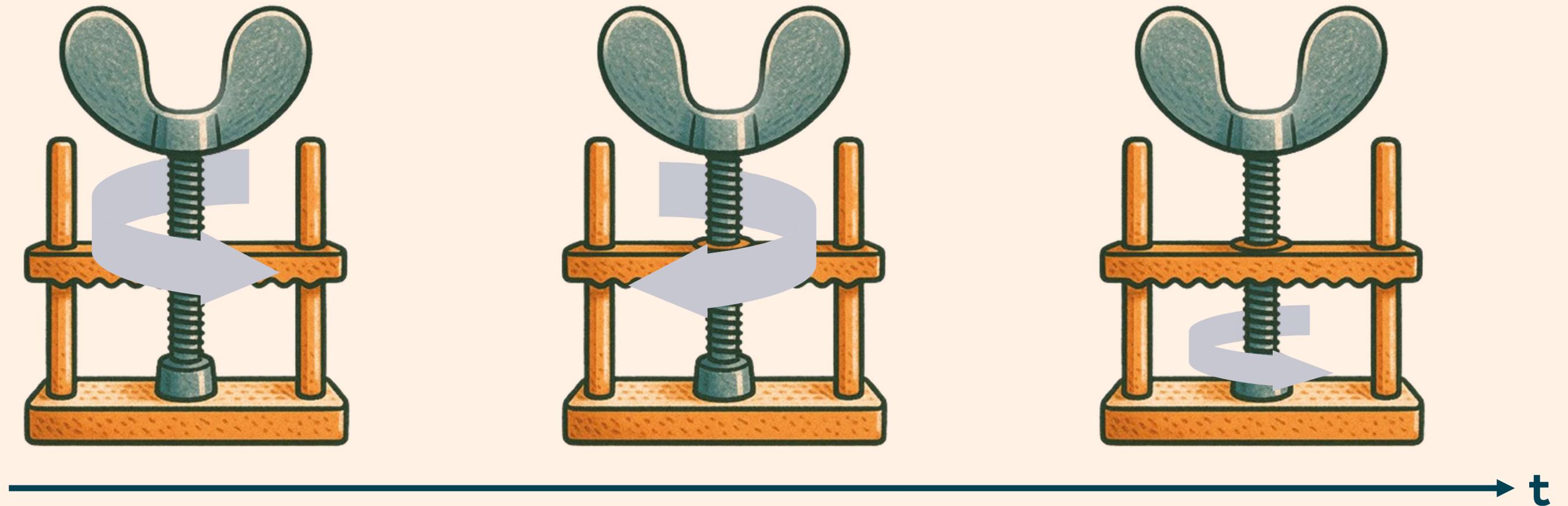
Angebotswettbewerb: Lieferfertigkeit muss nachziehen



Wie stark müssen wir regeln?



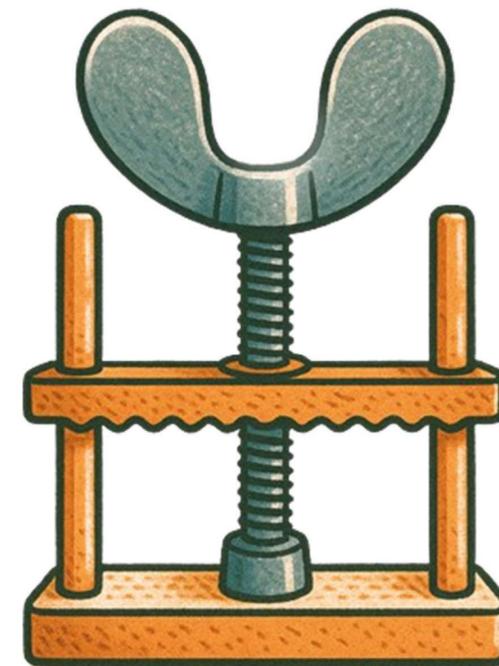
Was bedeutet das für uns?



Das richtige Drehmoment für die Daumenschrauben zu finden, ist eine kontinuierliche Aufgabe!

Governance-Strategien

Governance-Strategien

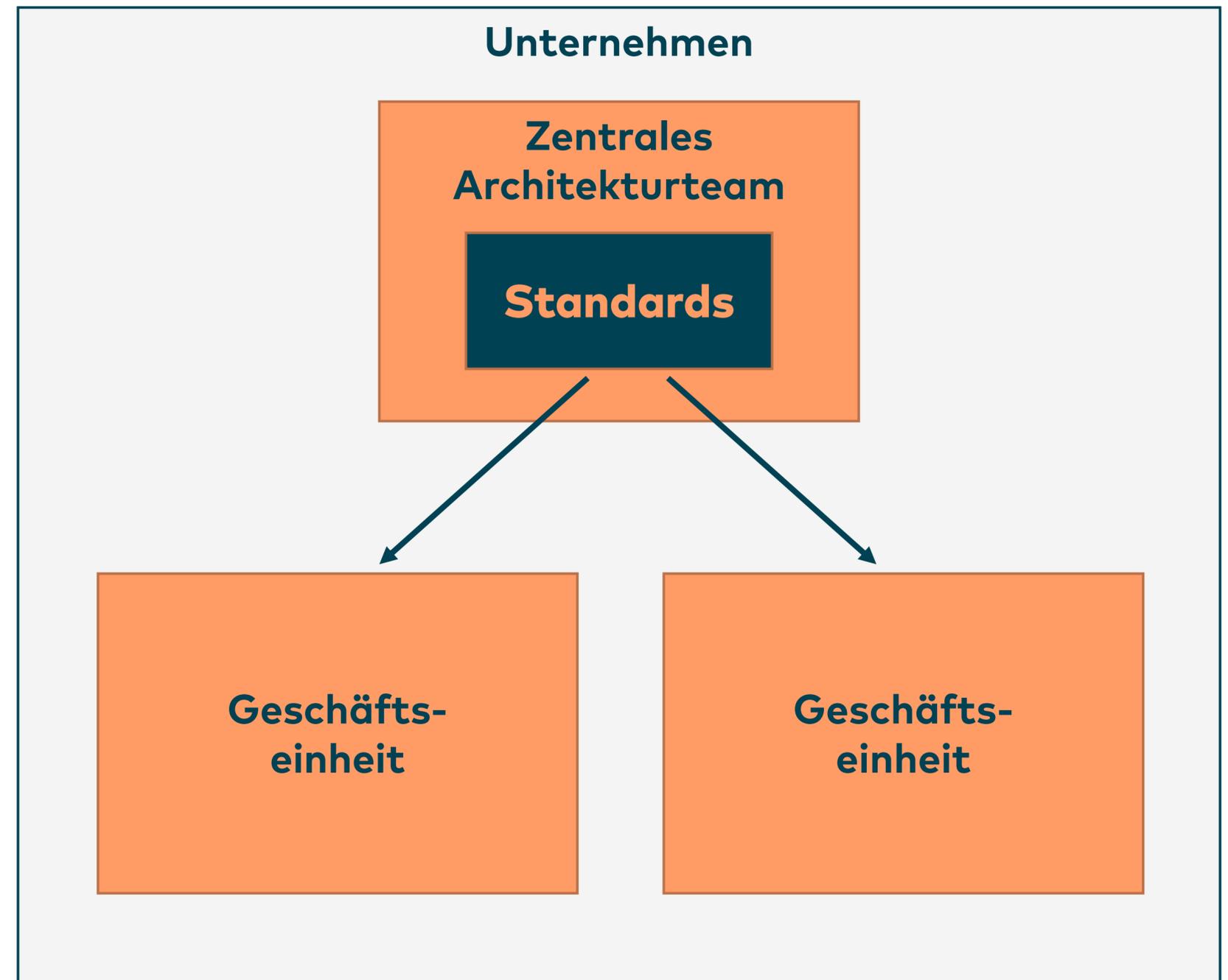


Fixe Vorgaben

zentral

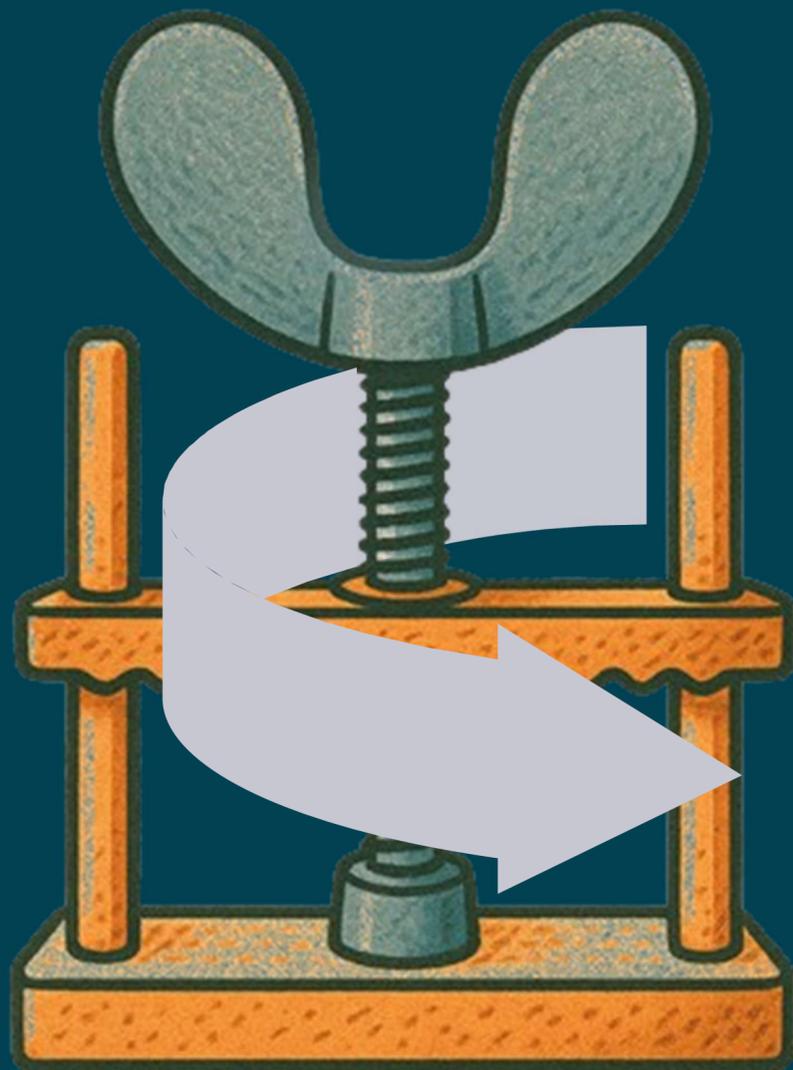
- + reduziert Komplexität
- + weniger Zeitbedarf bei Entscheidungen
- + hohe Wiederverwendung
- + kostengünstig
- passt evtl. nicht überall
- schwer, Akzeptanz zu schaffen
- nicht jede(r) ist glücklich
- starke Kontrolle notwendig

