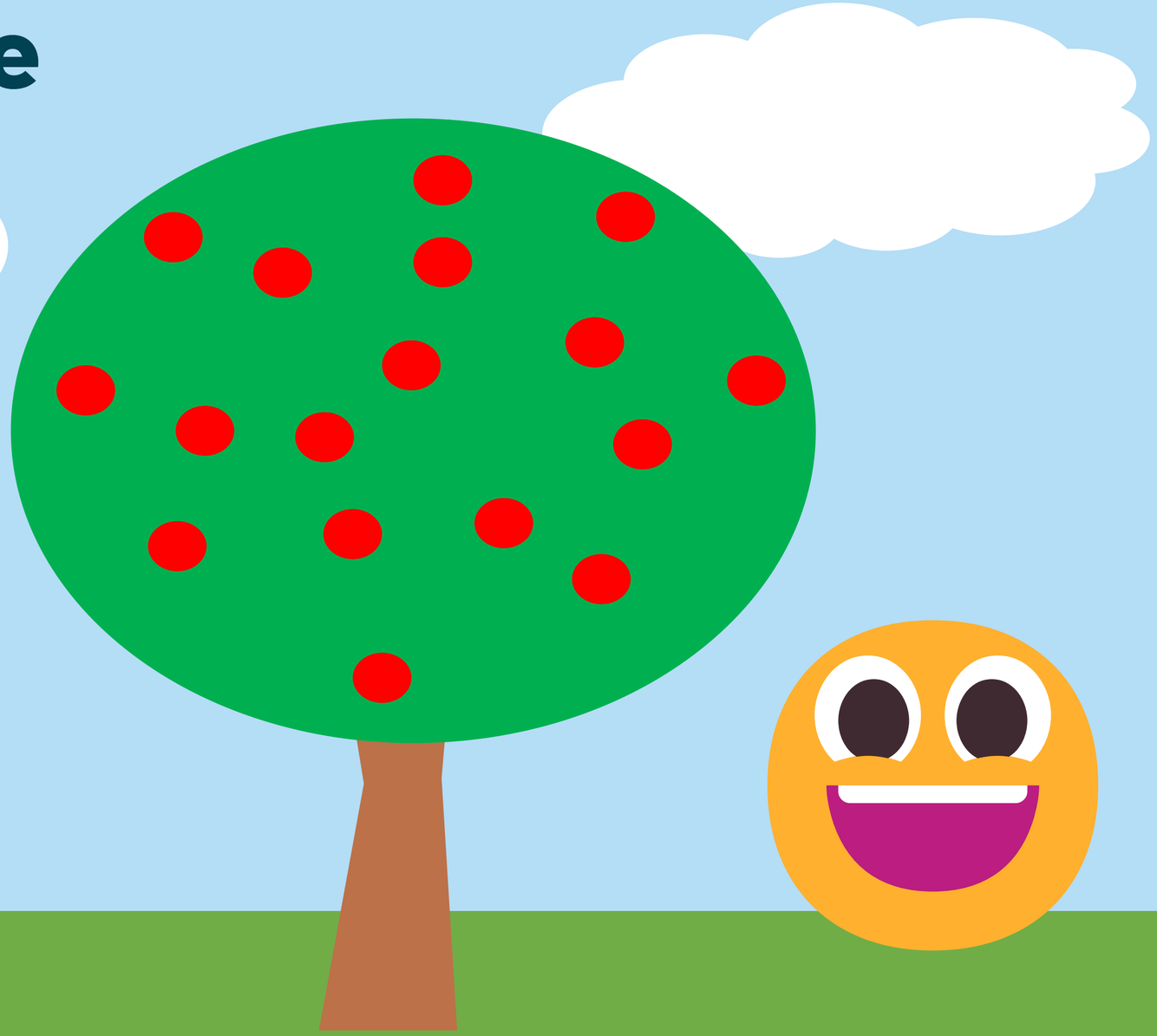
Softwaresysteme wie Apfelbäume pflegen

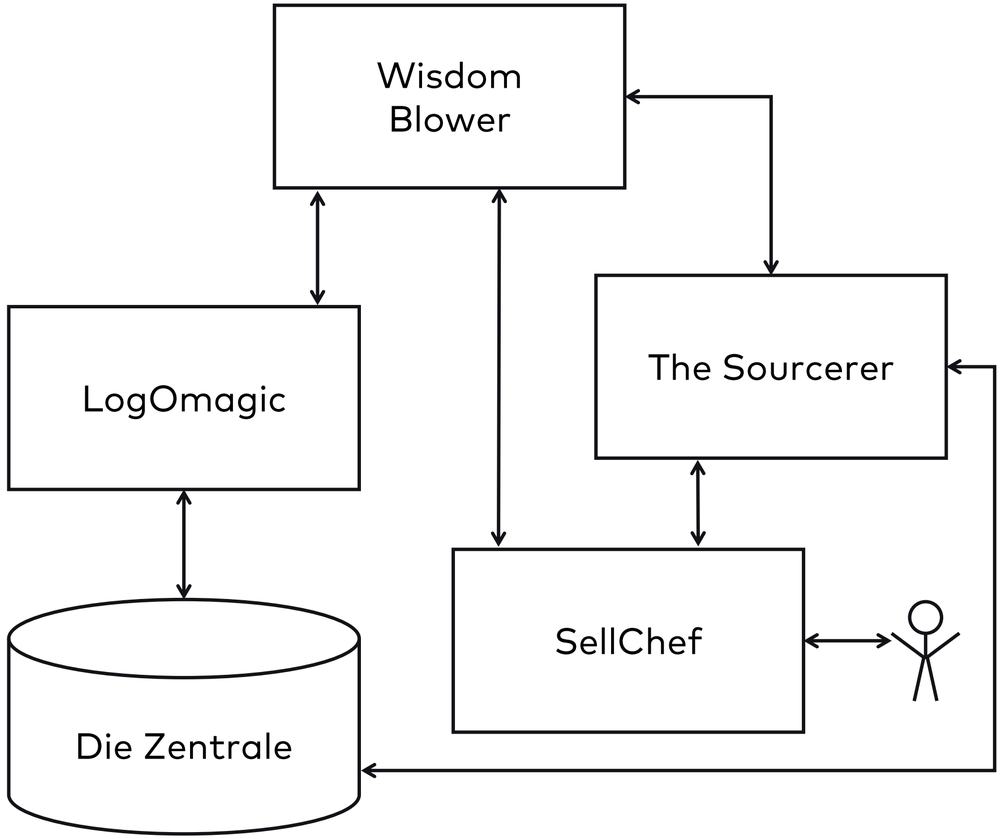


Über die Modernisierung von Softwaresystemen

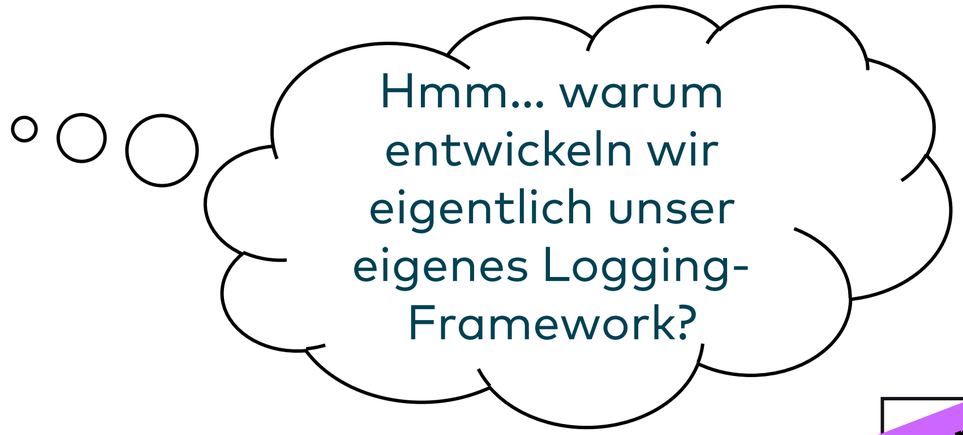
Markus Harrer

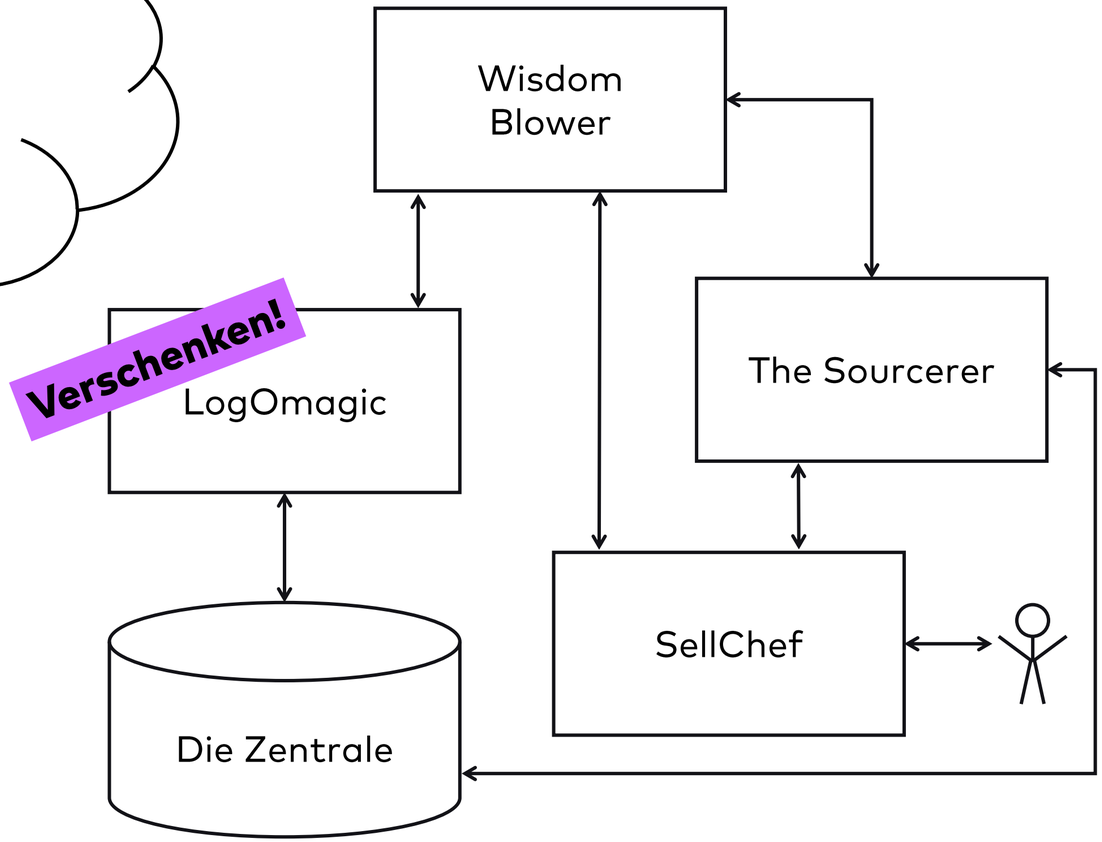
INNOQ

Typische Situation



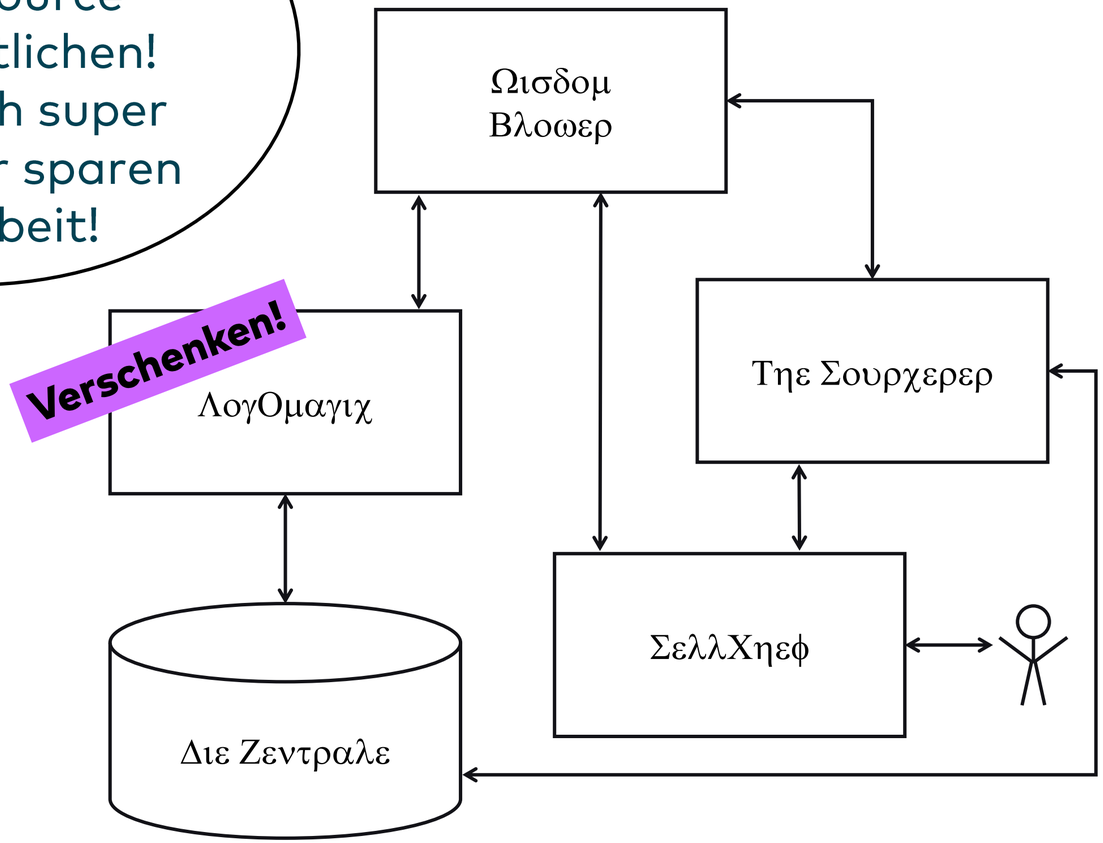
Typische Situation

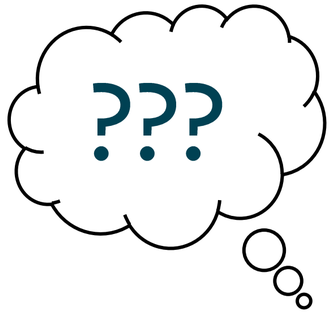


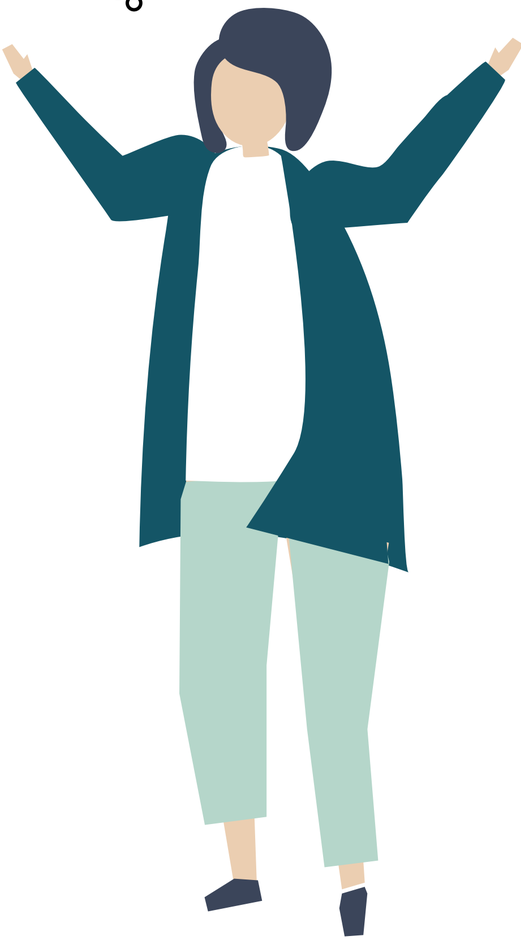


Typische Situation

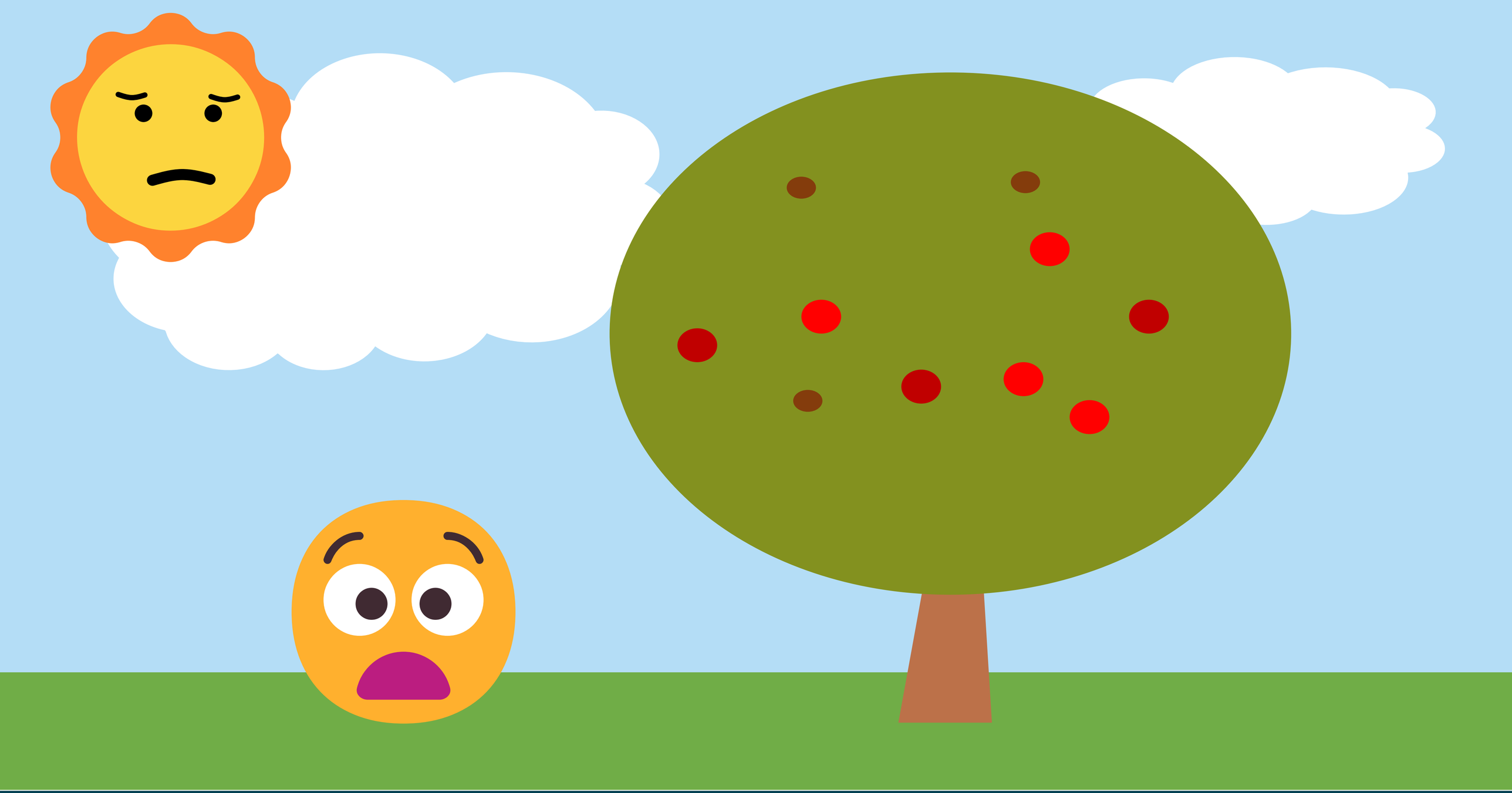
Wir sollten ΛογΟμαγιχ als Open Source veröffentlichen! Bietet sich super an und wir sparen uns Arbeit!