Harte Festlegungen



Fixe Vorgaben

zentral



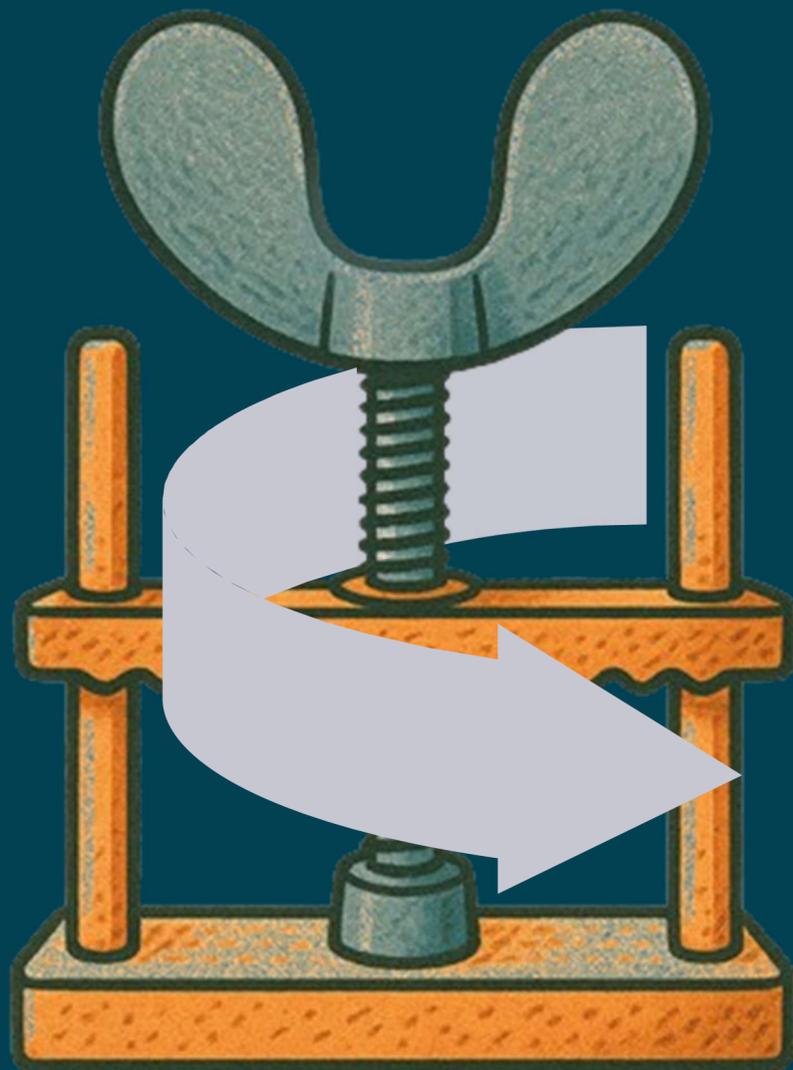
Alle Entwicklungsteams
müssen Oracle JDK
21.0.6 verwenden

Flexible
Entwicklungs-
planung

Standards

Fixe Vorgaben

zentral



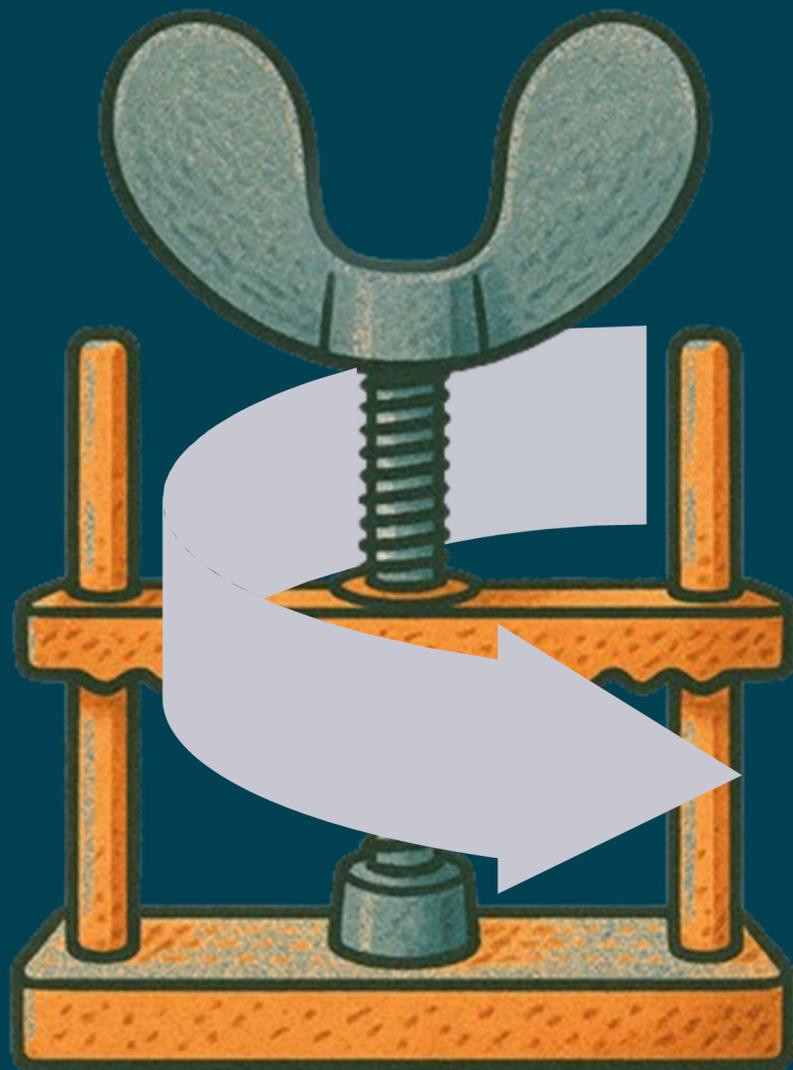
Der Default-Regelsatz
von SonarQube meldet
keine Major und Critical
Findings

Hohe Code-
Qualität

Standards

Fixe Vorgaben

zentral



Die Test-Coverage der
Unit-Tests muss min.
70 % betragen.

Funktionierende
Software

Standards

Fixe Vorgaben zentral

Was soll schon schief gehen?

Teams könnten
fangen an zu
cheaten

JETZT SCHLAEGT POWER 2 ZURUECK
ACTION REPLAY
PRO Version Cartridge DM 149,- **WERDEN SIE UNBESIEGBAR!!**
ZZGL DM 10,- VERSANDKOSTEN
JETZT BEEHAELTLICH FUER DEN
* **SUPER NES**™
(FUNKTIONIERT ALS ADAPTER FUER AMERIKANISCHE UND JAPANISCHE SPIELE)
* **MEGADRIVE**™

WERDEN SIE SPIEL-PROFI MIT ACTION REPLAY
MIT DEM ACTION REPLAY CARTRIDGE KOENNEN SIE IHRE LIEBLINGSSPIELE BIS ZUM LETZTEN LEVEL DURCHSPIELEN
UNENDLICHES LEBEN, UNBEGRENZTE ENERGIE, UNBEGRENZTE POWER, MEHR TREIBSTOFF, WERDEN SIE UNBESIEGBAR MIT DEM ACTION REPLAY CARTRIDGE FUER IHRE MEGADRIVE™ UND SUPER NES™ CONSOLE

- ACTION REPLAY IST EINE NEUE ENTWICKLUNG, DIE DEN LSI CUSTOM CHIP BEIHÄLTET, DIESER WURDE SPEZIELL ANGEFERTIGT, UM DEN BENÜTZER DIE ÄNDERUNG DER SPIELPROGRAMMIERUNG ZU ERMOEGELICHEN, SO DASS MAN LIEBLINGSSPIELE BIS ZUM ENDE DURCHSPIELEN KANN
- DURCH DEN EINZIGARTIGEN "GAME-TRAINER" IST ES MOEGlich, LEVELS ZU SPIELEN, DIE MAN ZUVOR NOCH NIE GESEHEN HAT UND IHRE EIGENE PARAMETER ZU FINDEN, FUER MEHR POWER, ENERGIE, TREIBSTOFF, UNENDLICHES LEBEN, USW. . . .
- MIT ACTION REPLAY PRO FINDEN SIE IHRE EIGENEN PARAMETER, ES DAUERT NUR MINUTEN, UM NEUE PARAMETER ZU FINDEN, DIESES IST DAS MODUL, DAS VON PROFIS EINGESETZT WIRD
- ACTION REPLAY FUNKTIONIERT AUCH ALS ADAPTER FUER JAPANISCHE UND AMERIKANISCHE SPIELMODULE. DABURCH HABEN SIE DIE MOEGlichkeit, IMPORTIERTE SPIELMODULE AUF DEUTSCHE SPIELCONSOLES ZU SPIELEN
- MIT DER SPEZIELL ENTWICKELTEN ASIC HARDWARE IST ES MOEGlich, DIE NEUESTEN SPIELE DIREKT NACH DEREN NEUERSCHEINUNG ZU MANIPULIEREN
- KEINE SPEZIELLEN VORKENNTNISSE NOTWENDIG. WENN SIE EIN SPIEL SPIELEN KOENNEN, IST ES SCHON MOEGlich, DAS ACTION REPLAY EINZUSETZEN. ALLE EINGABEN WERDEN UEBER DEN JOYSTICK/PAD EINGEGEBEN. EINFACHER GEHT ES NICHT.

DAS ACTION REPLAY MODUL IST EIN UNENTBEHRliches HILFSMITTEL FUER JEDEN SPIELCONSOLE-BESITZER. LESEN SIE DEN TESTBERICHT IM GAMERS-MAGAZIN 03/92

JETZT KOENNEN SIE AMERIKANISCHE UND JAPANISCHE MODULE AUF IHRER DEUTSCHEN SUPER NES-CONSOLE SPIELEN

- MIT DIESEM UNIVERSELLEN SPIELADAPTER KOENNEN SIE DIE ENORME ANZAHL VON AMERIKANISCHE UND JAPANISCHE MODULE AUF IHRER DEUTSCHEN SUPER NES-CONSOLE SPIELEN
- FUNKTIONIERT AUCH ALS DOPPELMODULADAPTER. ES ERLAUBT, ZWEI MODULE GLEICHZEITIG ANZUSCHLIESSEN. EINFACH ZWISCHEN DEN MODULEN HIN- UND HERSCHALTEN. HIERDURCH WIRD IHR GAMEPORT GEGEN VERSCHLEISS GESCHUETZT
- SEHR EINFACH IN DER HANDHABUNG
- ES ERLAUBT, DEUTSCHE MODULE AUF AMERIKANISCHE UND JAPANISCHE CONSOLES EINZUSETZEN

DM 59,-
ZZGL DM 10,- VERSANDKOSTEN

* **WICHTIG!**
ACTION REPLAY IST NICHT ENTWICKELT, WIRD NICHT HERGESTELLT UND VERTRIEBEN DURCH SEGA ENTERPRISES LTD.

DATA Flash
WIE BESTELLEN SIE AM SCHNELLSTEN!
TELEFON (24 STUNDEN SERVICE) 02822 68545 | 687182
ALLE BESTELLUNGEN NORMALERWEISE IN 48 STUNDEN LIEFERBAR...
DATAFLASH GMBH
WASSENBERGSTRASSE 34 4240 EMMERICH
FAX 02822 68547
BEI BESTELLUNG BITTE IHREN CONSOLE-TYP ANGEBEN !!!

DATAFLASH GMBH
4240 EMMERICH
WASSENBERGSTR. 34

Foto: Kantenflimmern CC BY-SA 2.0

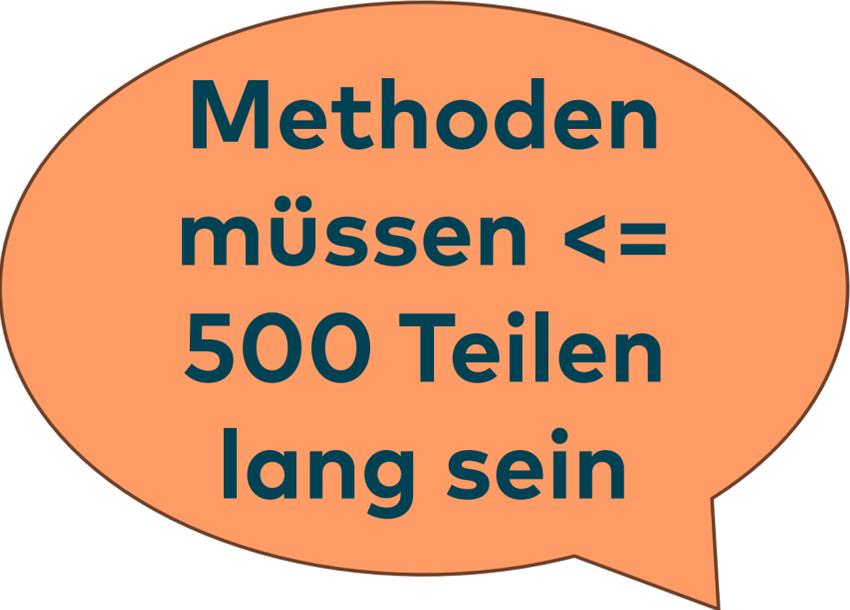
<https://www.flickr.com/photos/kantenflimmern/3078108108/>

```
protected void processRequest1 (HttpServletRequest request,
    HttpServletResponse response) throws Exception {

    locale = LocaleResolver.getLocale (request);
    EventCRFBean ecb =
        (EventCRFBean) request.getAttribute (INPUT_EVENT_CRF);
    SectionBean sb = (SectionBean) request.getAttribute (SECTION_BEAN);
    <496 Lines of Code>
    processRequest2 (request, response, fp);
}
```

```
protected void processRequest2 (HttpServletRequest request,
    HttpServletResponse response) throws Exception {

    Boolean b = (Boolean)
        request.getAttribute (INPUT_IGNORE_PARAMETERS);
    isSubmitted = fp.isSubmitted() && b == null;
    int eventDefinitionCRFId = 0;
    ...
}
```

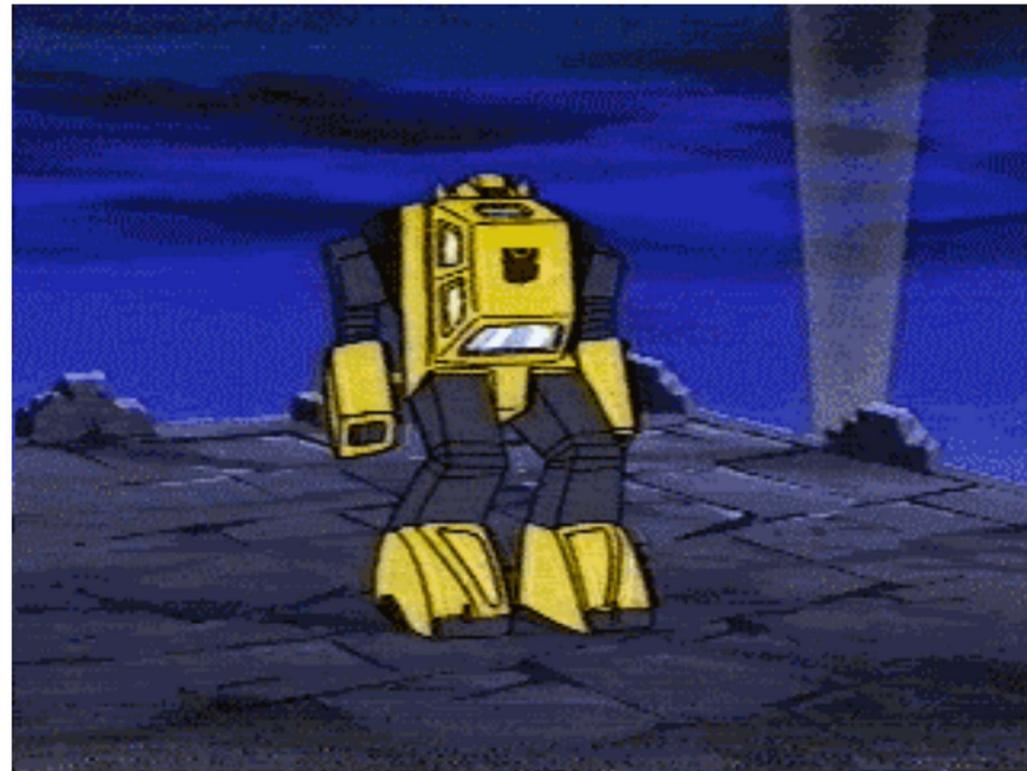


**Methoden
müssen <=
500 Teilen
lang sein**

volkswagen

Volkswagen detects when your tests are being run in a CI server, and makes them pass.

build unknown code style standard build passing 



<https://github.com/auchenberg/volkswagen>

Fixe Vorgaben

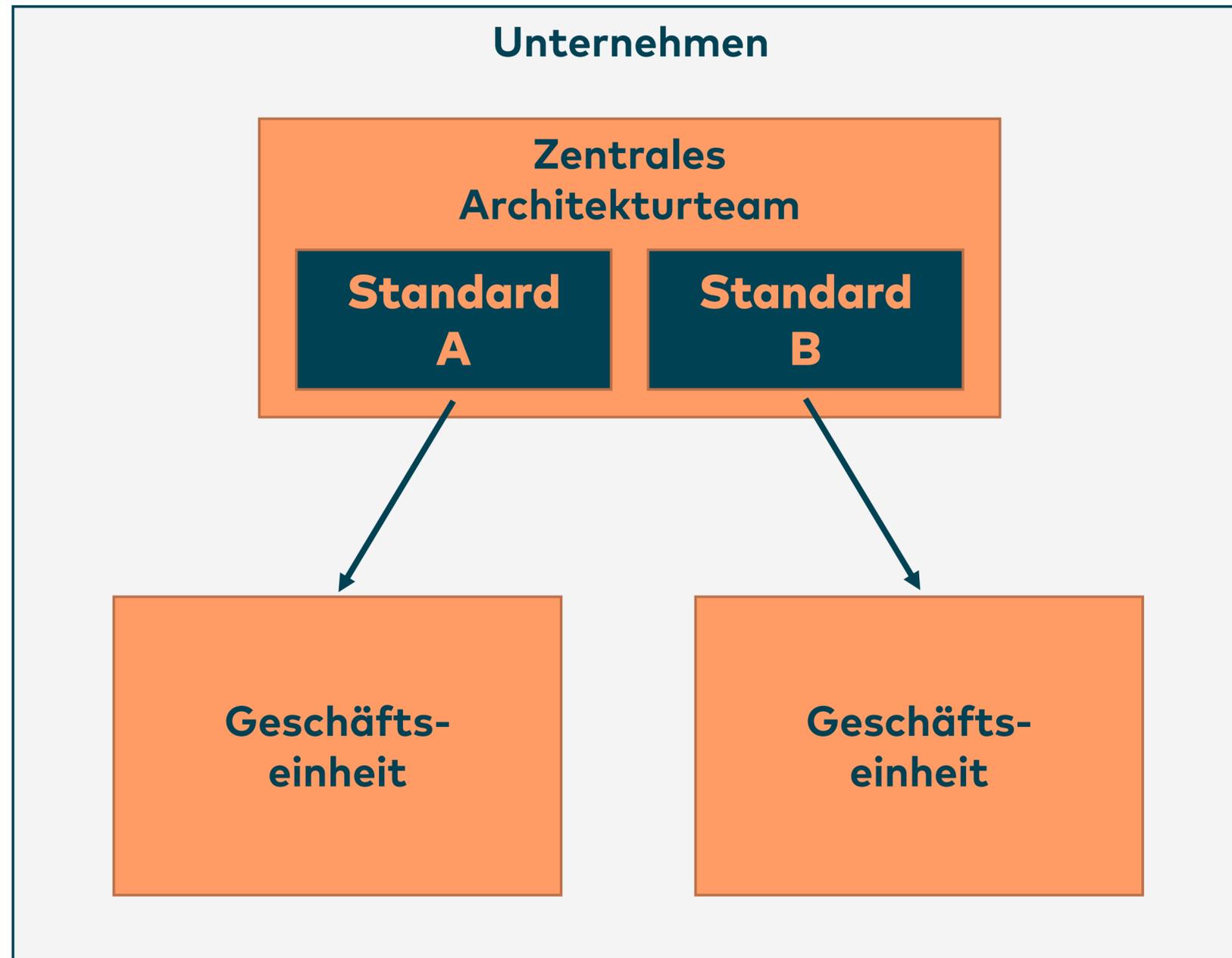
zentral

Tipps

- Das „Warum“ deutlich machen
- Ausgeschlossene Varianten kennzeichnen (ADRs)
- Regelmäßig Feedback einholen (oder Entscheidungen selbst ausbaden)
- Bei Bedarf anpassen