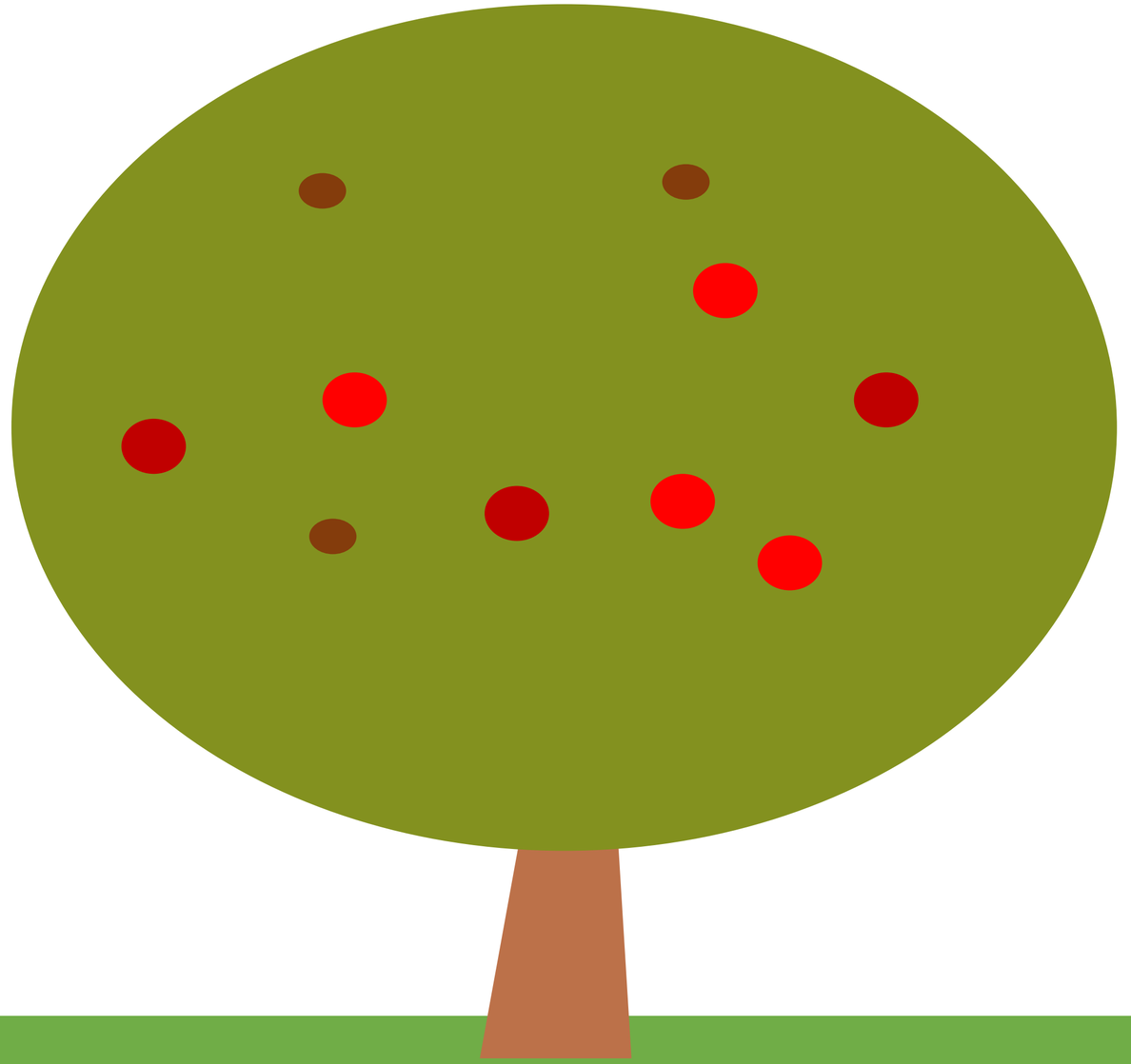






Es war einmal ...





zurechtstutzen?

ignorieren?

auslichten?

umwidmen?

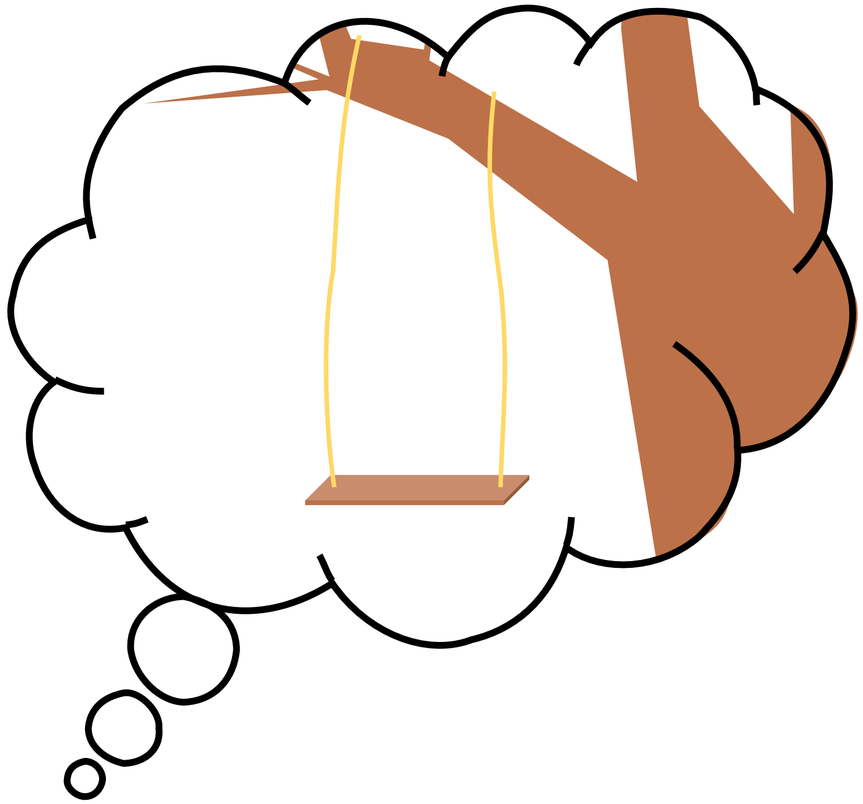
veredeln?

verbessern?

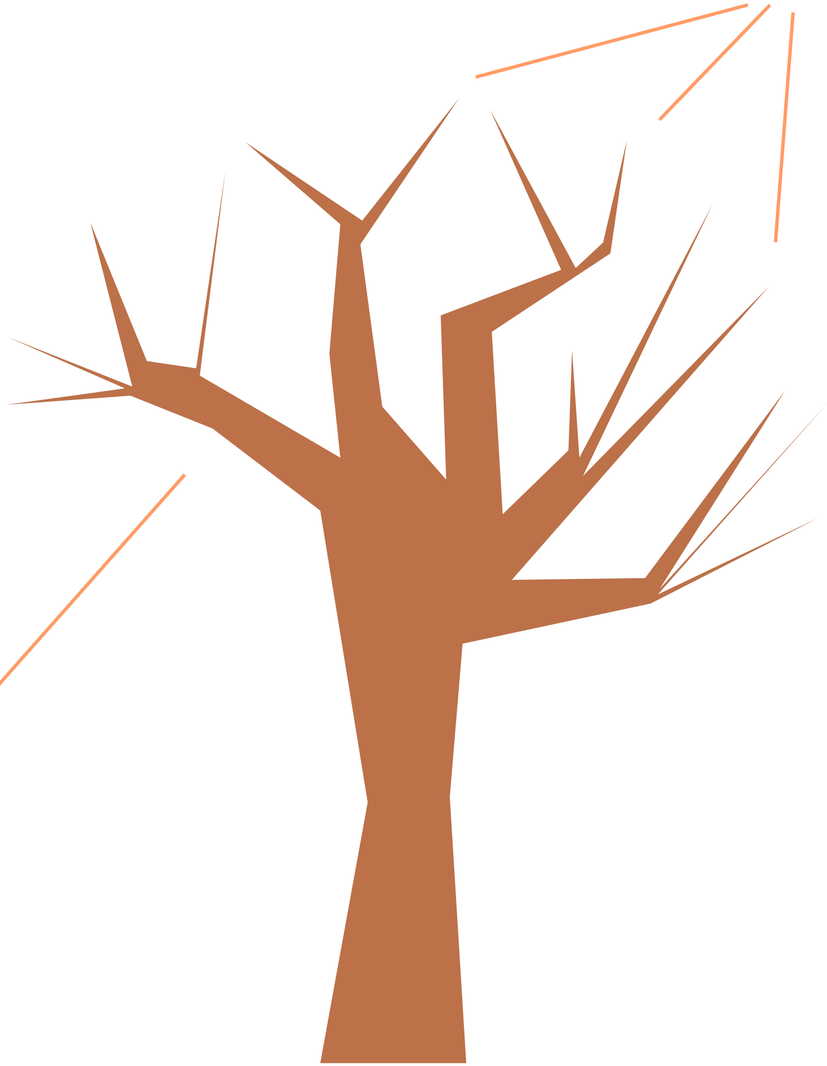
abschneiden?

ersetzen?

verpflanzen?

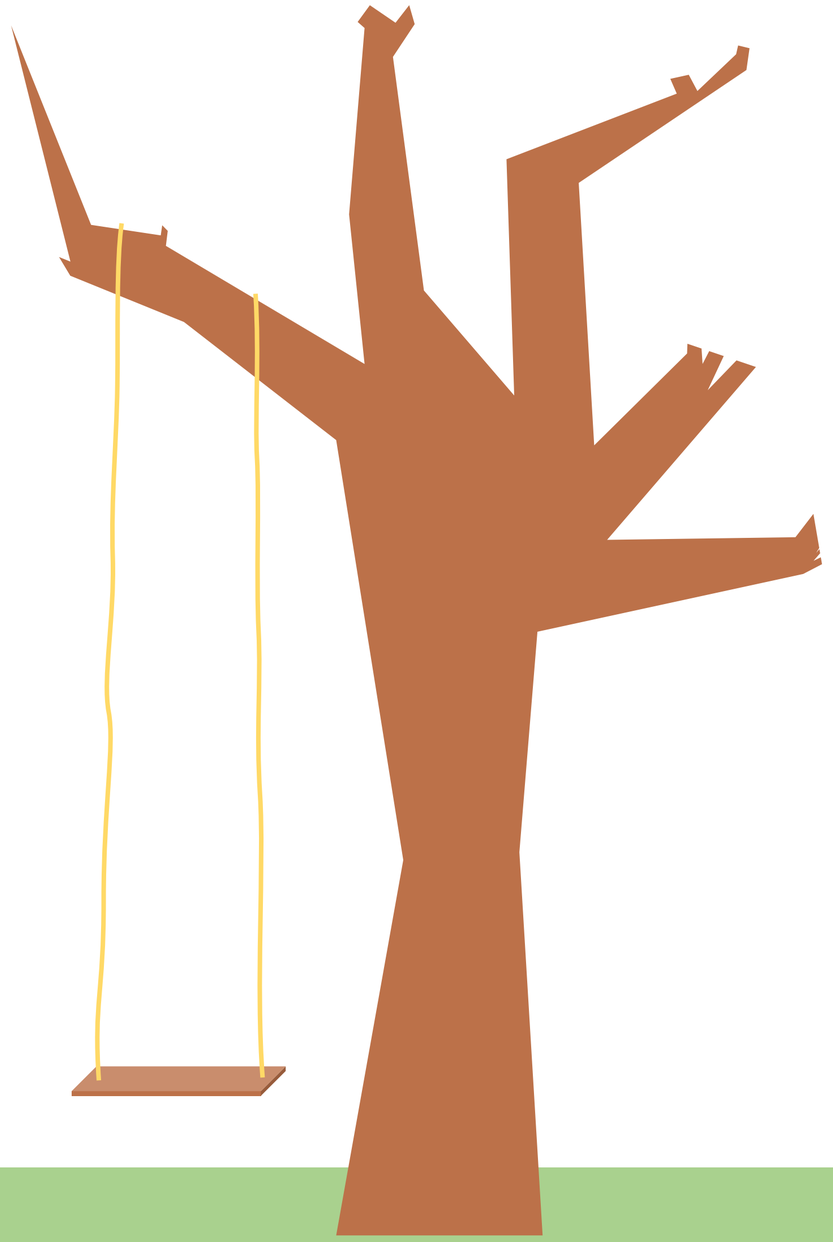


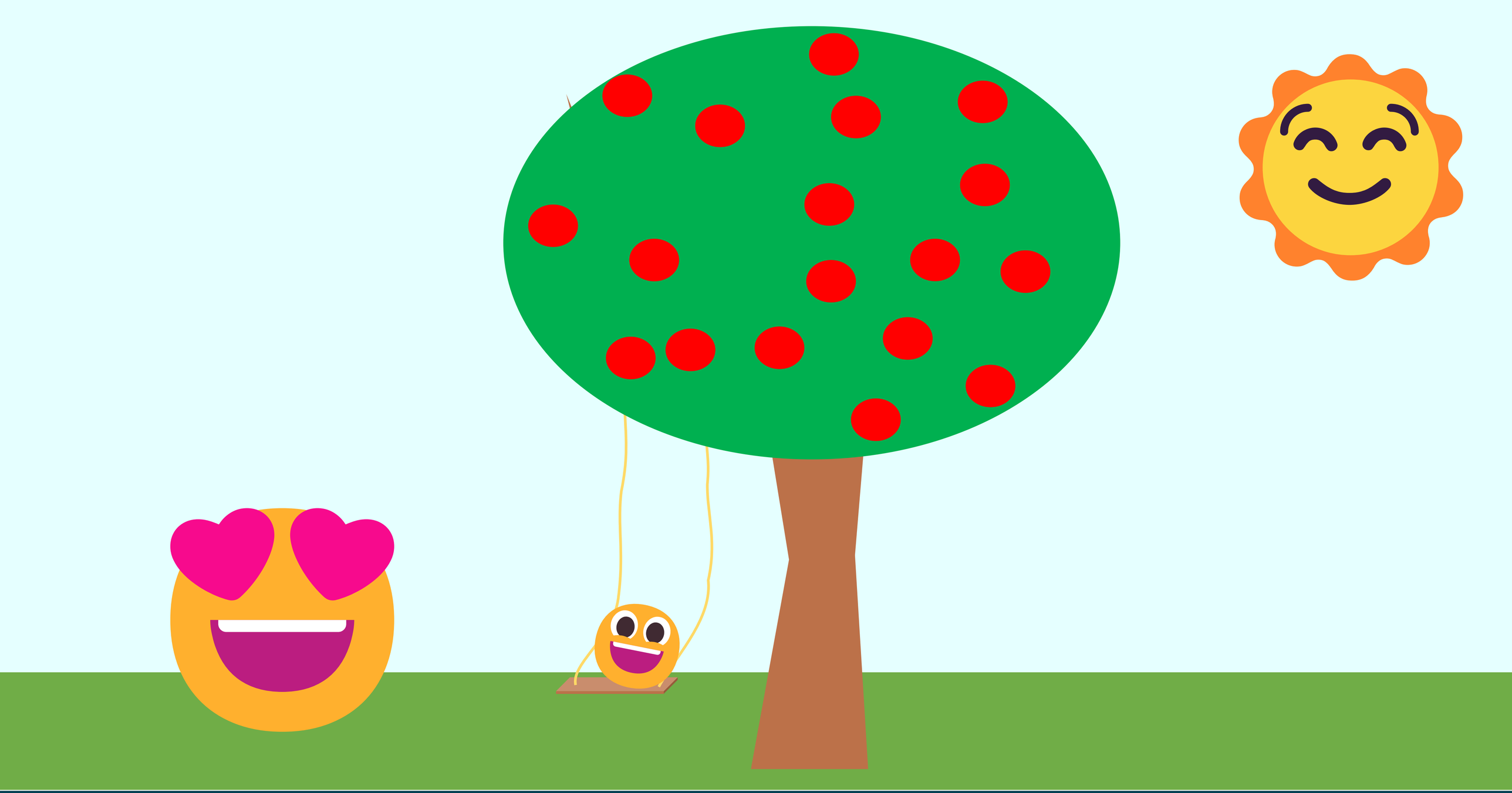
1) auslichten!