ADR: Architecture Decision Record

Mehrere Alternativen



Paved Roads eher zentral

- + richtiges Werkzeug für den Job
- + mehr Kontrolle der Auswahl
- + bessere Zufriedenheit
- braucht Zeit für Auswahl
- Wahl / Zusammensetzung könnte nicht die richtige sein
- höhere Kosten

**Wir entwickeln unsere
Enterprise-Applikationen
mit Java und unsere
Datenanalysen mit
Python / Pandas**

**Flexible
Entwicklungs-
planung**

Standards

**Paved
Roads
eher zentral**

Mögliche Paved Roads

Attraktive Defaults statt harter Vorgaben

Infrastruktur ✓

Tooling ✓

Automatisierung ✓

...

CI/CD-Pipeline ✓

Guidelines ✓



Tipps

- Alle Alternativen gleichwertig behandeln und weiterpflegen
- Ausnahmeregelungen dennoch zulassen
- Feedback-Loop einbauen

ADR: Architecture Decision Record

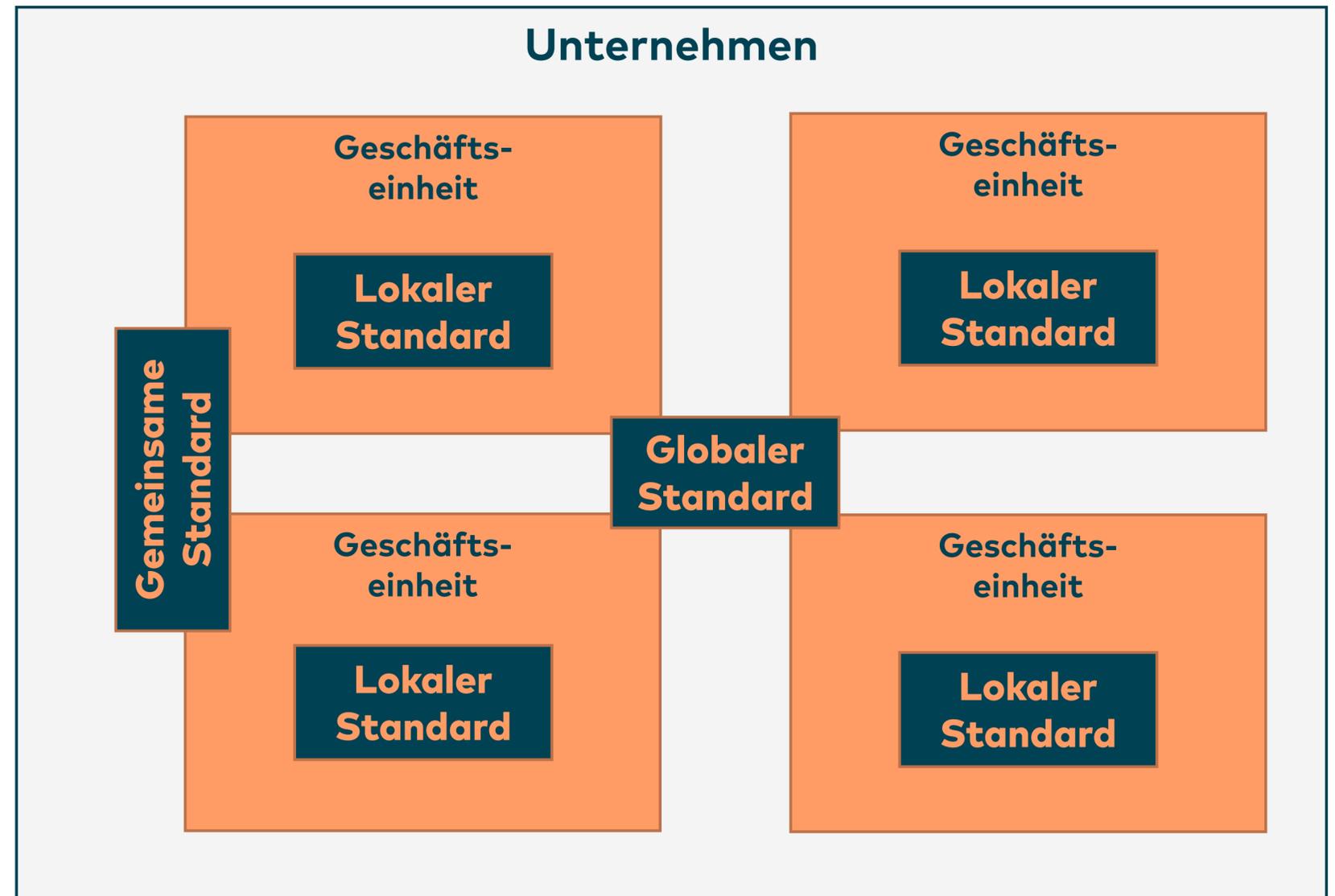
**Paved
Roads**
eher zentral

Multi Level

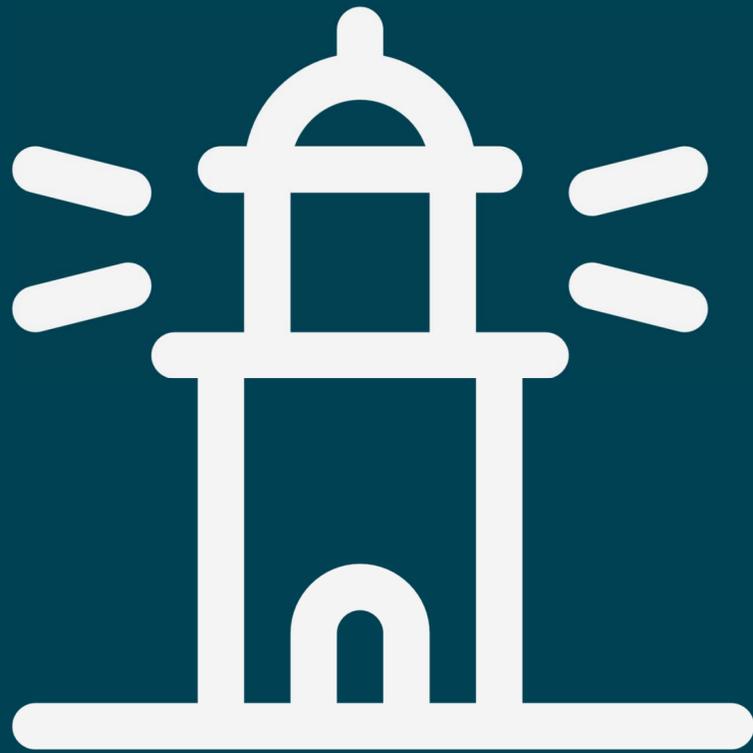
eher dezentral

- + richtiges Werkzeug für den Job
- + Geschäftseinheit hat Kontrolle
- + minimale, zentrale Vorgaben
- + bessere Gesamtzufriedenheit
- fehlende Abstimmungen
- hohe Kosten
- erschwerte Kostenkontrolle
- schwer global zu steuern

Verteilte, hierarchische Standardisierung



(Architektur-)Prinzip



„Ein Architekturprinzip ist eine deklarative Aussage, die mit der Absicht getroffen wird, architektonische Designentscheidungen zu leiten, um eine oder mehrere Qualitäten eines Systems zu erreichen.“

- Eoin Woods

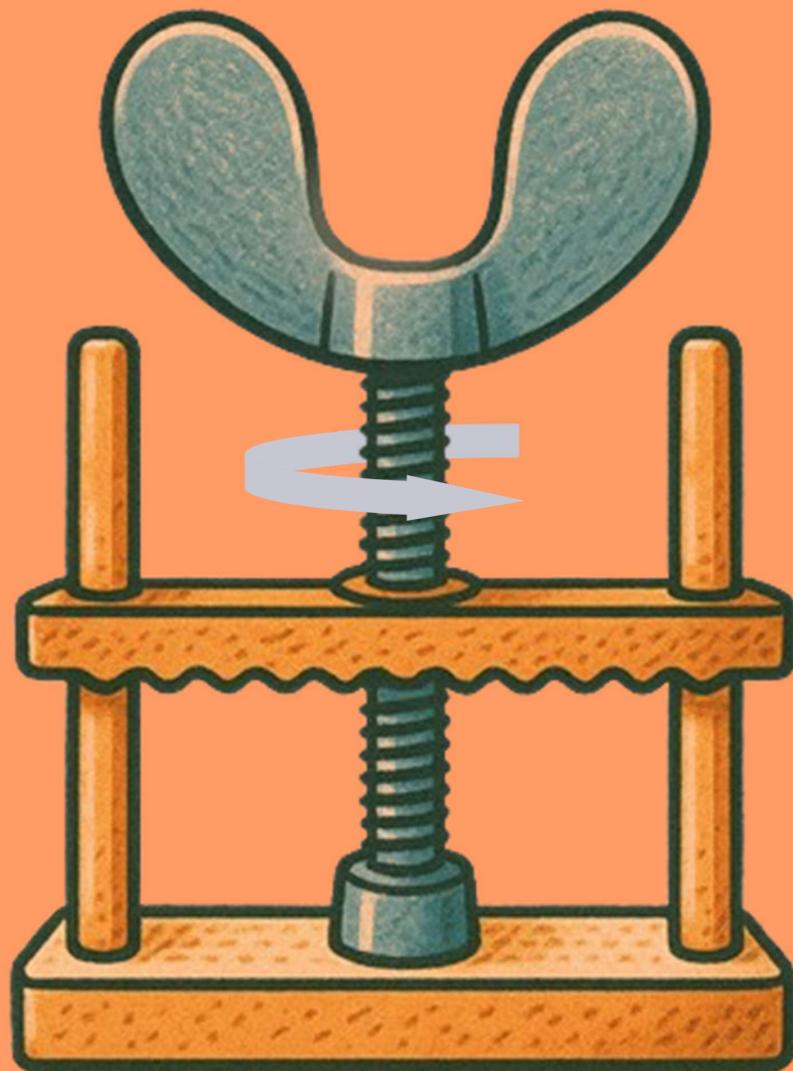
Leitplanken

- = **Universell nützliche High-Level-Prinzipien**
 - Legen die wichtigsten Prioritäten fest
 - Helfen dem Team, auf Kurs zu bleiben
- Vermeidet auch, dass es widersprüchliche Interessen gibt



Multi Level

eher dezentral



Flexible
Entwicklungs-
planung

High level

Uns ist wichtig,
dass wir
Entwicklerinnen
und Entwickler
schnell auf alle
anderen Projekte
einarbeiten können

Mid level

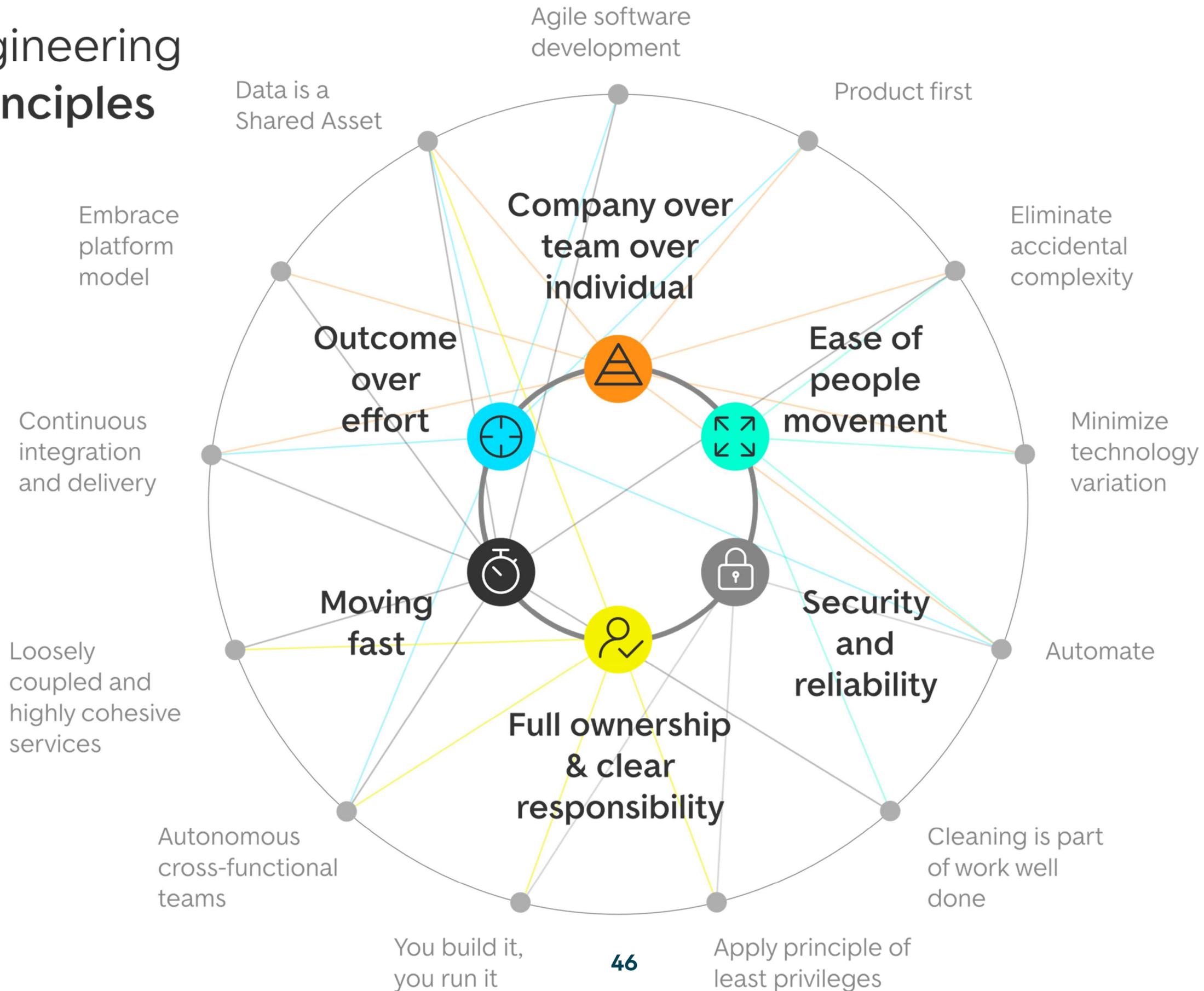
Alles auf
der JVM ist
OK

Standards

Lass mal
Java
nehmen

Low level

Scout24 Engineering Values & Principles



Values



Company over team over individual

We value decisions that have a positive impact for the entire tech organization even if they have a negative impact on our own team. We rationally assess the impact and acceptance of both new and current technologies on other teams. We look for opportunities to help other teams benefit from what we are building.



Ease of people movement

The movement of people into teams with complex, high priority work is an important part of how Scout24 will continue to grow as a business. To reduce onboarding barriers, we value standardization of technologies and processes across Scout24. We value engineers who can work across multiple stacks.

<https://github.com/Scout24/scout24-engineering-values-and-principles/>

Principles

Agile software development

We listen to user needs and release value often so we can learn from successes and failures and adjust our approach quickly.

Continuous integration and delivery

We, as team members, continuously integrate our software changes with those of our teammates and push these changes to production automatically.

Minimize technology variation

We deliberately minimize the variety of technologies in our system to drive efficiency, and, at the same time, have a managed program of innovation.

Automate

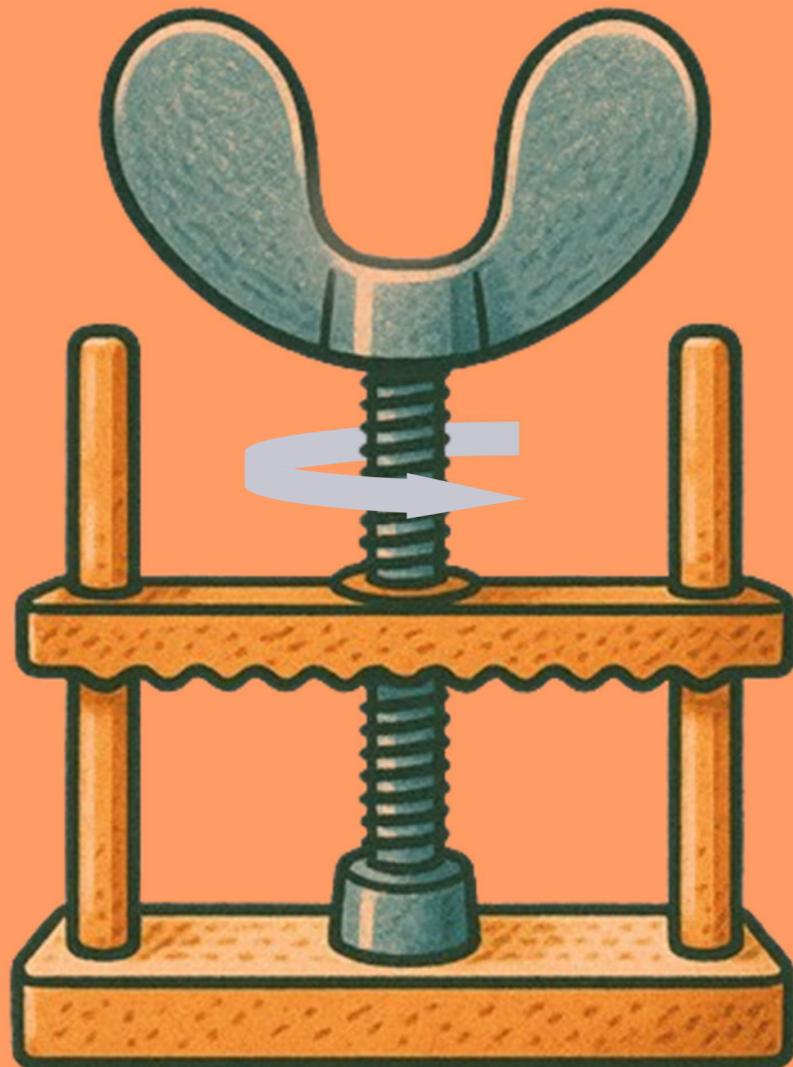
We automate repetitive activities to achieve better predictability, efficiency, reproducibility, resiliency, and security.

Loosely coupled and highly cohesive services

We strive to minimize the coupling of services and maximize their cohesion in order to, among other things, limit failure propagation, reduce response times and limit the scope of software changes.

Multi Level

eher dezentral



Funktionierende
Software

High level

Design für
Testbarkeit

Standards

Wir setzen für kritische
Komponenten Mutation
Testing ein

Low level

Beispiel Architekturprinzip 1/2

Prinzip: Design für Testbarkeit

Die Lösungen sollten so konzipiert - und der Code so strukturiert - sein, dass die Ausführung der Tests einfacher und schneller erfolgt.

Begründung

Lösungen, die unter Berücksichtigung von Testaspekten entwickelt werden, ermöglichen ein schnelleres Feedback und können letztendlich häufiger und sicherer an die Kunden weitergegeben werden. Eine testorientierte Entwicklung führt automatisch zu einer besseren Verständlichkeit und Evolvierbarkeit.

Beispiel Architekturprinzip 2/2

Auswirkungen

- Strukturiere deinen Code so, dass die Komponenten isoliert ausgeführt werden können, um ihr Verhalten bei der Überprüfung und beim Testen zu beobachten.
- Stelle sicher, dass die Komponenten eines Systems in klar definierte Verantwortlichkeiten aufgeteilt sind.
- Die Komponenten eines Systems sollten leicht zu verstehen sein, d.h. der Code sollte so geschrieben sein, dass er sich selbst erklärt und durch seine Tests oder, falls nötig, durch eine separate Dokumentation dokumentiert wird.