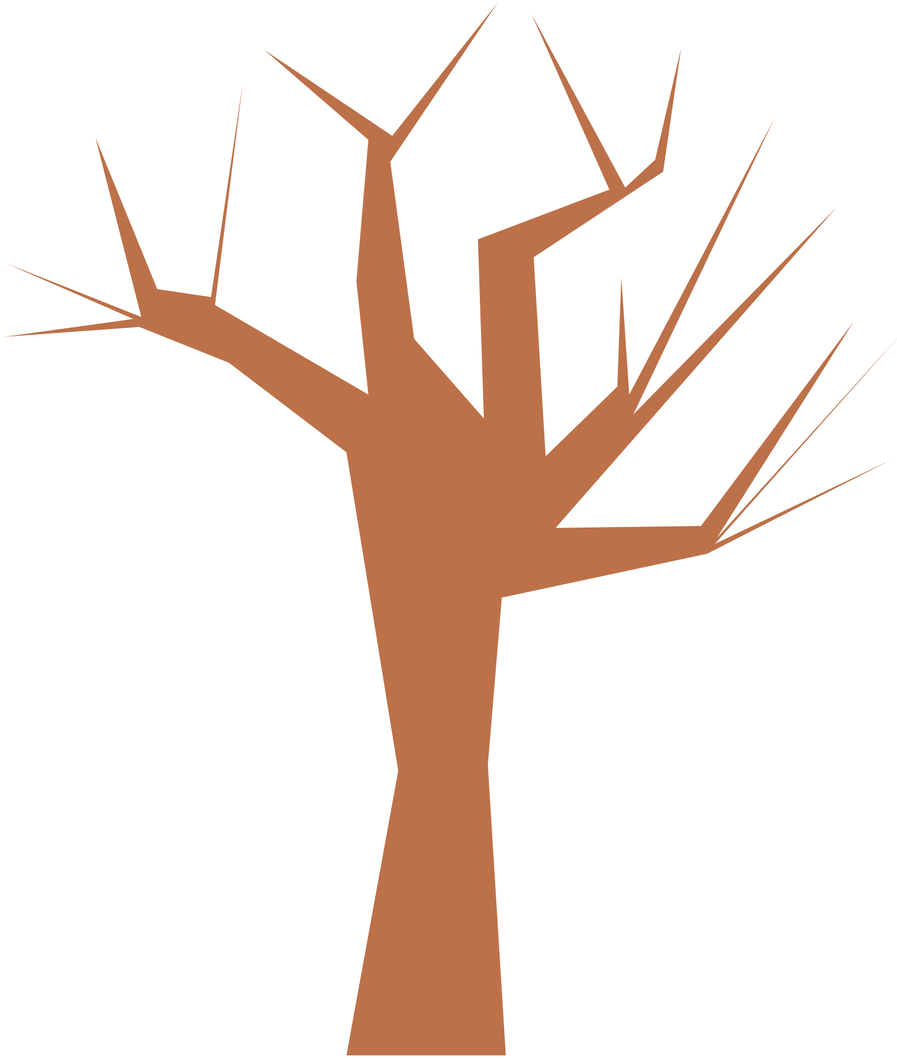
2) umwidmen!

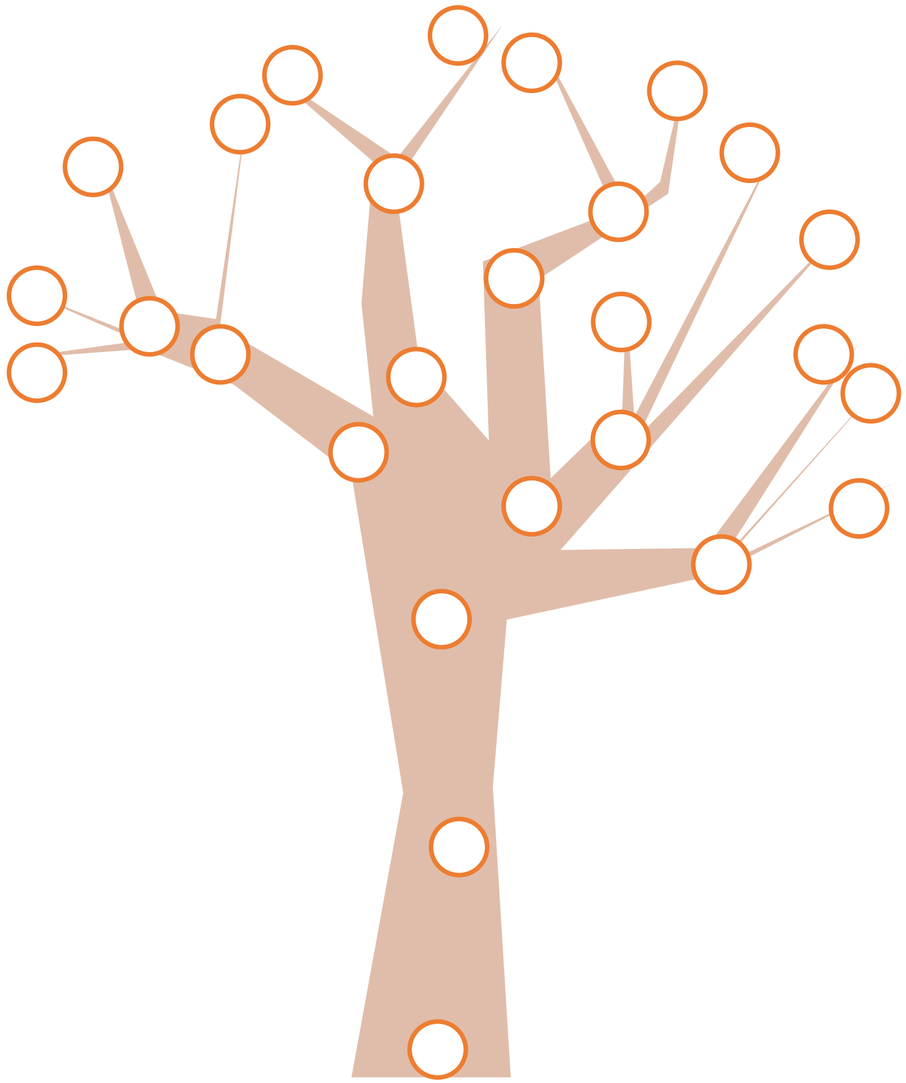

A FEW MOMENTS LATER

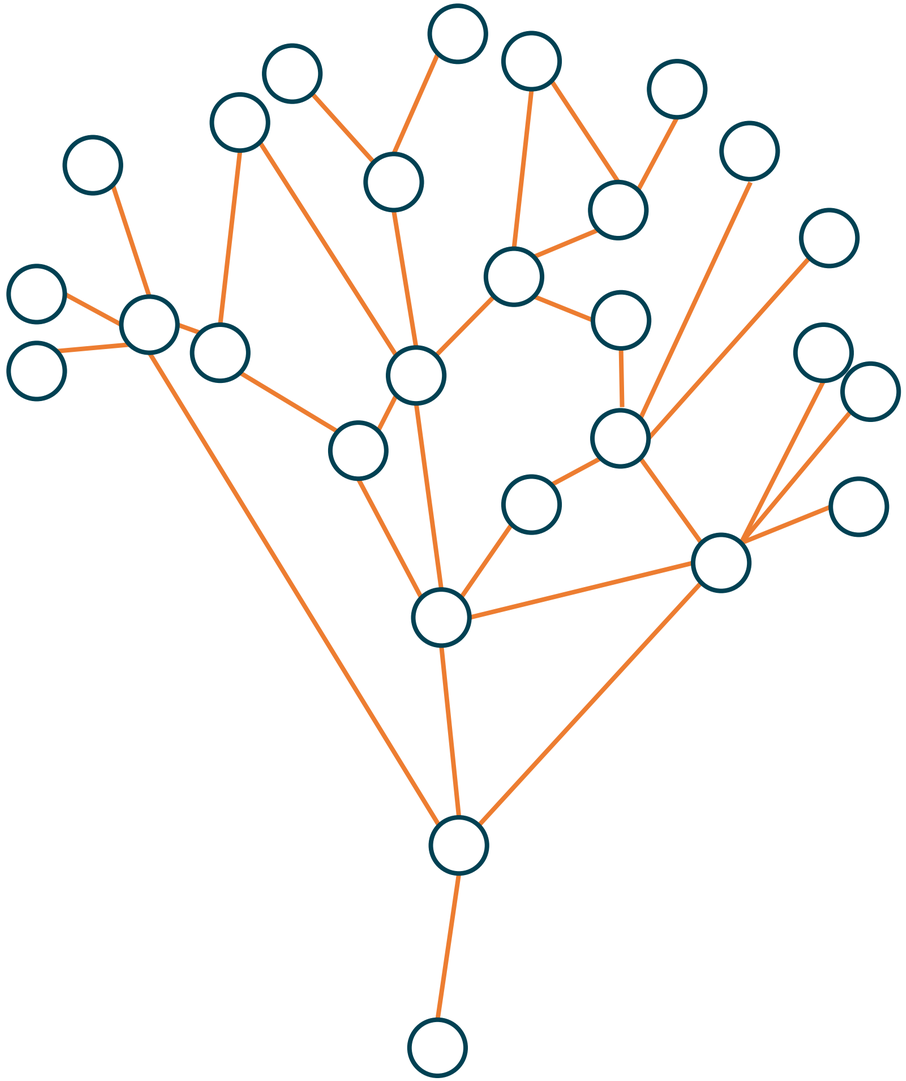




So what?

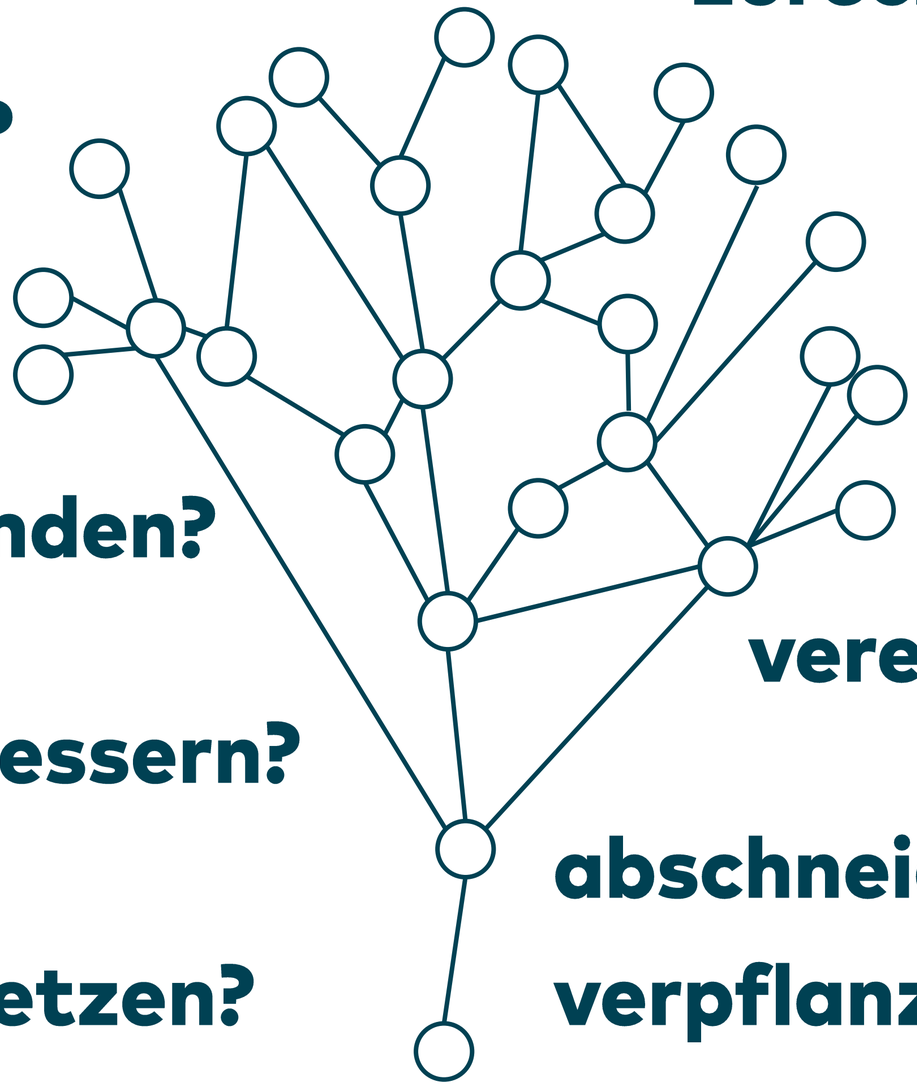






zurechtstutzen?

ignorieren?



auslichten?

zweckentfremden?

veredeln?

verbessern?

abschneiden?

ersetzen?

verpflanzen?

Was kann ich hier in der Softwareentwicklung tun?

Business

Ignorieren!

Abschalten!

Am Leben erhalten!

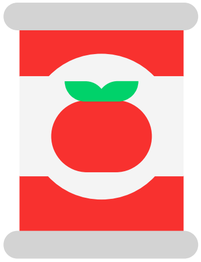
Verkaufen!

Outsourcen!

Verschenken!

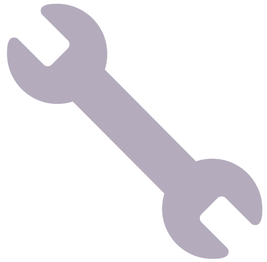
Produkt

Neukaufen!



Konservieren!

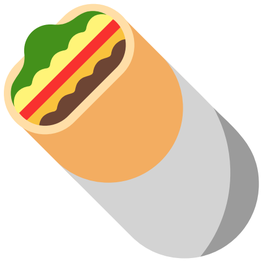
Umwidmen!



Ausschlachten!

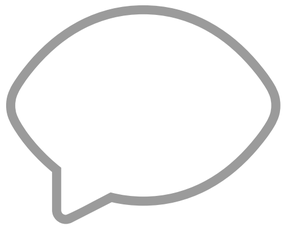
Zurechtstutzen!

Technik



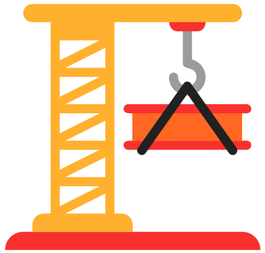
Umhüllen!

Ersetzen!

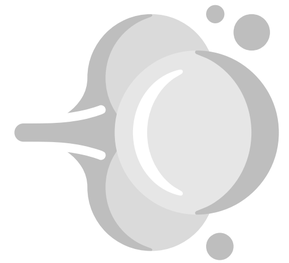


Konvertieren!

Verbessern!

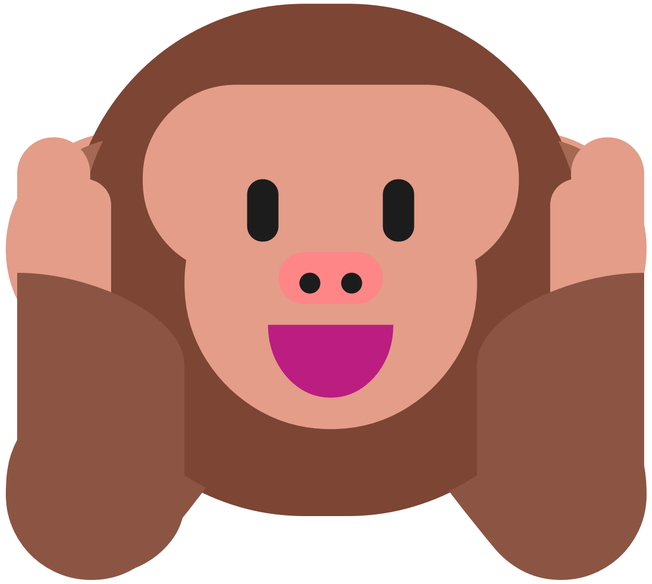


Replatform!



Rehost!

Ignorieren!

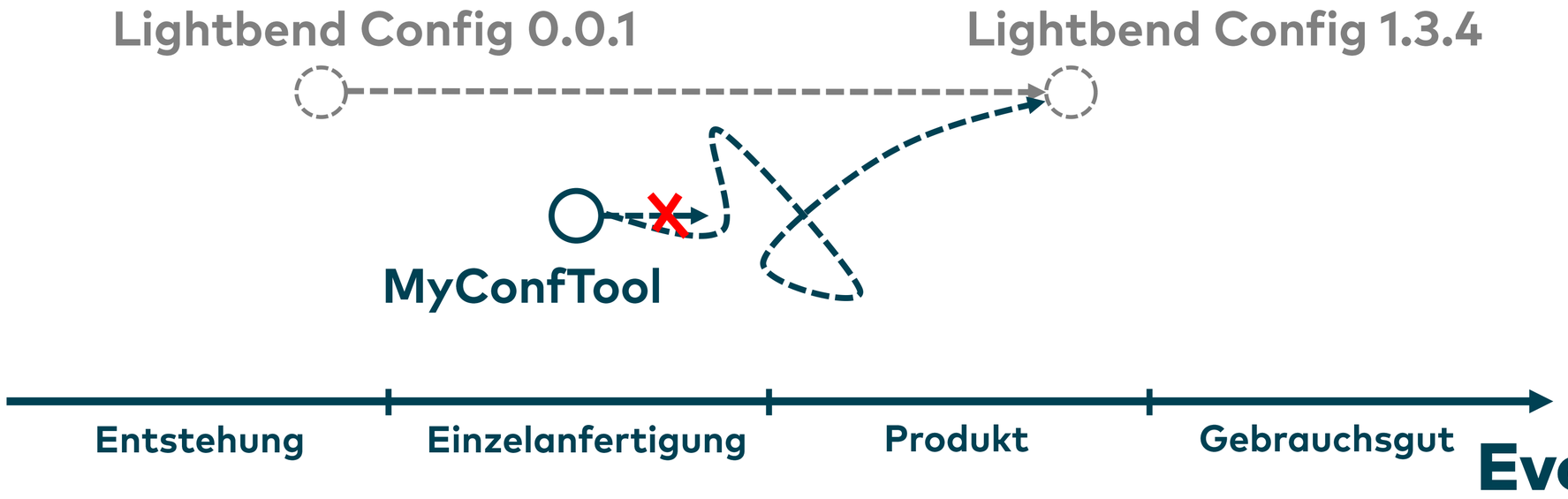


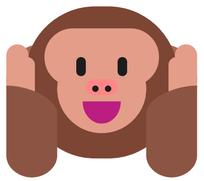
~~**Ignorieren!**~~

Verzögern!

Prokrastinieren!

Abwarten!

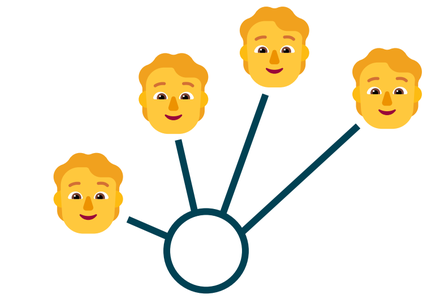


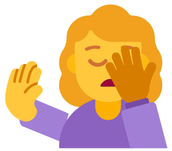


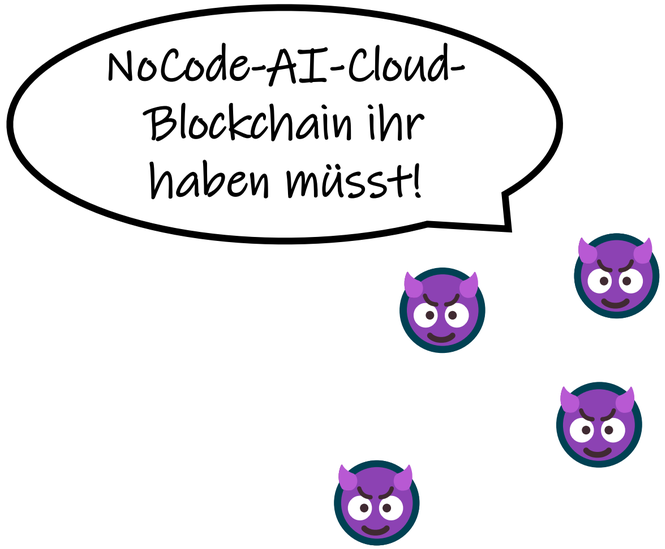
Abwarten!

+

Outsourcen!







Verzögern!

z. B. über Bildung

Business

Ignorieren!

Abschalten!

Am Leben erhalten!

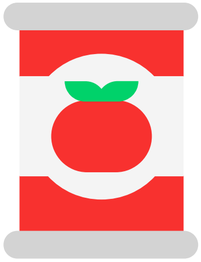
Verkaufen!

Outsourcen!

Verschenken!

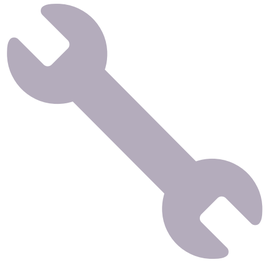
Produkt

Neukaufen!



Konservieren!

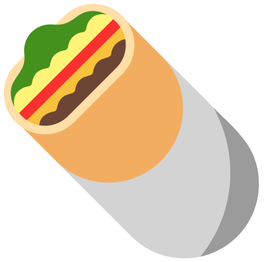
Umwidmen!



Ausschlachten!

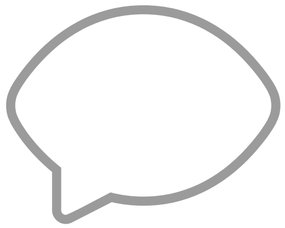
Zurechtstutzen!

Technik



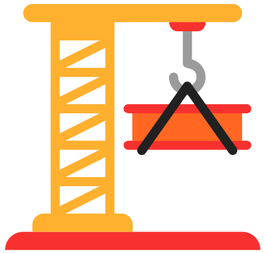
Umhüllen!

Ersetzen!

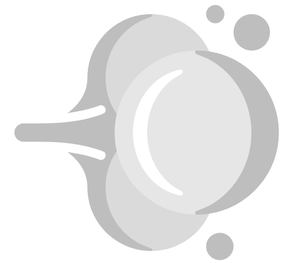


Konvertieren!

Verbessern!



Replatform!



Rehost!

Sunsetting

Abschalten!

+ behalten:

- **Methodik**
- **Patente**
- **Standards**
- **…**

„Je weniger du tust, desto mehr davon kannst du machen!“

Scott Hanselman

Business

Ignorieren!

Abschalten!

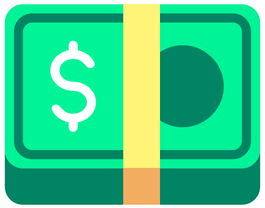
Am Leben erhalten!

Verkaufen!

Outsourcen!

Verschenken!

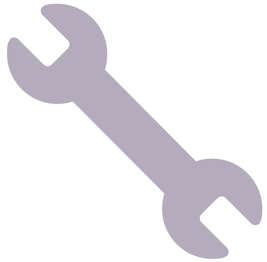
Produkt



Neukaufen!

Konservieren!

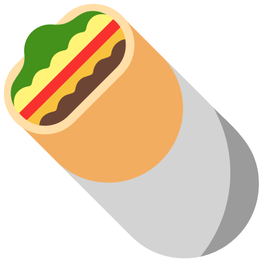
Umwidmen!



Ausschlachten!

Zurechtstutzen!

Technik

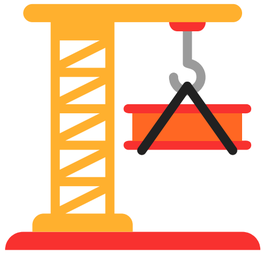


Umhüllen!

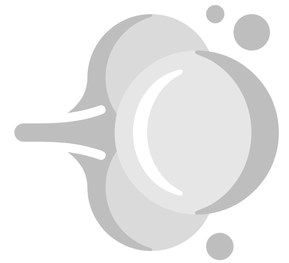
Ersetzen!

Konvertieren!

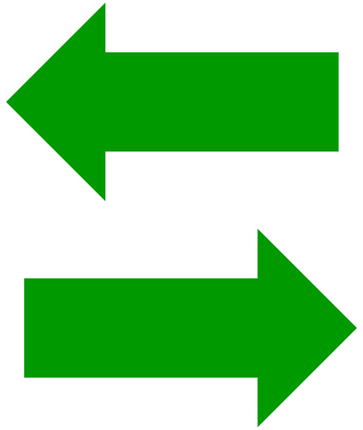
Verbessern!



Replatform!

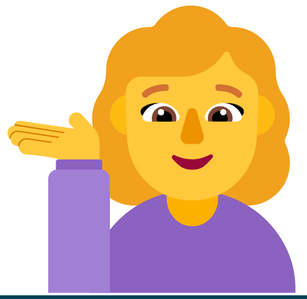


Rehost!



Am Leben erhalten!

Verkaufen!



Outsourcen!

Verschenken!

Business

Ignorieren!

Abschalten!

Am Leben erhalten!

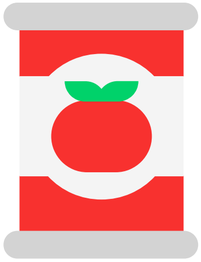
Verkaufen!

Outsourcen!

Verschenken!

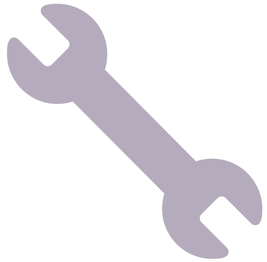
Produkt

Neukaufen!



Konservieren!

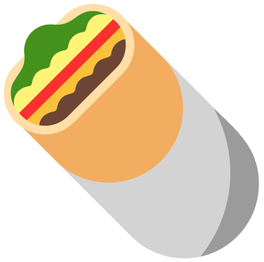
Umwidmen!



Ausschlachten!

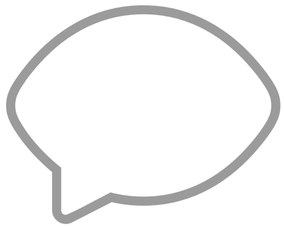
Zurechtstutzen!

Technik



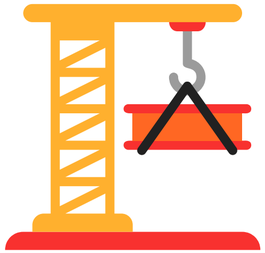
Umhüllen!

Ersetzen!

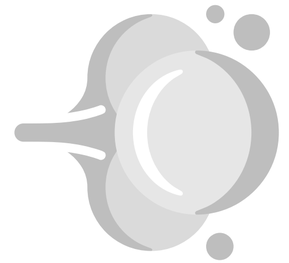


Konvertieren!

Verbessern!



Replatform!



Rehost!

Konservieren!

haltbar machen durch:

- **Virtualisierung**
- **Backups**
- **Datenexporte**
- **Archivierung**
- **Screen-Recordings**
- **…**

Business

Ignorieren!

Abschalten!

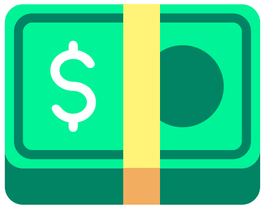
Am Leben erhalten!

Verkaufen!

Outsourcen!

Verschenken!

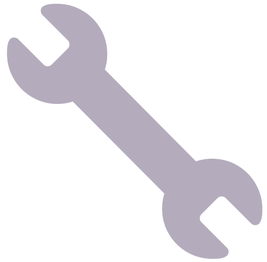
Produkt



Neukaufen!

Konservieren!

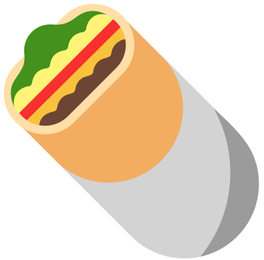
Umwidmen!



Ausschlachten!

Zurechtstutzen!

Technik

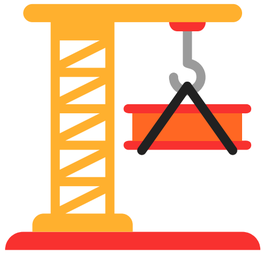


Umhüllen!

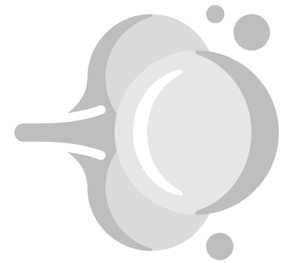
Ersetzen!

Konvertieren!

Verbessern!

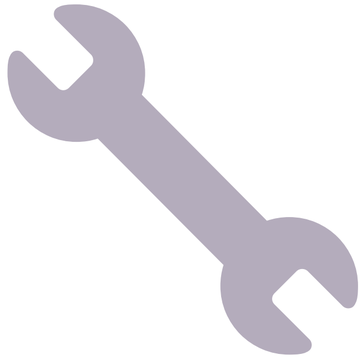


Replatform!



Rehost!

Umwidmen!



Ausschlachten!

Zurechtstutzen!

Business

Ignorieren!

Abschalten!

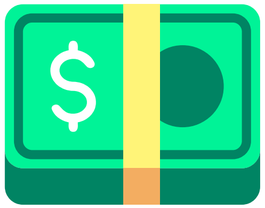
Am Leben erhalten!

Verkaufen!

Outsourcen!

Verschenken!

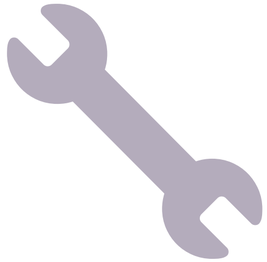
Produkt



Neukaufen!

Konservieren!

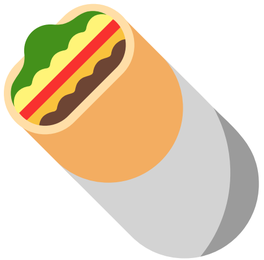
Umwidmen!



Ausschlachten!

Zurechtstutzen!

Technik

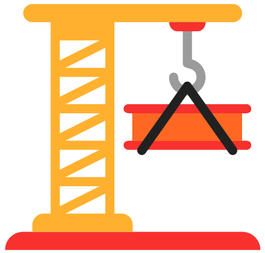


Umhüllen!

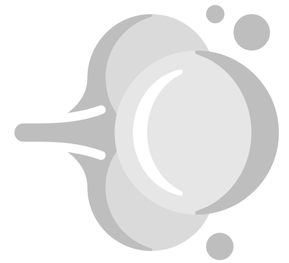
Ersetzen!

Konvertieren!

Verbessern!

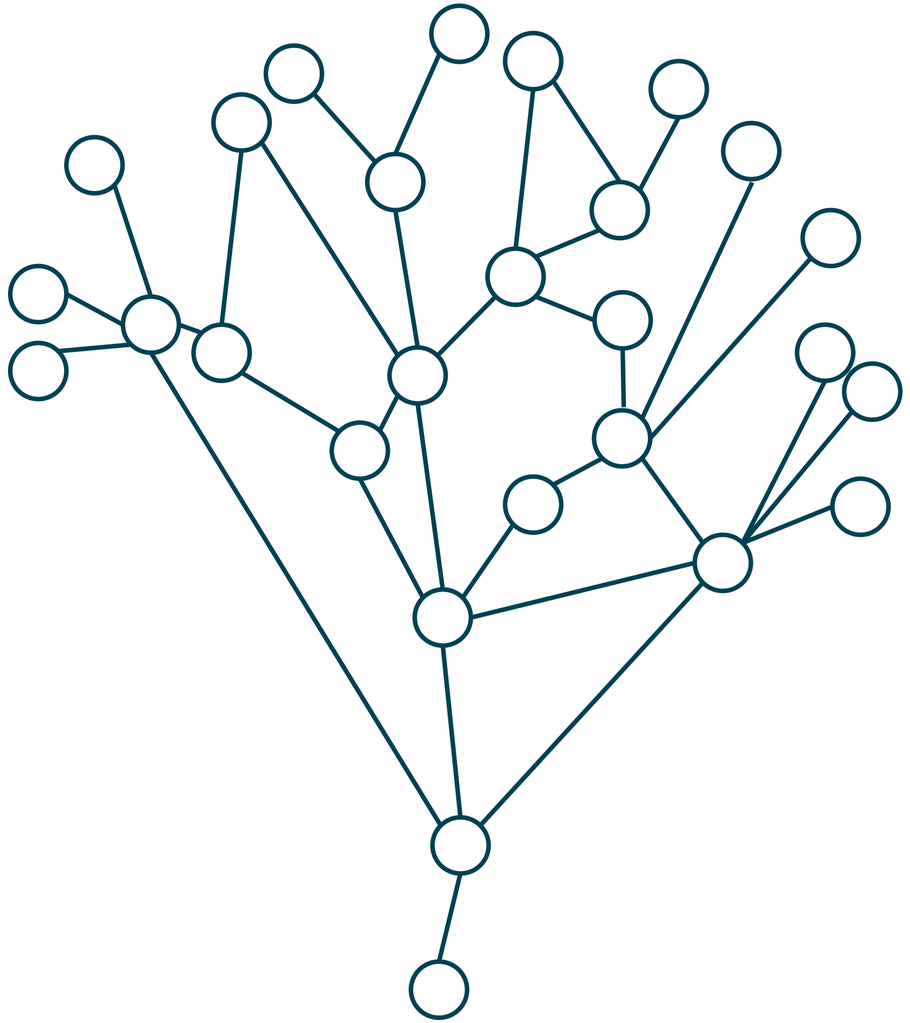


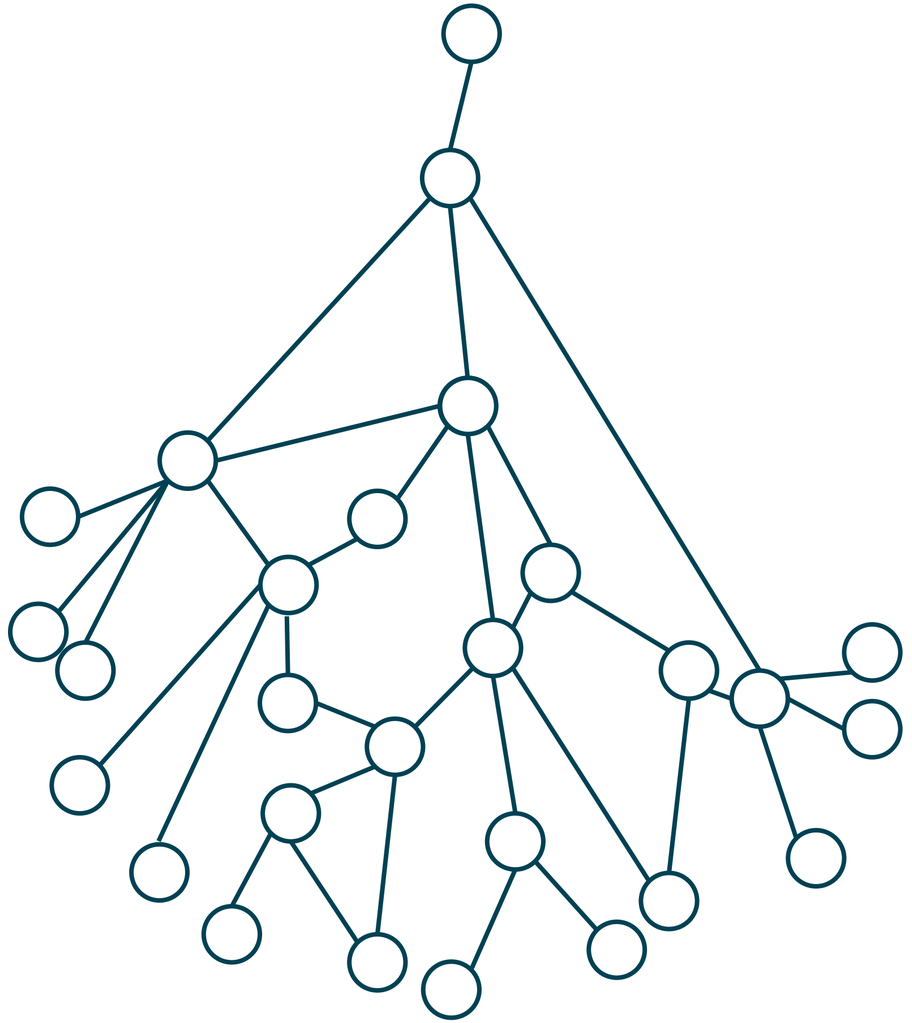
Replatform!