Weitere Informationen

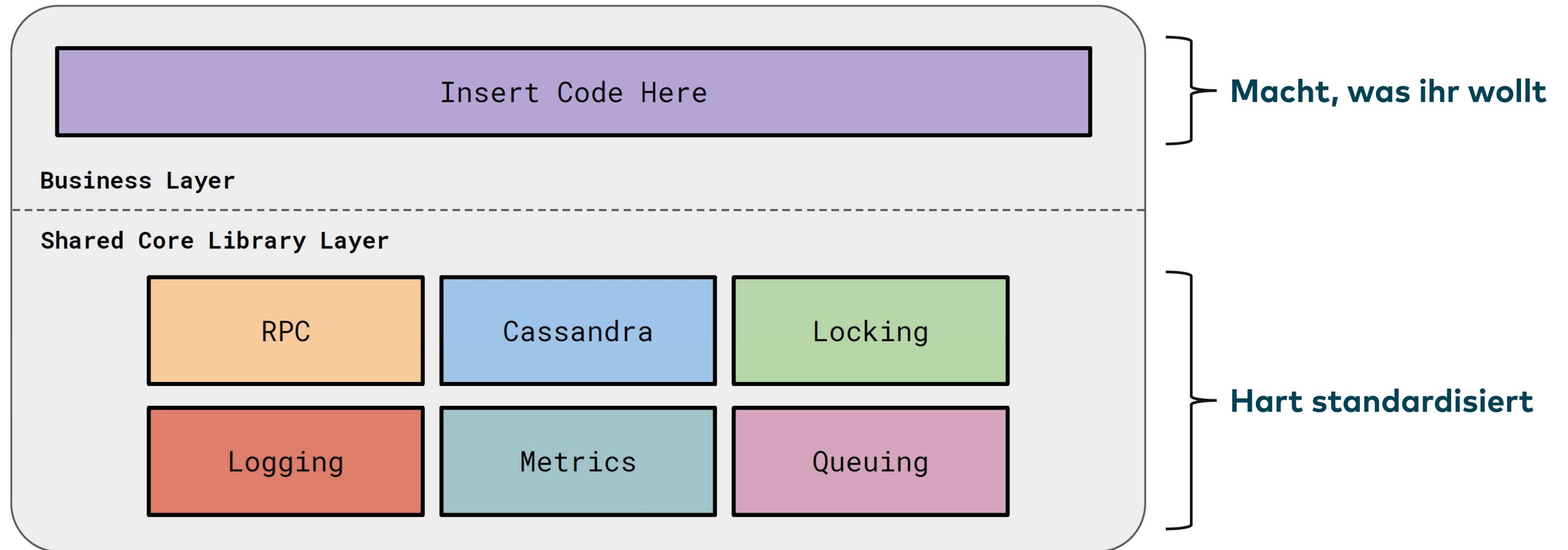
- Testability blog post von Michael Bolton.
- Team Guide to Software Testability von Ash Winter and Rob Meaney.

Architecture Decision Record (ADR)

Entwurfsentscheidungen nachvollziehbar dokumentieren

- **Titel:** Um welche Entscheidung geht es?
- **Kontext:** Was waren die Rahmenbedingungen?
- **Entscheidung:** Was war die Entscheidung, was waren die Alternativen?
- **Status:** Wie steht es um das ADR?
 - hier auch abgelehnte oder überholte ADRs aufführen
- **Konsequenzen:** Mit welchen Vor- und Nachteilen wollen wir jetzt leben?

Beispiel Monzo (Bank aus UK)



Aus dem Vortrag "Modern Banking in ~~1500~~ 1600 Microservices"
<https://www.infoq.com/presentations/monzo-microservices/>

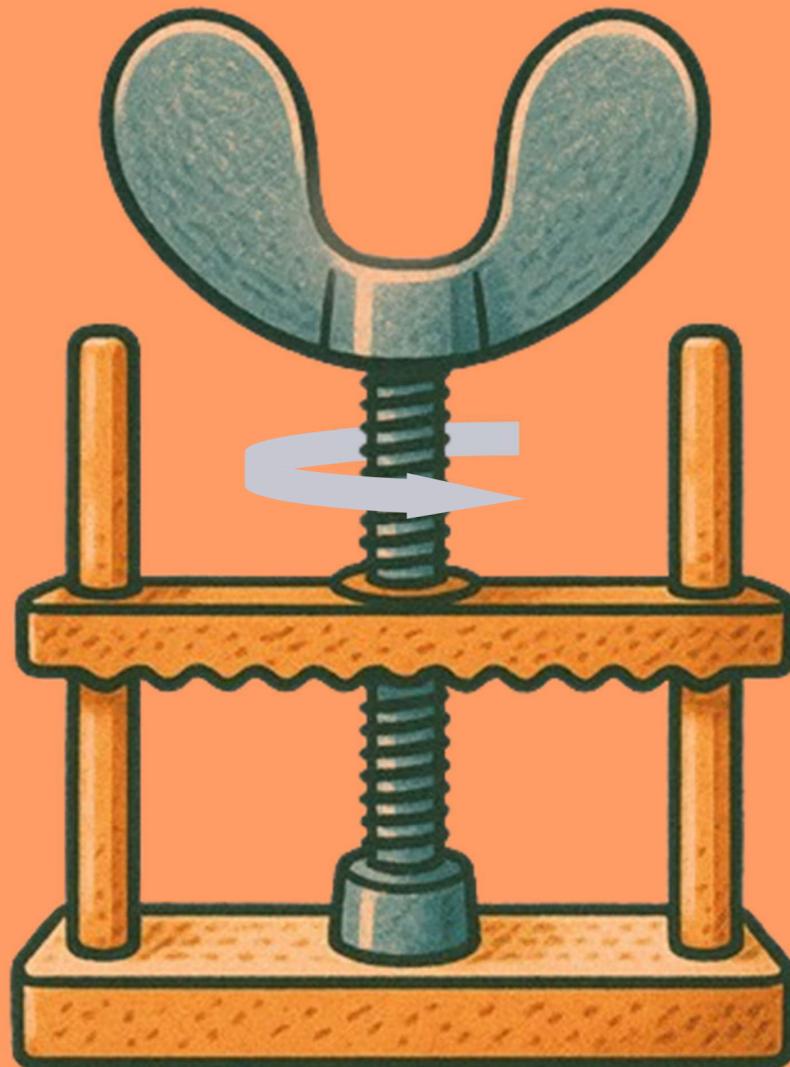
Beispiel Monzo (Bank aus UK)

Engineering Principles

1. Ship it and iterate.
2. Make changes small, make them often.
3. Technical debt is a useful tool.
4. Solve problems at the root.
5. Do not accept deviant system behaviour.
6. Write code to be read.
7. Write code to be debugged.
8. If you can't show it's a bottleneck, don't optimise it.
9. Unblock others whenever you can.
10. Leave the codebase better than you found it.

Multi Level

eher dezentral

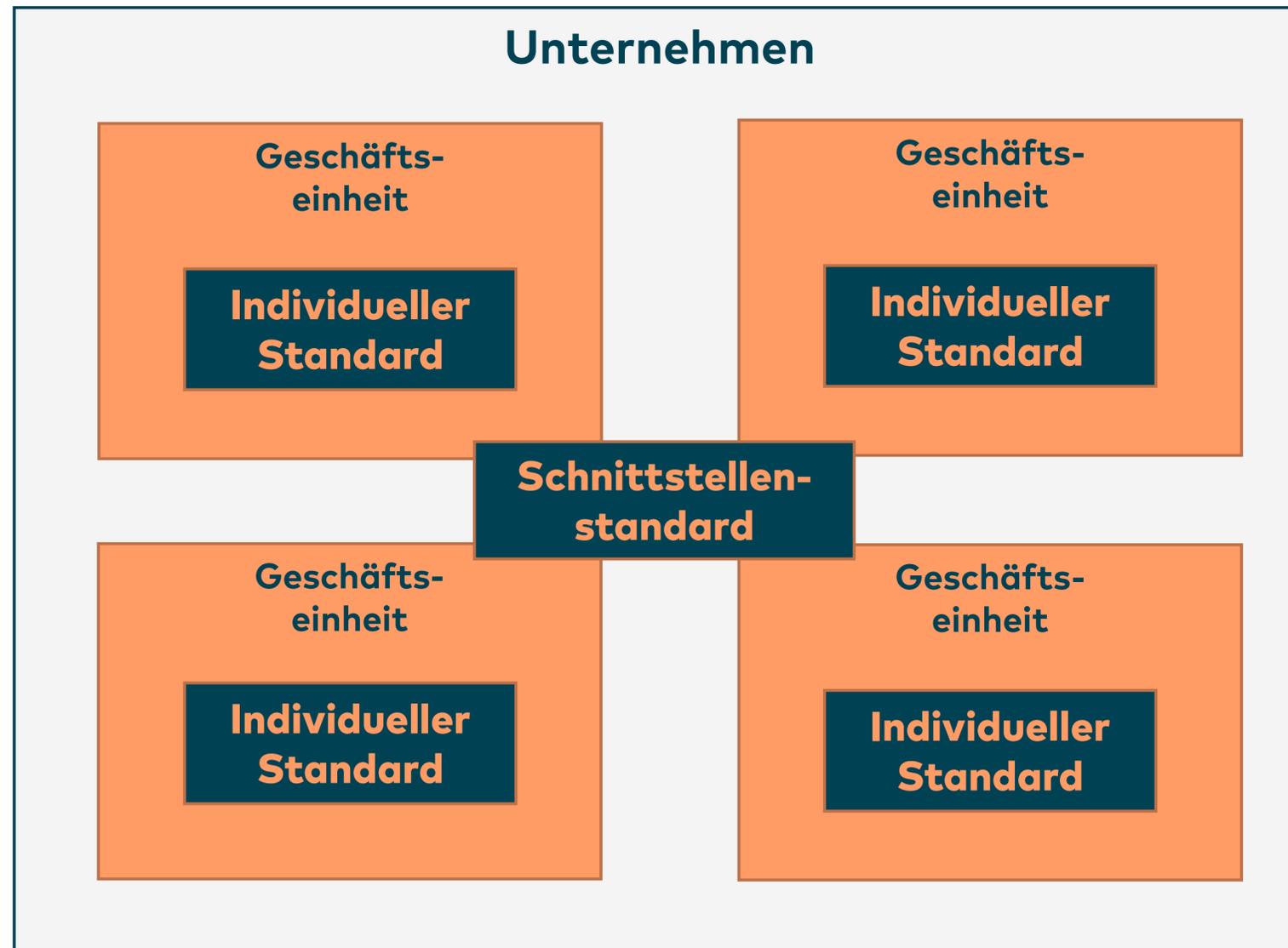


Tipps

- Fäden ziehen: Was wollen wir erreichen? Warum?
- Konkret beschreiben / kein Blabla
- Lokale Entscheidungen/ Best Practices kommunizieren
- Austauschformate einsetzen (CoP)
- Spezialistinnen und Spezialisten benennen als Ansprechpartner