Rehost!

Was kann ich hier in der Softwareentwicklung tun?





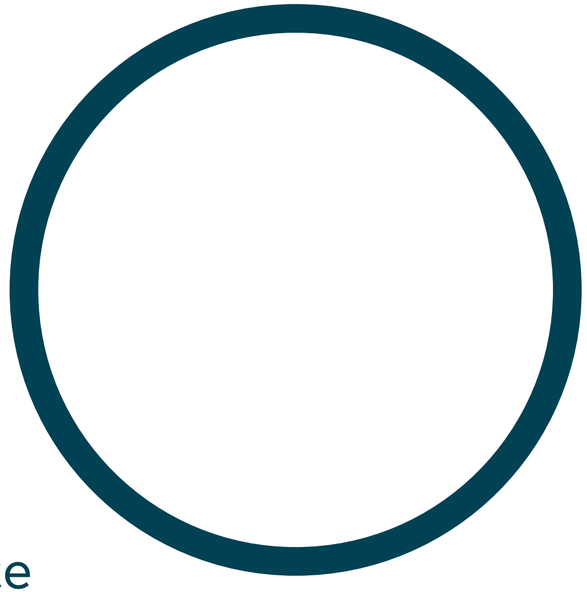
Komponente

„Ein Fragment der uns umgebenden Realität, das wir zum Zweck der weiteren Analyse (willkürlich) als eine einzige Einheit behandeln wollen.“

Chris Daniel, Mapping Glossary (https://community.wardleymaps.com/t/mapping-glossary/280)

Softwareentwicklung

- Bausteine
- Module
- Teilsysteme
- Services
- Softwaresysteme
- Infrastrukturelemente

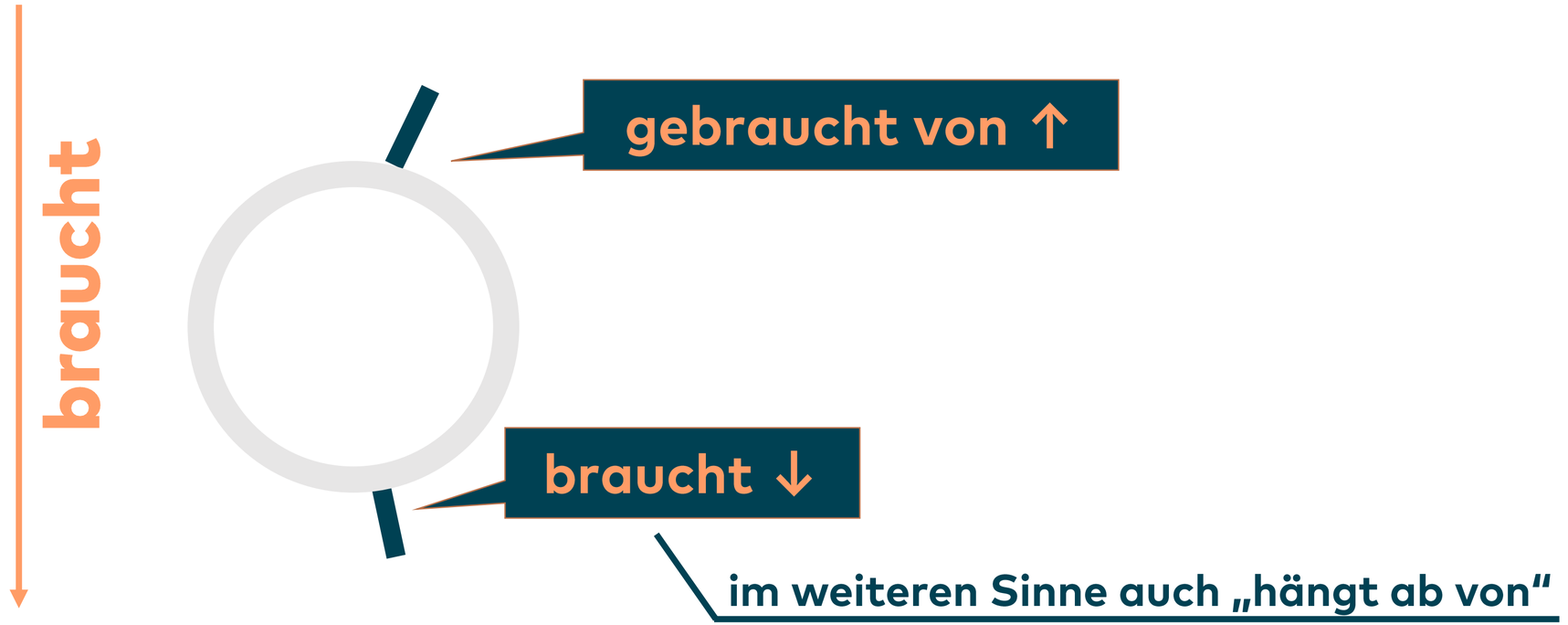


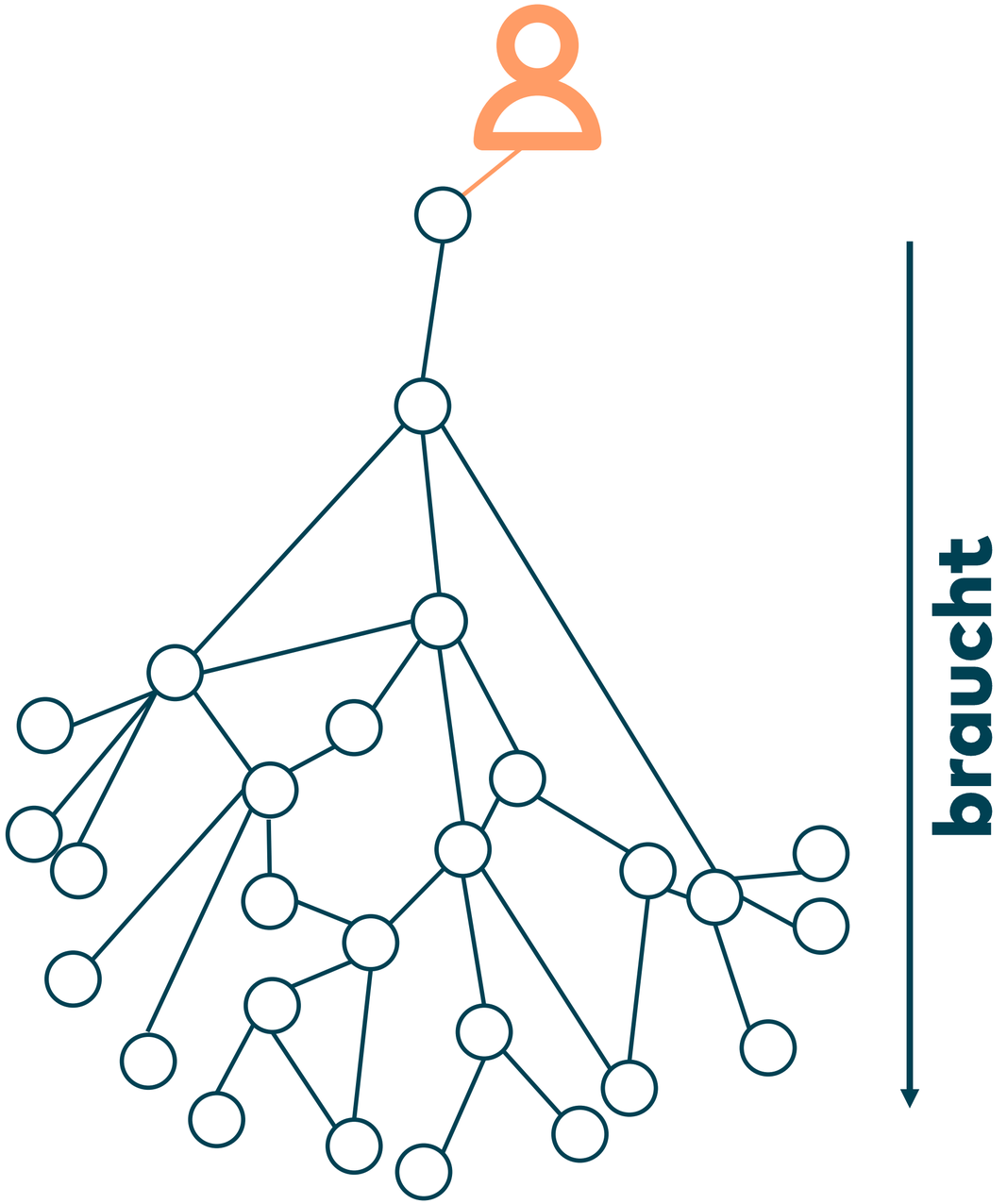
Allgemein

- Aktivitäten (Dinge, die wir tun)
- Praktiken (wie wir Dinge tun)
- Daten (wie wir etwas messen)
- Wissen (wie wir etwas verstehen)

Beziehung

Verbindung zwischen Komponenten




braucht

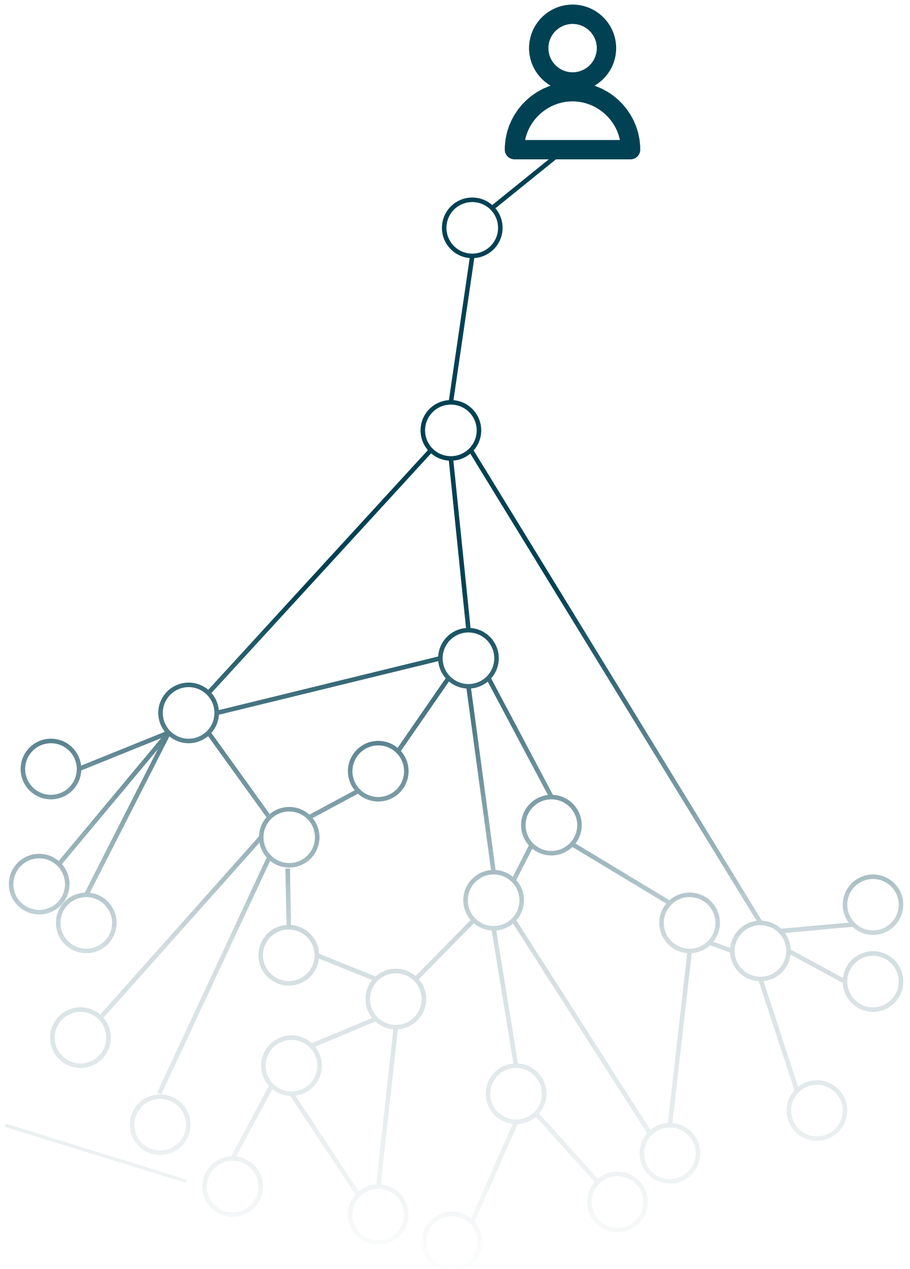

Wert


braucht

Wert

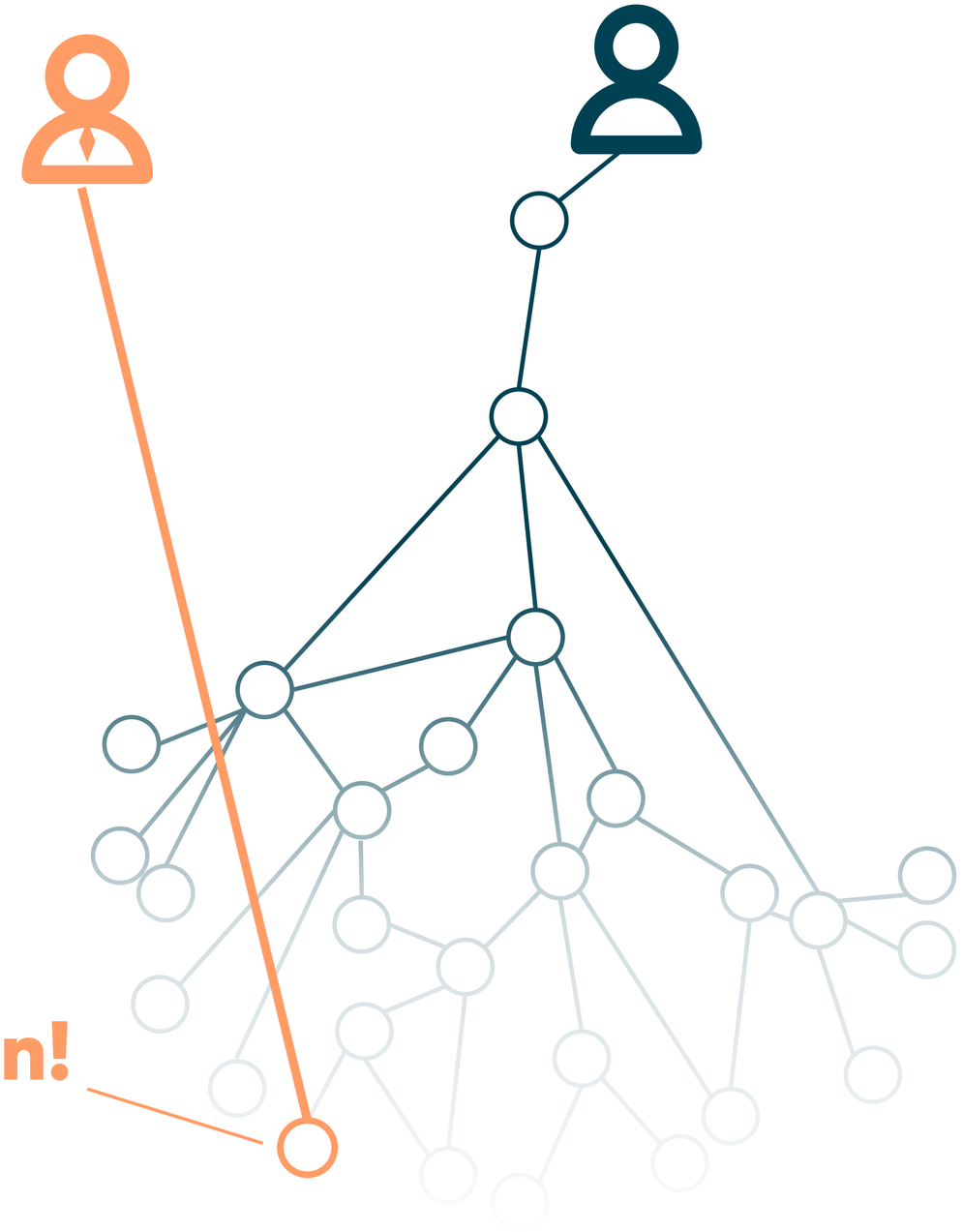
verbessern!

Wert



verschenken!

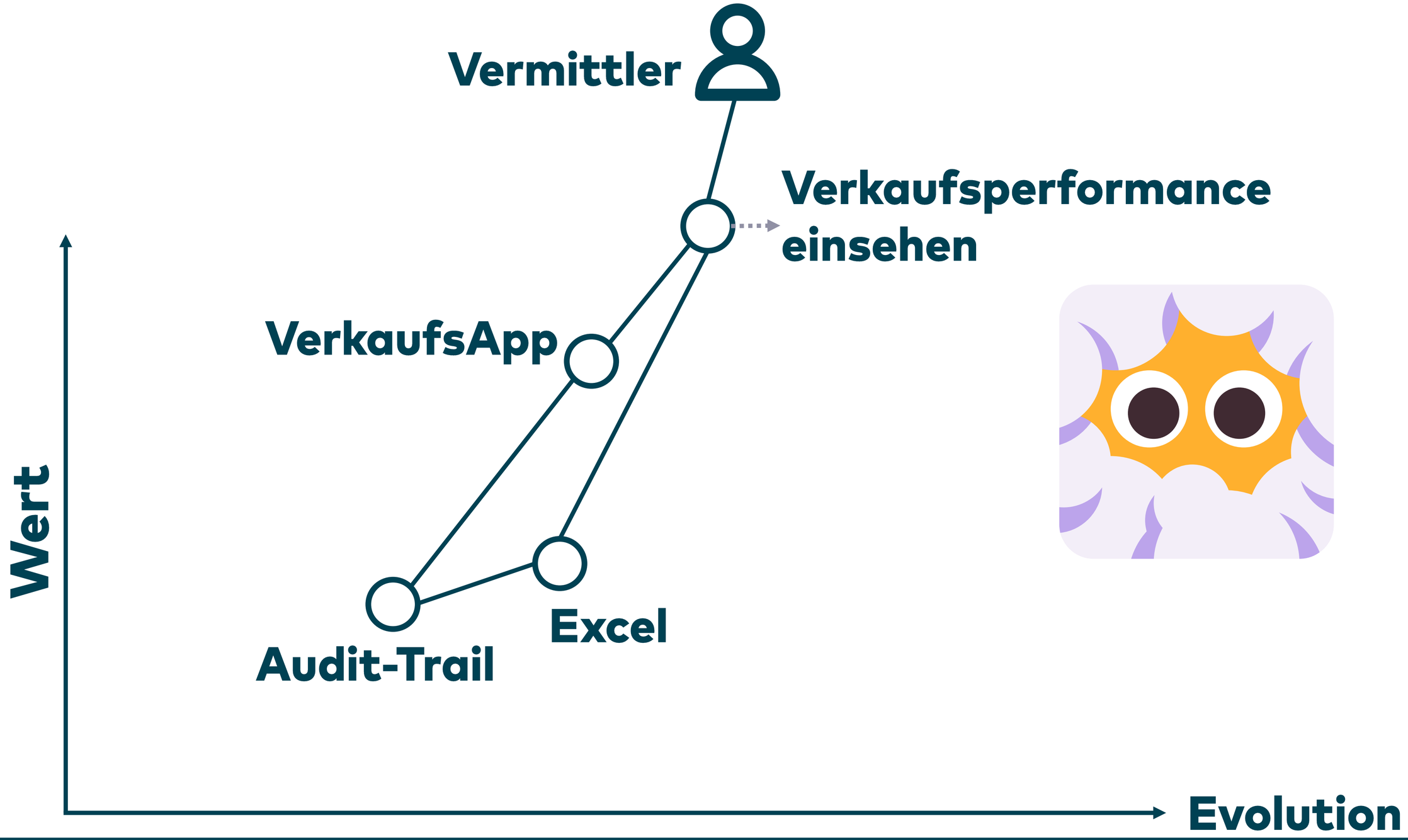
Wert

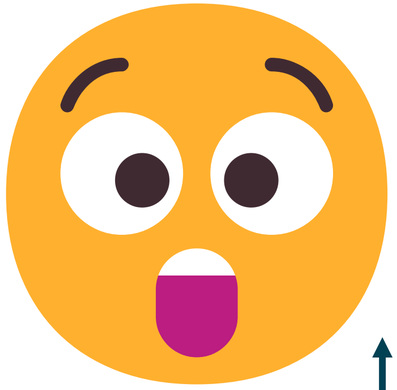


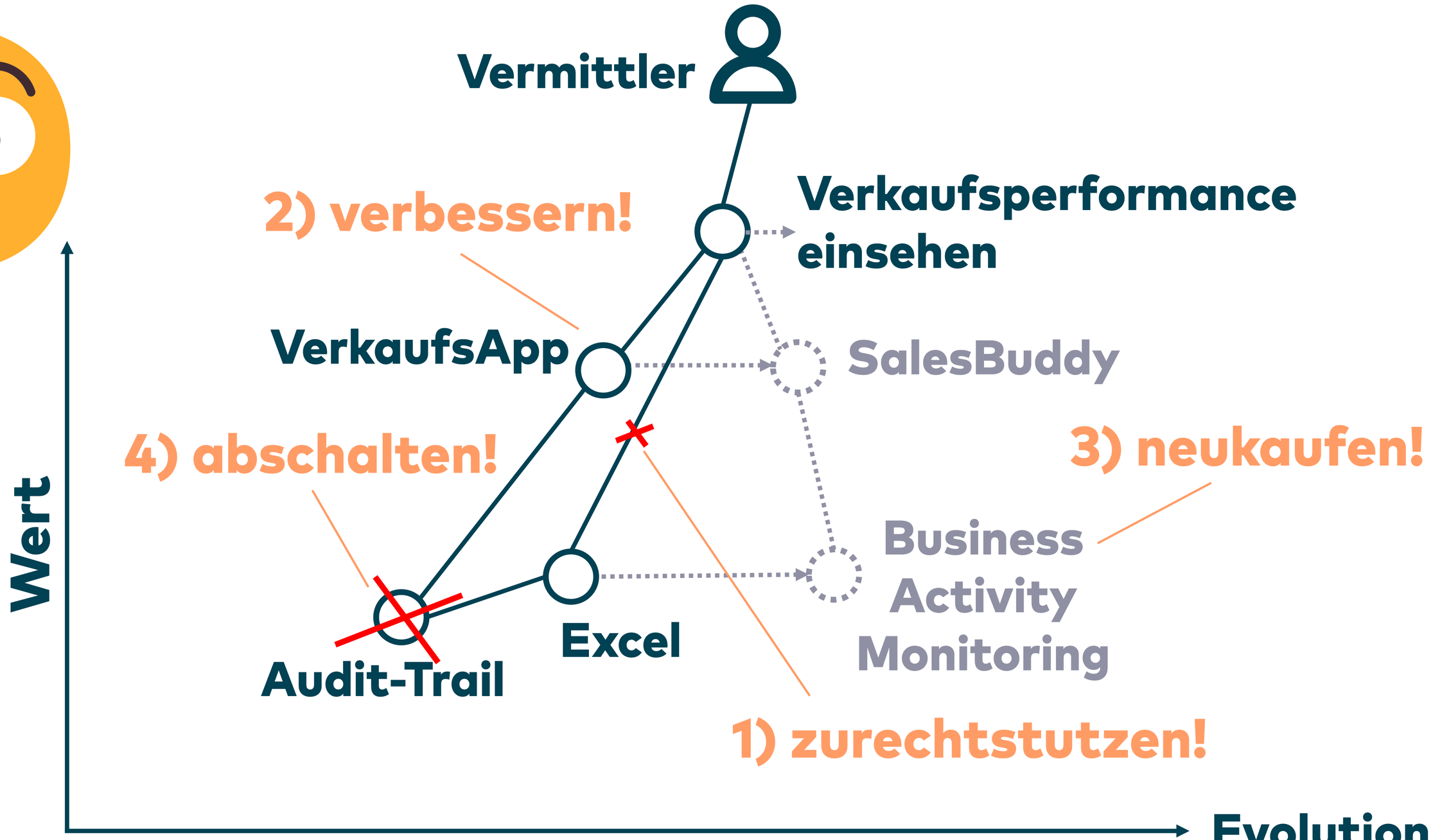
verschenken!

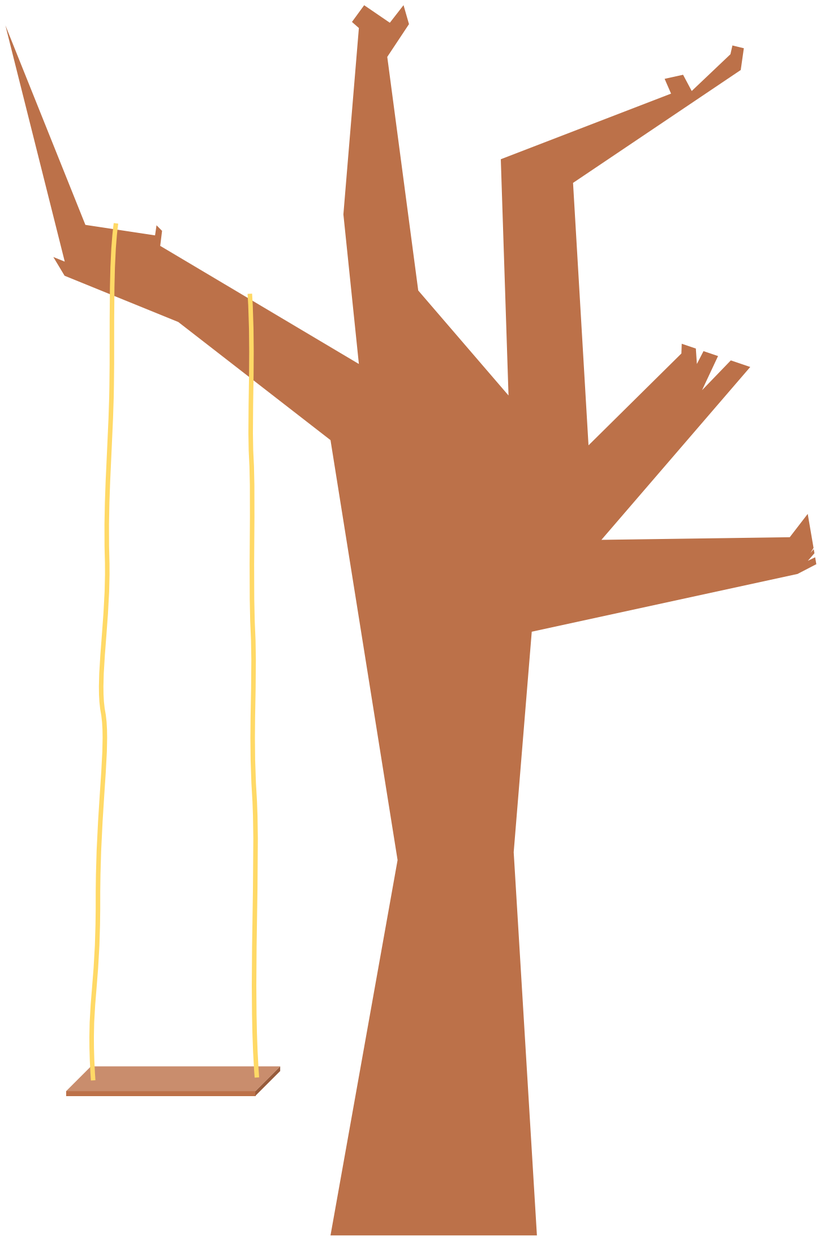
Schritt für Schritt


Wert
Vermittler
Verkaufsperformance einsehen
VerkaufsApp
Excel
Audit-Trail


Vermittler
Verkaufsperformance einsehen
VerkaufsApp
Excel
Audit-Trail
Wert
Evolution




Vermittler
2) verbessern!
Verkaufsperformance einsehen
VerkaufsApp
SalesBuddy
4) abschalten!
3) neukaufen!
Business Activity Monitoring
Excel
Audit-Trail
1) zurechtstutzen!
Wert
Evolution



Vermittler

Verkaufsperformance einsehen

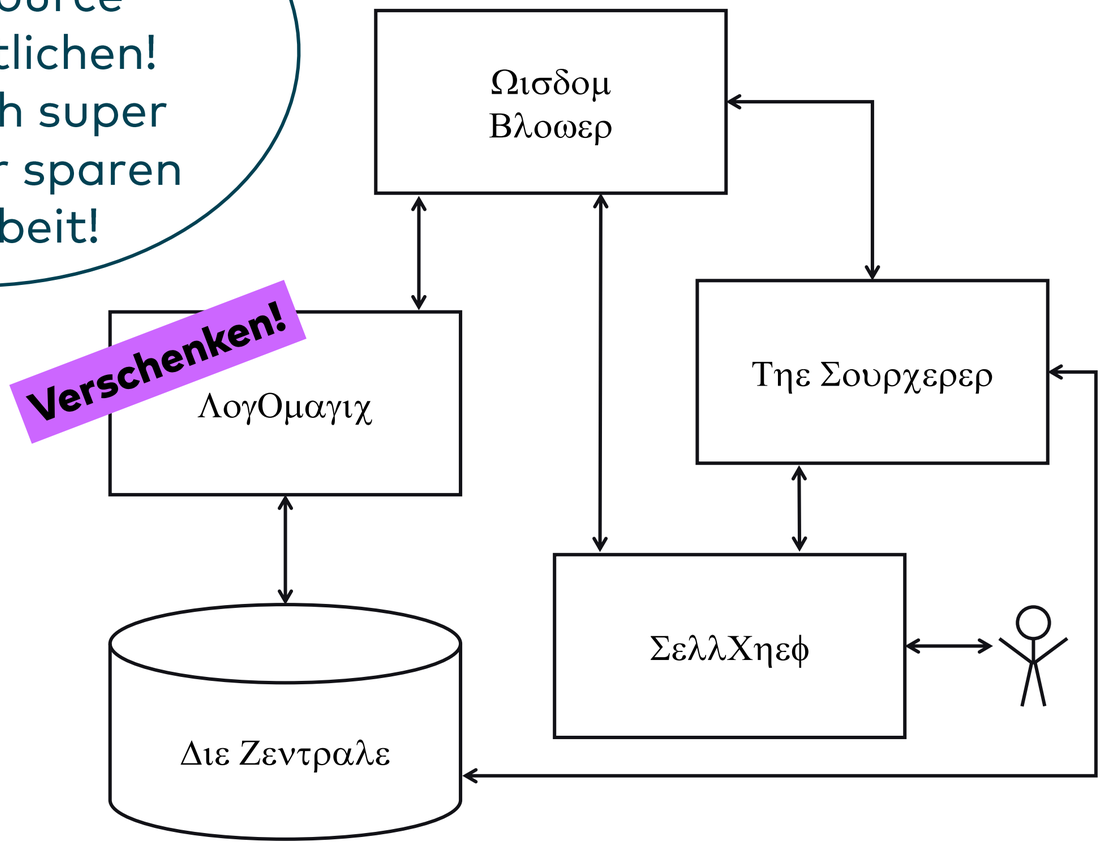
SalesBuddy

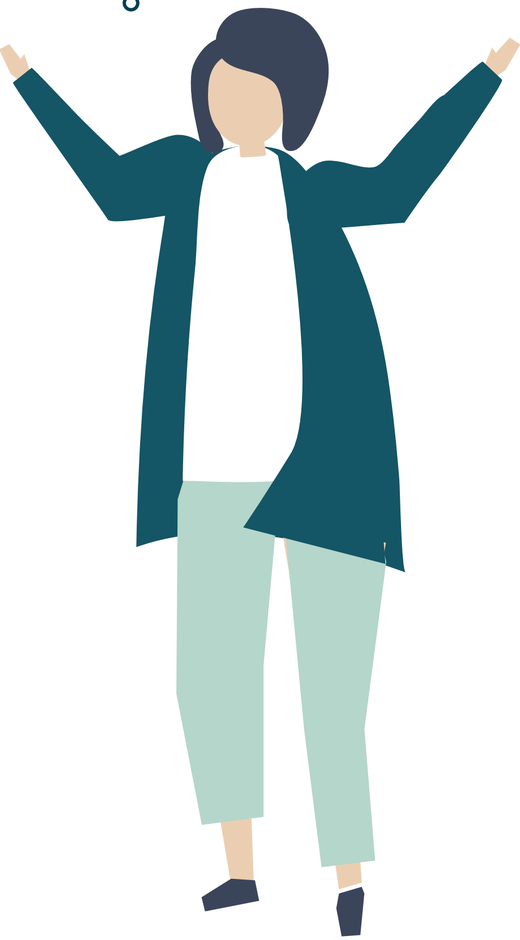
Business Activity Monitoring

Wie könnt ihr damit anfangen?

Typische Situation

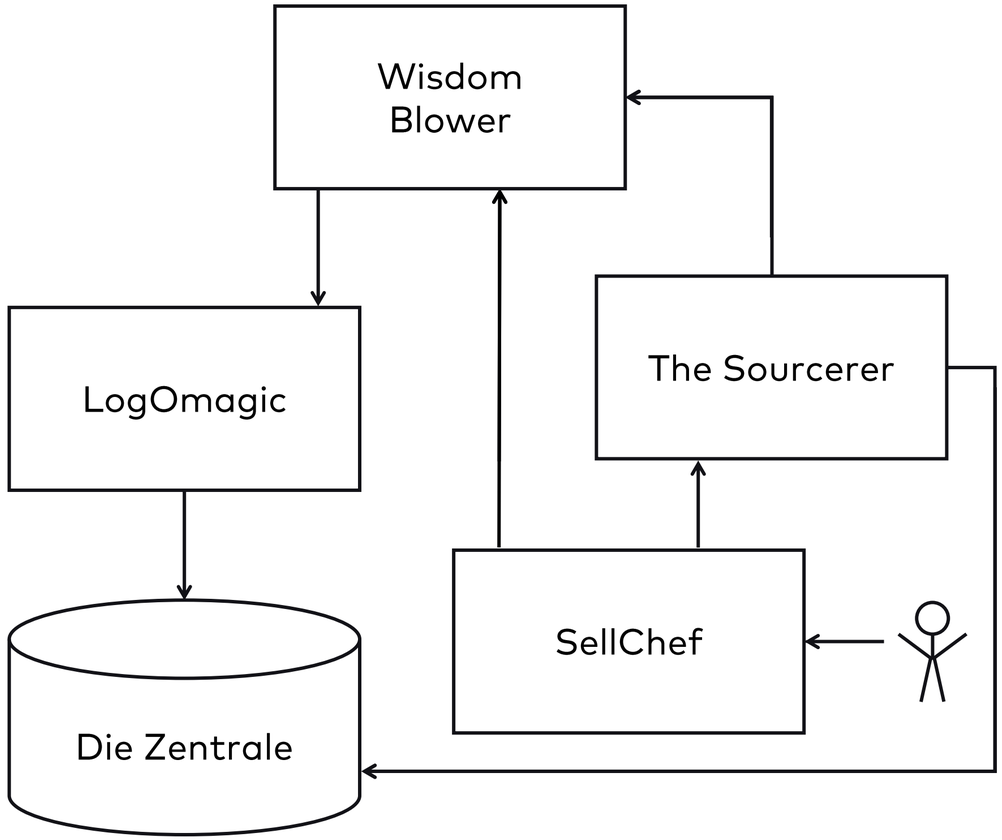
Wir sollten ΛογΟμαγιχ als Open Source veröffentlichen! Bietet sich super an und wir sparen uns Arbeit!