CoP: Community of Practice

Individuelle Standards mit definierten Schnittstellen



API Mandate

dezentral

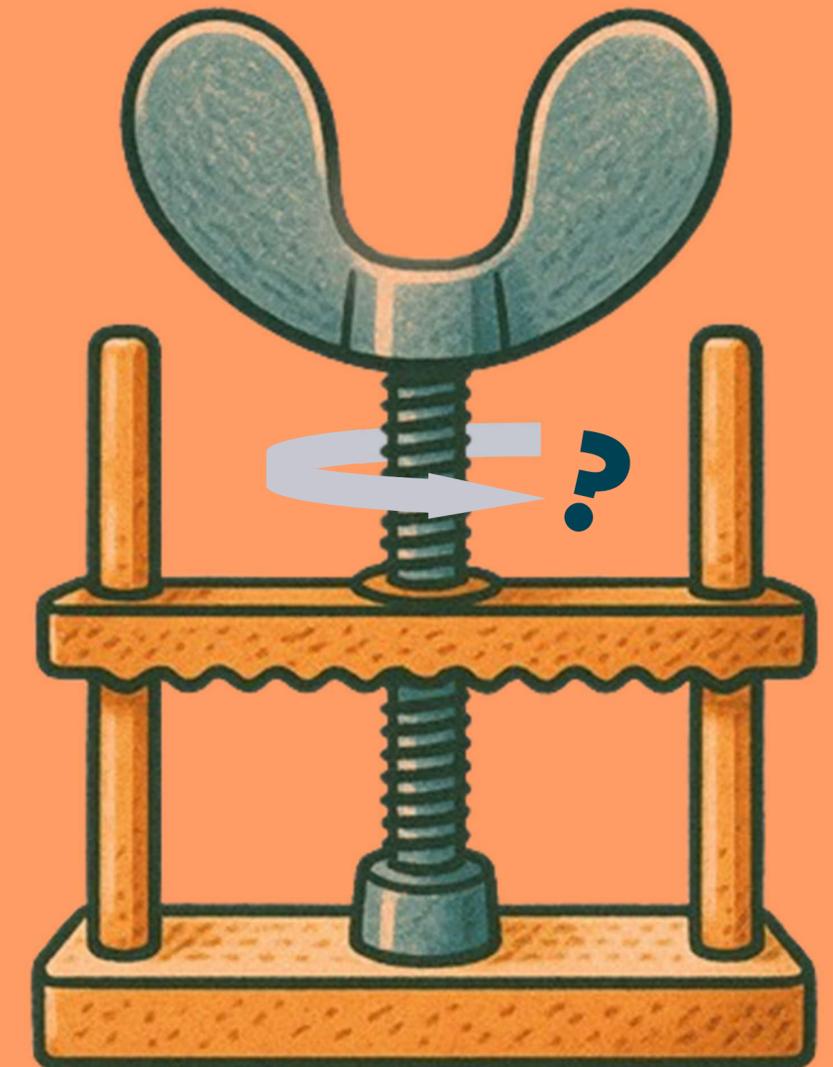
- + richtiges Werkzeug für den Job
- + Geschäftseinheit hat Kontrolle
- + Synergien zwischen Einheiten
- + bessere Zufriedenheit
- braucht Zeit für Auswahl
- höchste Kosten
- schwerste Kostenkontrolle

JEFF BEZOS API MANDATE

- All teams will expose their data and functionality through service interfaces.
- Teams must communicate with each other through these interfaces.
- There will be no other form of interprocess communication allowed.
- No exceptions,
- All service interfaces, without exception, must be designed from the ground up to be externalizable.
- Anyone who doesn't do this will be fired.
- Thank you; have a nice day!

API Mandate

dezentral

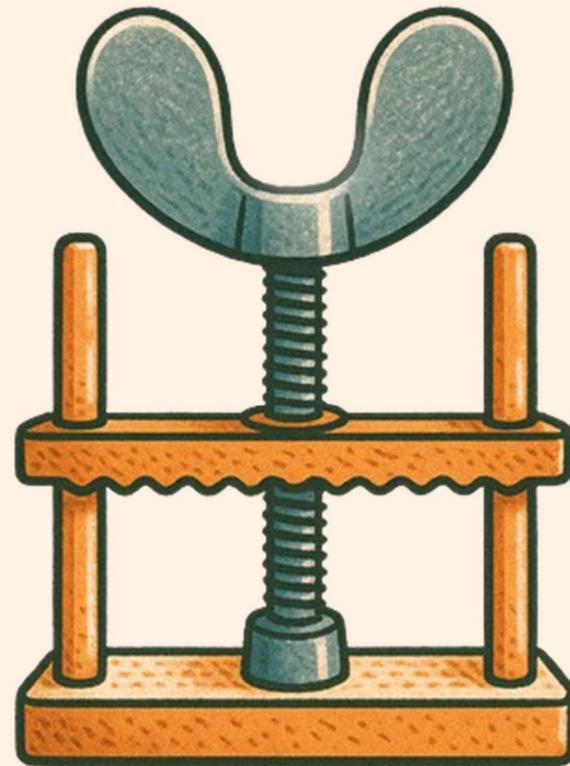


**"With great power
comes great
responsibility"**

Aus dem Vortrag "From Gates to Guardrails - Alternate
Approaches to Product Security" von Jason Chan
<https://www.youtube.com/watch?v=geumLjxtc54>



Was bedeutet das für uns?



Daumenschrauben lassen sich auch so anziehen, so dass man sie gar nicht bemerkt.

6 Tipps

zur Gestaltung
einer angenehmen
Architektur-Governance

Moderne Governance-**Meta**-Prinzipien

1. Weg vom Gesetz, hin zu Vision und Prinzipien (und Leitplanken)
2. Compliance automatisieren und delegieren
3. Gatekeeper zu Mitarbeitende machen
4. Gepflasterte Straßen bereitstellen
5. Informationen offen teilen / Kommunikationsmuster überdenken
6. An Evolution gewöhnen

Quelle: <https://www.thoughtworks.com/insights/articles/lightweight-technology-governance>

Beispiel

Compliance automatisieren und delegieren

Schlaue Architekturdokumentation

Business modules

Salespoint ships with a set of business modules that interact with each other. The following component diagram shows the overall structure of the setup. The relationships are used as follows:

- *uses* — The module has a Spring component dependency and actively interacts with the target module. Implies a type dependency into the target module as well.
- *listens to* — The module contains an event listener for events the target modules publishes. Implies a type dependency into the target module as well.
- *depends on* — The module has a general type dependency to the target module, i.e. it uses the target module as library.

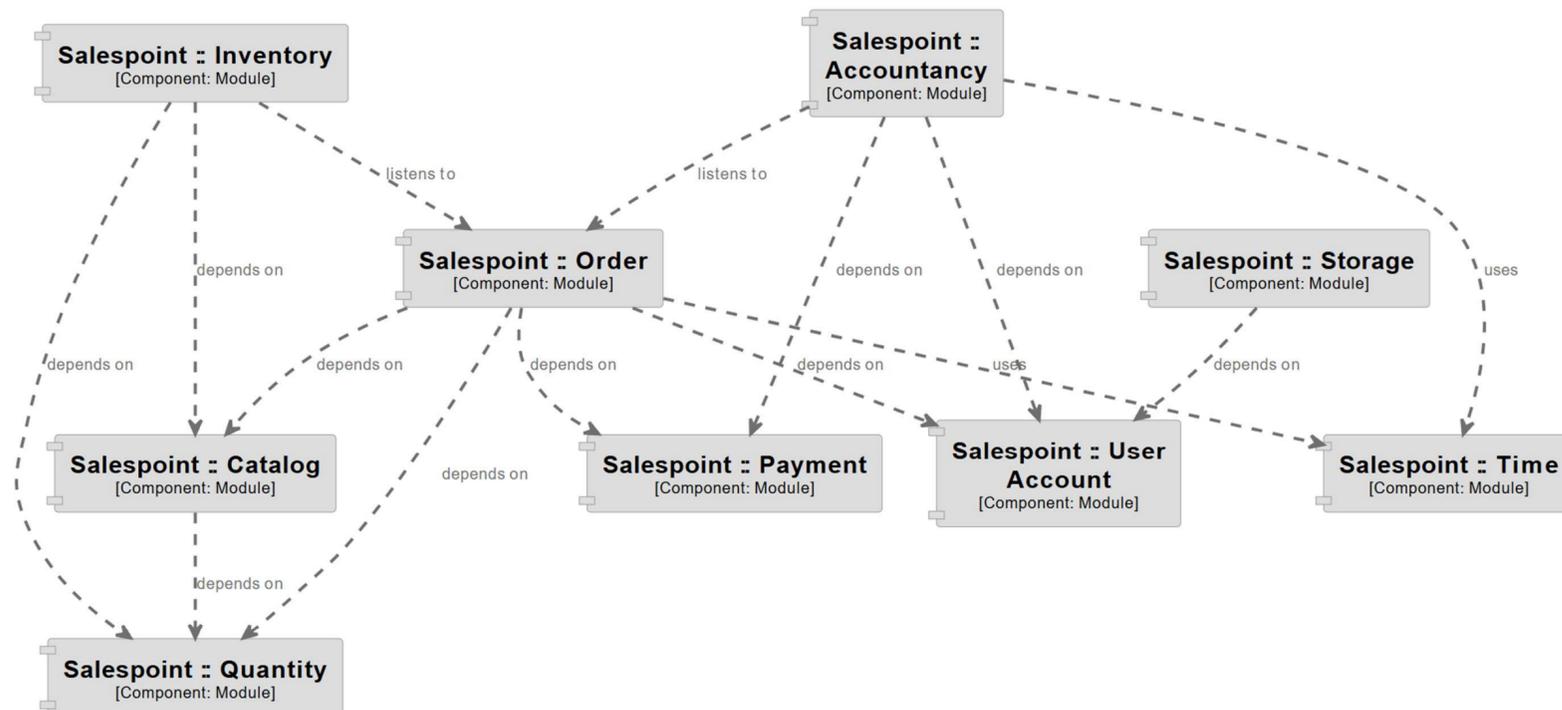
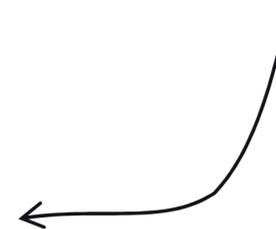


Figure 1. Salespoint component overview

Direkt aus dem Code erzeugte Doku



Schlaue Architekturdokumentation

User accounts



Figure 2. User account component

Base package	<code>org.salespointframework.useraccount</code>
Spring components	<i>Services</i> <ul style="list-style-type: none">• <code>o.s.u.UserAccountManagement</code> (via <code>o.s.u.PersistentUserAccountManagement</code>) <i>Others</i> <ul style="list-style-type: none">• <code>o.s.u.AuthenticationManagement</code> (via <code>o.s.u.SpringSecurityAuthenticationManagement</code>)
Aggregate roots	<ul style="list-style-type: none">• <code>o.s.u.UserAccount</code>
Value types	<ul style="list-style-type: none">• <code>o.s.u.EncryptedPassword</code>• <code>o.s.u.UnencryptedPassword</code>• <code>o.s.u.Role</code>
Published events	<ul style="list-style-type: none">• <code>o.s.u.UserAccountCreated</code> created by:<ul style="list-style-type: none">◦ <code>o.s.u.UserAccount.onCreate()</code>
Properties	<ul style="list-style-type: none">• <code>salespoint.authentication.login-via-email</code> — <code>java.lang.Boolean</code>, default <code>false</code>. Enables the login procedure to use the email address to lookup a user instead of their username. Defaults to <code>false</code>.

Schlaue Architekturdokumentation

```
@Entity
@NoArgsConstructor(access = AccessLevel.PRIVATE)
public class UserAccount extends AbstractAggregateRoot<UserAccountIdentifier> {
```

Salespoint :: User Account
[Component: Module]

```
/**
 * User account management.
 *
 * @see org.salespointframework.useraccount.UserAccountManagement
 */
@org.springframework.lang.NonNullApi
@org.springframework.modulith.ApplicationModule(displayName = "Salespoint :: User
package org.salespointframework.useraccount;
```

ArchUnit – Left Shifting Governance

Eine entwicklungsnahe Variante von Architektur-Governance

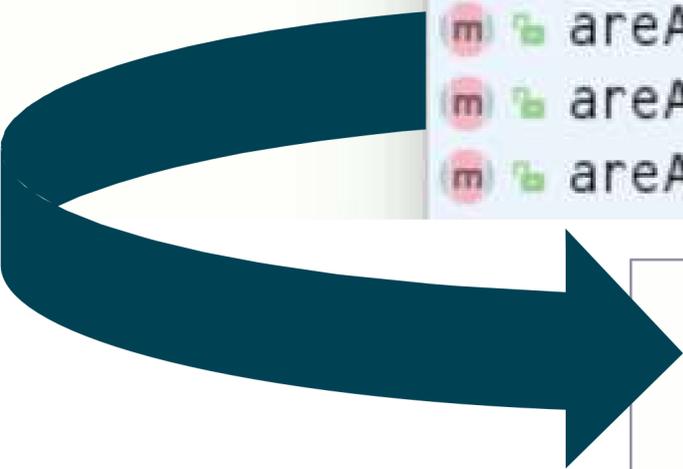
- ✓ Erstellung von Architekturregeln mittels eigener domänenspezifischer Sprache (DSL)
- ✓ Prüfung von Entwurfsregeln zur Testzeit

```
public class MyArchitectureTest {  
  
    @Test public void some_architecture_rule() {  
  
        JavaClasses importedClasses = new ClassFileImporter().importPackages("com.myapp");  
        ArchRule rule = classes()...; }  
  
}
```

ArchUnit Code-/API-Beispiel

```
classes().that().|
```

```
m areAnnotatedWith(String annotationTypeNa... GivenClassesConjunction  
m areAnnotatedWith(Class<? extends Annotation> annotationType) GivenClassesConjunc  
m areAnnotatedWith(DescribedPredicate<? su... GivenClassesConjunction  
m haveNameMatching(String regex) GivenClassesConjunction  
m areNamed(String name) GivenClassesConjunction  
m areNotAnnotatedWith(String annotationTyp... GivenClassesConjunction  
m areNotAnnotatedWith(Class<? extends Anno... GivenClassesConjunction  
m areNotAnnotatedWith(DescribedPredicate<?... GivenClassesConjunction  
m resideInAPackage(String packageIdentifie... GivenClassesConjunction  
m areAssignableFrom(Class<?> type) GivenClassesConjunction  
m areAssignableFrom(String typeName) GivenClassesConjunction  
m areAssignableFrom(DescribedPredicate<? s... GivenClassesConjunction
```

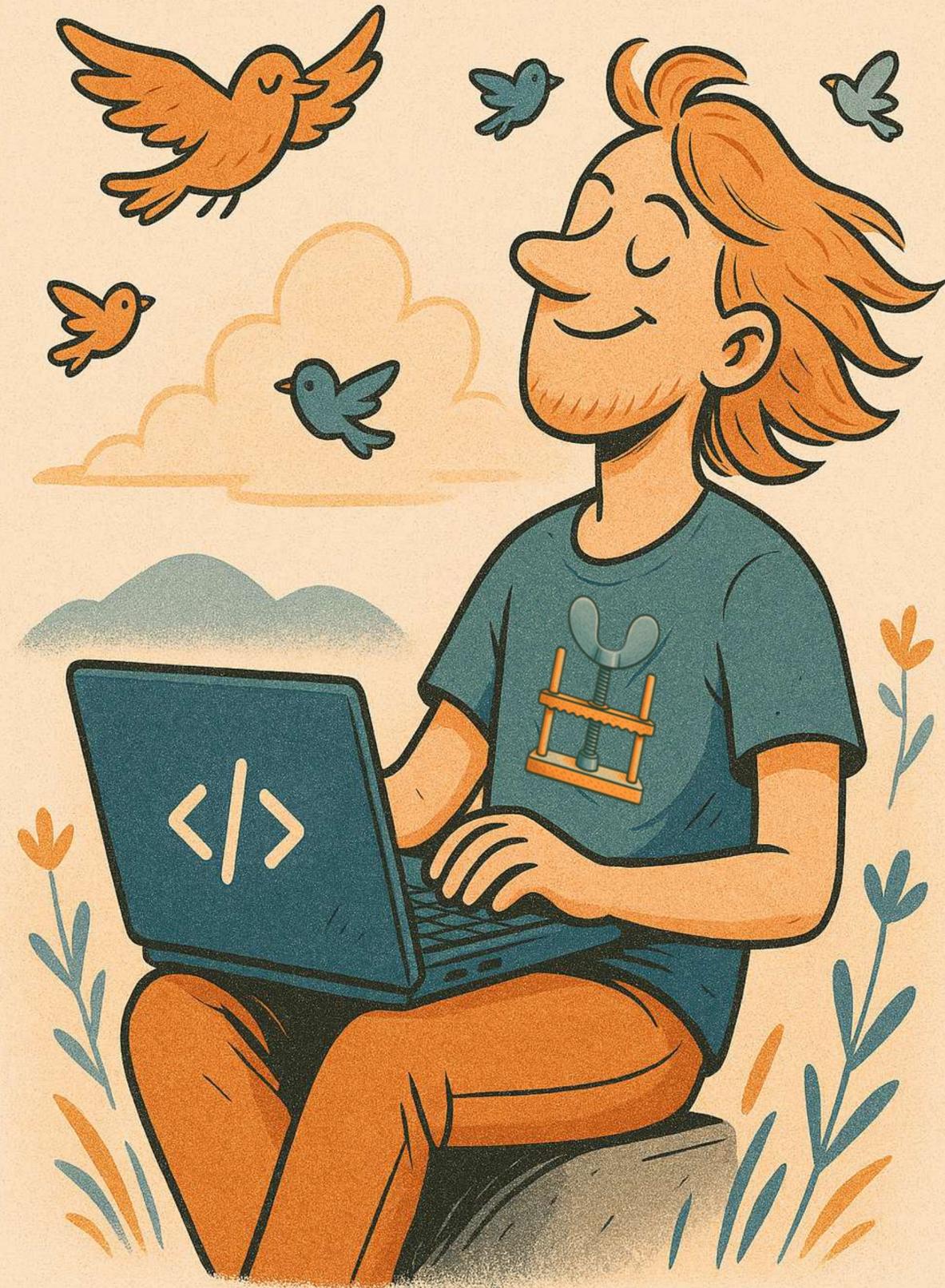


```
classes().that().areAnnotatedWith(Service.class)  
    .or().haveNameMatching(regex: ".*Service")  
    .should().resideInAPackage(packagelidentifier: "..service..")  
    .orShould().beAnnotatedWith(LegacyBridge.class);
```

Fazit

Was bedeutet das für uns?

Geschicktes Daumenschrauben muss nicht nerven!



Bildnachweise

- Bitmap-Grafiken wurden mit OpenAI GPT 4o (March 25, 2025 Release) erzeugt
- Vektorgrafiken kommen von Streamline

Danke!

Fragen? Diskussion!

INNOQ

UNSER ANGEBOT

Produktkonzeption & Design
Software-Entwicklung & -Architektur
Technologie-Beratung
Infrastruktur & Betrieb
Wissenstransfer, Coaching & Trainings

FAKTEN

~160 Mitarbeitende
1998 gegründet
9 Standorte in
D & CH

FOKUS

Webapplikationen
SaaS
IoT
Produktentwicklung
ML/AI
Blockchain

TECHNOLOGIEN

(Auswahl)

Java/Spring	JavaScript
Ruby/Rails	Python
Scala	C#
AWS	ML/AI
Kubernetes	Blockchain
Azure	

KLIENTEN

Finance ● Telko ● Logistik ● E-Commerce ● Fortune 500 ● KMUs ● Startups



71