Schritt 1: Wer braucht was?

Systemüberblick



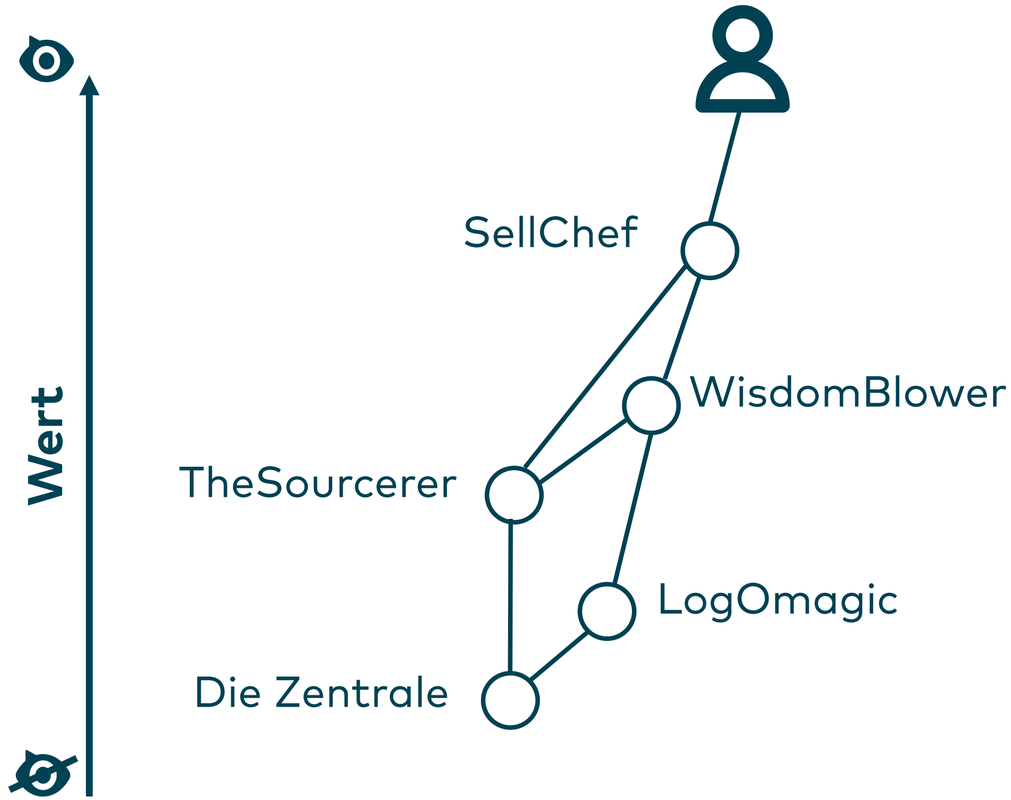
Legende

Nutzer

Kompo-nente

hängt ab von

Wardley Map



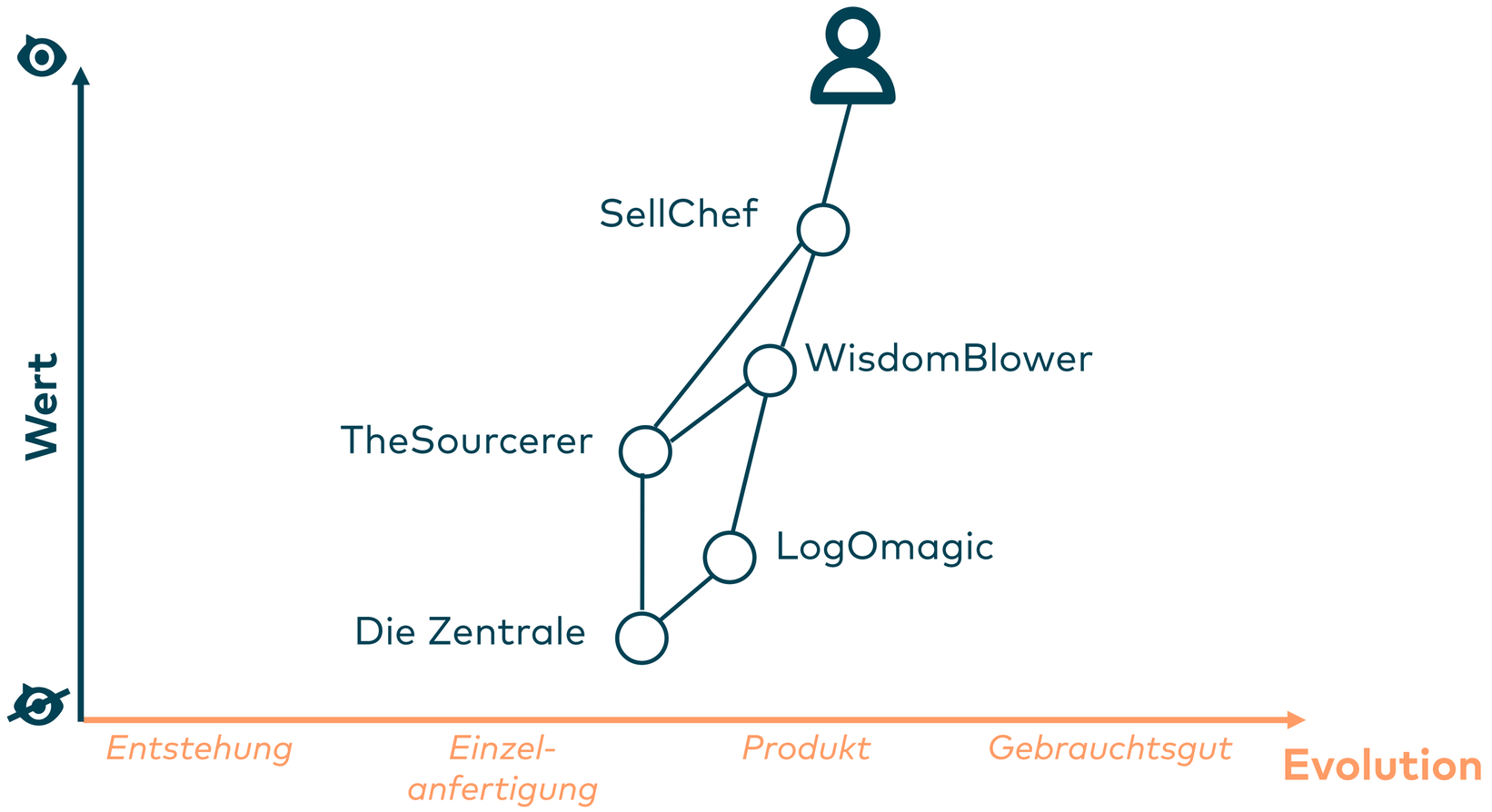
Legende

Nutzer

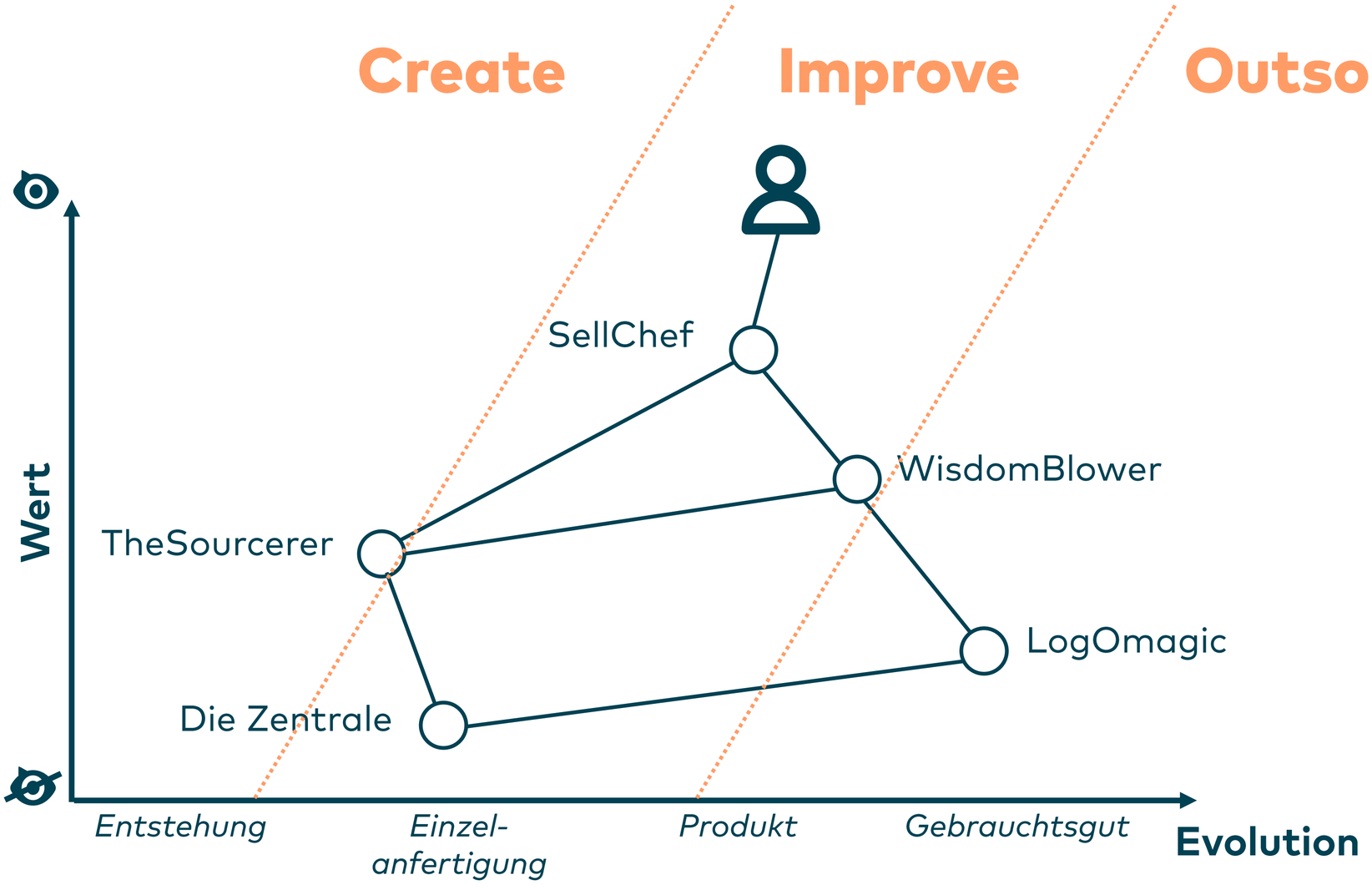
Komponente

braucht

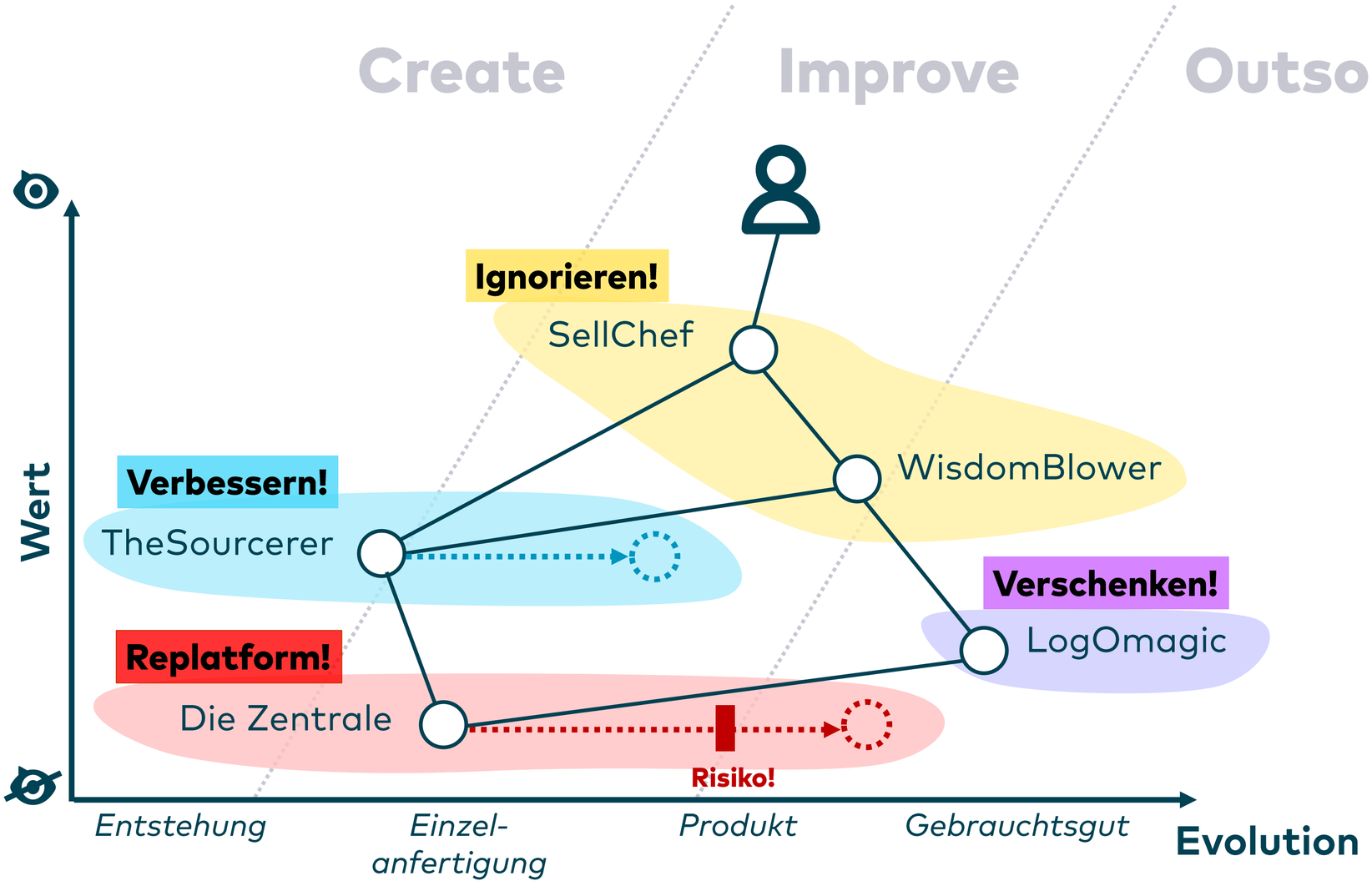
Schritt 2: Wie weit sind wir?



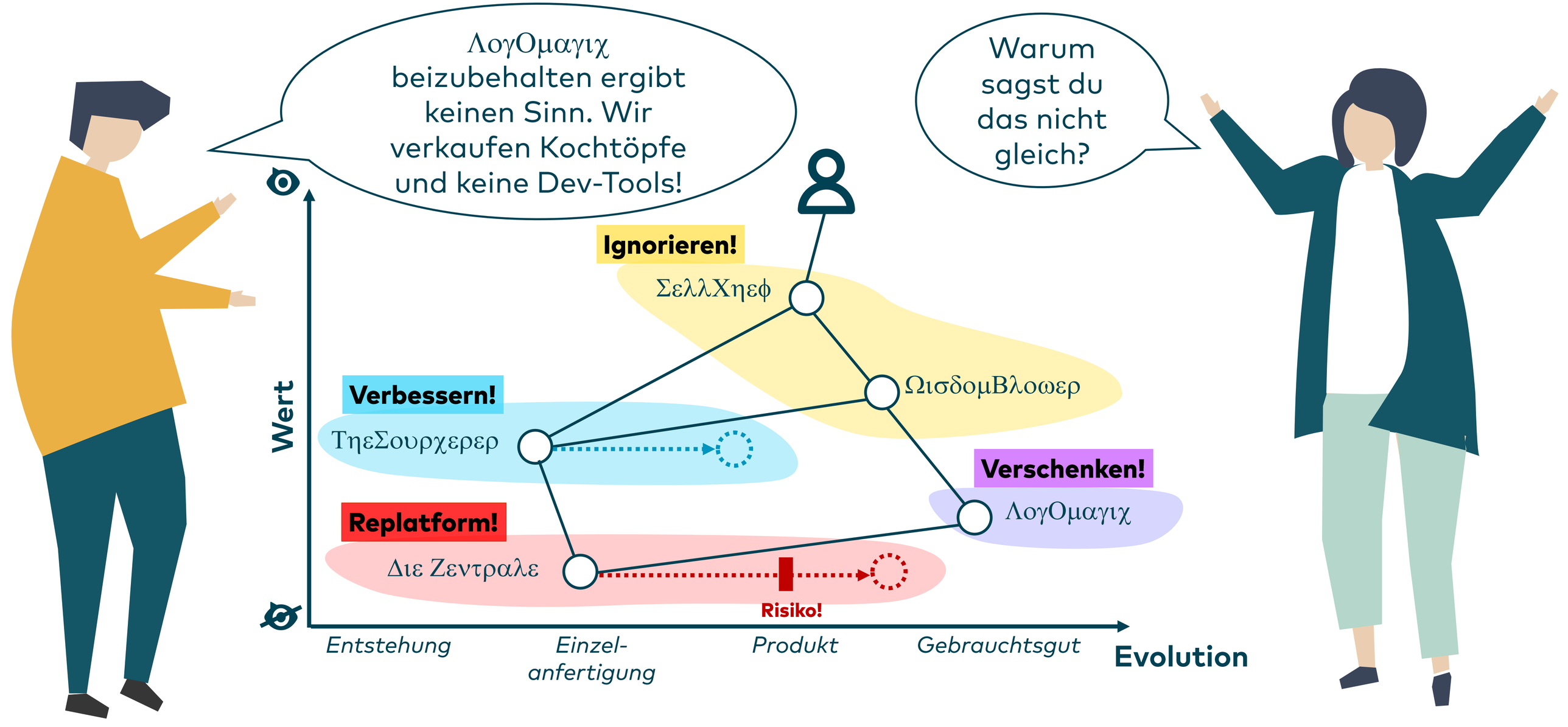
Schritt 2: Wie weit sind wir?



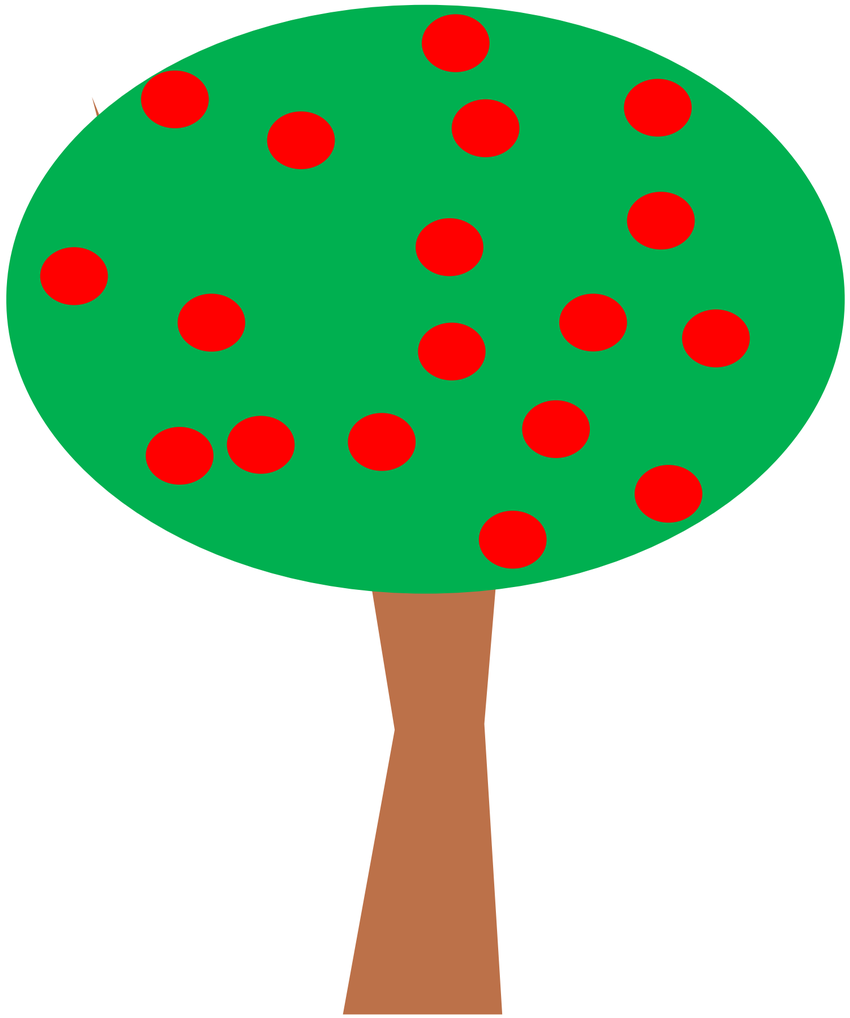
Schritt 3: Wo wollen wir hin?



Schritt 3: Wo wollen wir hin?



Happy End!



Welche Fragen gibt es noch?

Diskussionen, Anmerkungen, Sonstiges

https://www.linkedin.com/in/markus-harrer/

Mehr Informationen zum Thema

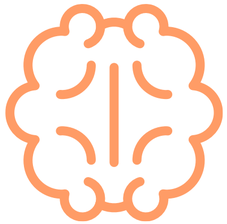
- Blog-Post „Evolving software like an orchardist"
  - https://www.innoq.com/en/blog/2023/05/evolving-software-like-an-orchardist/
- Mein Buch „Strategische Spielzüge – Softwaresysteme listig weiterentwickeln" *
  - https://leanpub.com/strategische-spielzuege/
- Awesome Legacy Systems
  - https://github.com/feststelltaste/awesome-legacy-systems/
- Architecture Improvement Method aim42
  - https://aim42.github.io/
- Meine TOP 5 Empfehlungen für Wardley Mapping
  - https://www.feststelltaste.de/top-5-learning-wardley-maps/

* Early Access

INNOQ Deutschland

Gegründet

1999



Mitarbeitende

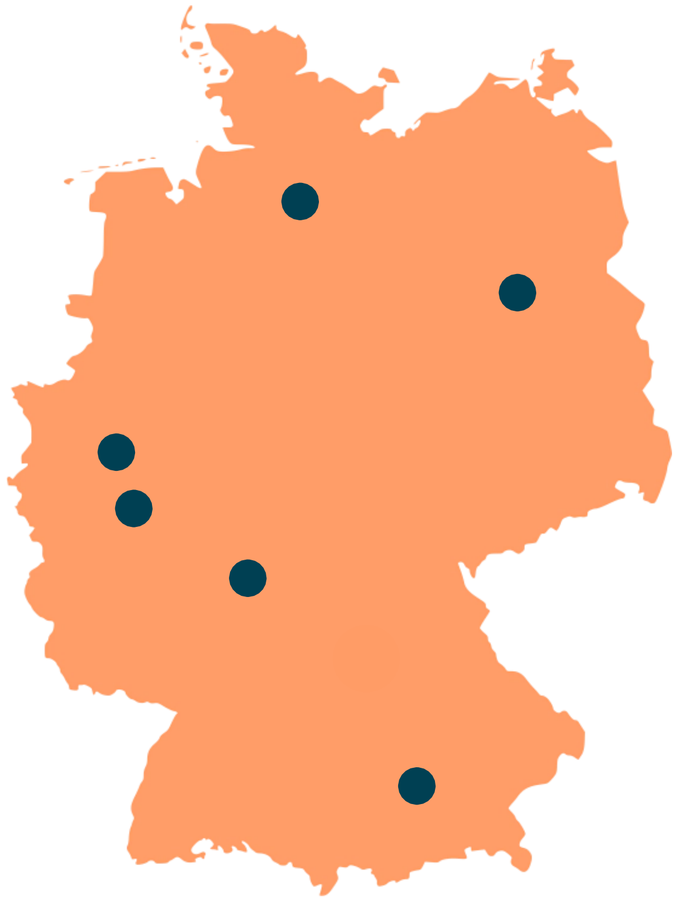
150+

Kunden

300+

Jahresumsatz 2022

~22 Mio.

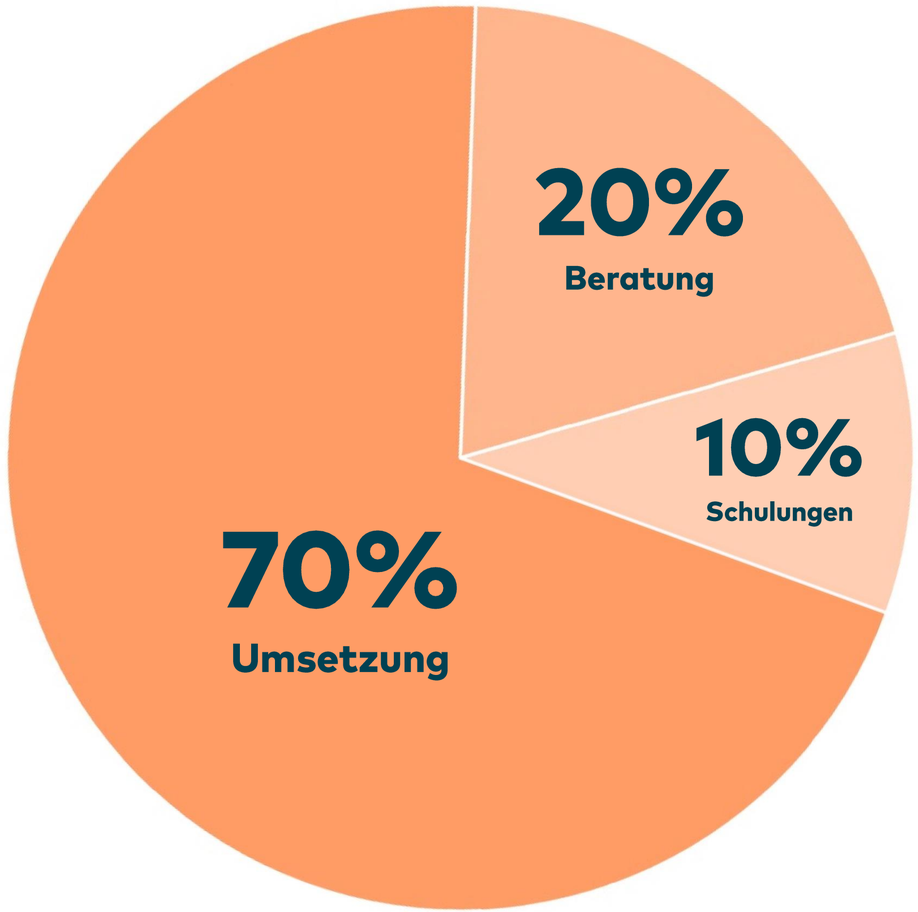


Kunden **Finance** | **Telko** | **Logistik** | **E-Commerce** | **Fortune 500** | **KMU** | **Start-ups**

AXA Eckelmann ebay VORWERK breuninger mobile.de

Unsere Leistungen

- Software-Entwicklung & -Architektur
- Strategie- und Technologie-Beratung
- Produktkonzeption & Design
- Data, AI, Infrastruktur & Betrieb
- Wissenstransfer, Coaching & Trainings



Schwerpunkt Modernisierung

Wir sind auf der OOP 2024!

https://briefing.innoq.com/

Harrer'sche Checkliste

17 Maßnahmen, die du ergreifen kannst, um dein Altsystem voranzubringen

Ignorieren!
Abschalten!
Am Leben halten!
Verkaufen!
Outsourcen!
Verschenken!
Neukaufen!
Präservieren!
Umwidmen!
Ausschlachten!
Zurechtstutzen!
Umhüllen!
Ersetzen!
Konvertieren!
Verbessern!
Replatform!
Rehost!

Business

Development

Operations

17 Maßnahmen, die du ergreifen kannst, um dein Altsystem voranzubringen

Ignorieren!

Eine einfache, aber wirkungsvolle „Maßnahme“ ist es, mit dem auszukommen, was du hast. Vielleicht ist dein System gar nicht so schlecht und erfüllt seine Aufgabe gerade gut genug. Diese Umstände könnten ein Grund sein, gar nichts zu tun.

Abschalten!

Stelle die Arbeit an und den Support für das System ein. Vielleicht ist eine Änderung deines Geschäftsmodells so bedeutend, dass du auch dein Produktportfolio ändern und alte Systeme hinter dir lassen musst.

Am Leben erhalten!

Vielleicht hast du ein System, das auf veralteten Hardware- oder Softwareplattformen basiert. Du könntest alle notwendigen Ressourcen kaufen, damit dieses System so lange wie möglich weiterlebt, wenn keine grundlegende Veränderung nötig ist.

Verkaufen!

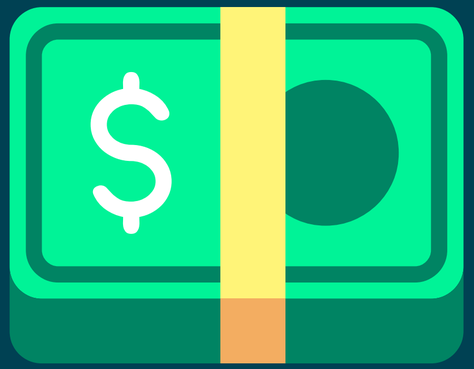
Du siehst vielleicht keine Zukunft für die Software, die du entwickelst. Aber andere vielleicht schon! Vielleicht wären sie froh, wenn sie das System von dir kaufen könnten, um ihre Zukunft zu sichern und ihr Geschäft am Leben zu erhalten.

Outsourcen!

Mach deine Probleme zu Problemen anderer, indem du sie das System für dich weiterentwickeln und es dir über eine Bibliothek, einen Service oder ein Softwareprodukt zur Verfügung stellen lässt.

Verschenken!

Du brauchst das Softwaresystem nicht mehr, aber da draußen gibt es eine Community, die es unbedingt nutzen will oder darauf angewiesen ist? Tue ihnen den Gefallen und übertrage ihnen die Rechte am Code. Auch die Freigabe der Software als Open Source könnte hier eine gute Maßnahme sein.

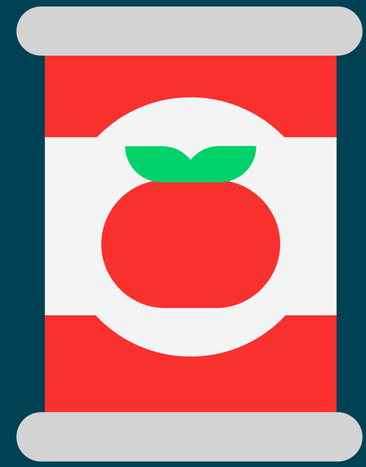


Neukaufen!

Deine selbst entwickelte Software tut das, was andere Software-systeme da draußen auch tun (und/oder sie tun es sogar besser für weniger Geld?). Warum solltest du Geld für Entwicklungsaktivitäten verschwenden? Der Kauf von Software von anderen, die die Arbeit für dich erledigt, könnte eine bessere Alternative sein.

Präservieren!

Vielleicht bist du dir zu 99,99% sicher, dass du das Softwaresystem nicht mehr brauchst, aber du würdest viel besser schlafen, wenn du es dir noch einmal ansehen könntest, nur für den Fall? Lege es in eine Umgebung (Emulatoren, virtuelle Maschinen), in der du es bei Bedarf wieder ausführen kannst.



Umwidmen!

Einige zusätzliche Funktionen, die im Laufe der Jahre entwickelt wurden, könnten eine ganz andere Kundengruppe ansprechen als die ursprüngliche Idee deines Systems. Mache diese Funktionalität zum neuen Kern deines Systems und biete sie unter einem anderen Produktnamen, aber mit der gleichen Codebasis an.

Ausschlachten!

Wenn fast alle Systemfunktionen nicht mehr benötigt werden, aber viel Wartungszeit kosten, könntest du die wenigen Funktionen identifizieren, die es noch wert sind. Nimm diese Funktionen aus dem alten System heraus, „verpflanze“ sie in ein neueres System und lösche das hinterlassenen Chaos.

Zurechtstutzen!

Es kann sein, dass andere Systeme dein System missbrauchen, indem sie ihre Daten direkt in deiner Datenbank speichern oder über dein System auf externe Dienste zugreifen. Entferne diese zusätzlichen Funktionen und Daten, die nichts mit dem Umfang und Zweck deines Systems zu tun haben.

Umhüllen!

Wenn der Technologie-Stack deines Systems nicht mit der ausgefallenen Technologie da draußen kompatibel ist, kannst du einen Adapter zwischen beide Welten setzen. Auf diese Weise bringst du deinem alten Softwaresystem neue Tricks (oder Integrationen) bei, indem du es einwickelst.

Ersetzen!

Genug ist manchmal genug und ein würdiger Nachfolger für dein System muss her. Auch wenn es riskant ist, gibt es in manchen Fällen triftige Gründe, ein Softwaresystem technisch komplett neu zu schreiben. Befreie dich von den Fesseln deiner technischen Vergangenheit mit einer kompletten Neuentwicklung auf der grünen Wiese!

Konvertieren!

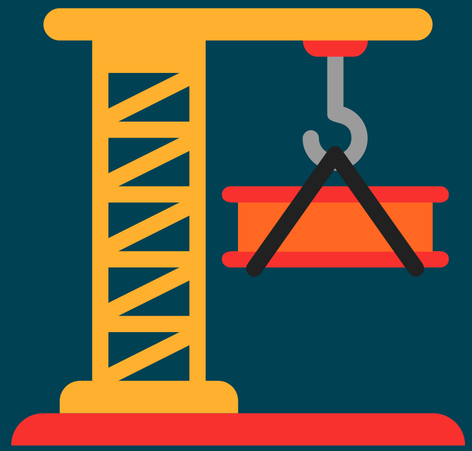
Wenn du Schwierigkeiten hast, Leute zu finden, die eine bestimmte Programmiersprache oder Plattform beherrschen, kannst du darüber nachdenken, den Quellcode des Systems von einer alten Sprache in eine neuere zu übersetzen.

Verbessern!

Ein schlecht strukturiertes und wenig leistungsfähiges System ist nicht immer ein Grund, es komplett aufzugeben. Es gibt viele Techniken, die dir helfen, dein System parallel zur Feature-Entwicklung zu modernisieren. Stetig weiterentwickelte Systeme werden nicht so schnell veralten.

Replatform!

Es ist nicht immer schlechter Code, der uns zurückhält. Auch archaische und nicht mehr unterstützte Plattformen wie proprietäre Umgebungen oder teure Datenbankmanagementsysteme können uns aufhalten. Überlege dir, ob du dein System auf ein neues Fundament stellen kannst, um es für die Zukunft fit zu machen.



Rehost!

Angenommen, du bist mit begrenzter Hardware und Infrastruktur in deinem lokalen Rechenzentrum konfrontiert oder dein Betriebsteam kann deinen Anforderungen nicht mehr gerecht werden. In diesem Fall kannst du darüber nachdenken, dein System in einer anderen Umgebung zu hosten, in der es keine solchen Einschränkungen gibt.

Ignorieren!
Abschalten!
Am Leben halten!
Verkaufen!
Outsourcen!
Verschenken!
Neukaufen!
Präservieren!
Umwidmen!
Ausschlachten!
Zurechtstutzen!
Umhüllen!
Ersetzen!
Konvertieren!
Verbessern!
Replatform!
Rehost!

Business

Development

Operations

Welche der Aktionen kommt für euch in Frage?

Quellen der 17 Aktionen

- Der Thread im Original von Markus Harrer auf Twitter https://twitter.com/feststelltaste/status/1451148235121299461
- Nine Things You Can Do with Old Software by Grady Booch https://computer.org/publications/tech-news/on-architecture/nine-things-you-can-do-with-old-software
- 6 Strategies for Migrating Applications to the Cloud from the AWS Cloud Enterprise Strategy Blog https://aws.amazon.com/de/blogs/enterprise-strategy/6-strategies-for-migrating-applications-to-the-cloud/