

Software Product Mastering



Inhaltsverzeichnis

- Eisbrecher

Theorie

- Software, das unsichtbare Biest
- Das "Was" und das "Wie"
- Stacey Matrix
- Cynefin-Modell
- Software-Komplexität im Laufe der Jahre
- Conways Gesetz
- Alles im Zusammenhang
- Projekte & Produkte

Kultur

- Die fünf agilen Werte
- Fehlerkultur
- Management Stile
- Rollen
 - Devteam
 - Product Owner

- Scrum Master

Methoden & Werkzeuge

- Systemischer Ansatz
- Methoden Übersicht
 - Scrum
- Arbeitselemente & Konzepte Übersicht
- Deliberating structures & practices Übersicht
 - Paperplane Game
- Werkzeuge & Automatisierung Übersicht
 - CI/CD mit Gitlab
- Architektur & Techstacks Übersicht
 - Sample: eShopOnWeb (.NET Stack)

Roter Faden durch die Themen →

Software Product Mastering

[See the board](#)



Software Product Mastering ↗

Modified 19 hours ago

[Icebreaker_kickoff.de](#)

Eisbrecher



1. tscheck.in: „Wie komme ich heute hier an?“

- Wir starten gemeinsam und schauen, wie jede*r heute hier ankommt.

2. Vorstellungsrunde: „Wer bist du?“

- Jetzt lernen wir uns besser kennen.
- Wenn du schon etwas auf dem Miro-Board gepostet hast, kannst du kurz einen Punkt daraus vorstellen.
- Für alle anderen gibt es drei spannende Fragen – such dir eine aus:
 - „Wenn ich ein Tier wäre, welches wäre das und warum?“
 - „Was ist ein überraschender Fun Fact über mich?“
 - „Was möchte ich in dieser Veranstaltung besonders lernen oder ausprobieren?“
- Wir gehen reihum und du kannst so viel oder so wenig erzählen, wie du möchtest.

3. Unsere Erwartungshaltung

- Lass uns darüber sprechen, was wir gemeinsam von dieser Veranstaltung erwarten.
- Wir stellen uns drei Fragen:
 1. „Angenommen, die Veranstaltung ist genial, was ist dann anders?“
 2. „Woran merke ich das?“
 3. „Woran merken das die anderen?“
- Deine Antworten kannst du mündlich teilen, oder wir sammeln sie direkt auf einem Miro-Board.
- Gemeinsam fassen wir die wichtigsten Punkte zusammen, damit wir ein gutes gemeinsames Bild bekommen.

4. Den Tag gemeinsam planen

- Jetzt gestalten wir den Tag zusammen.
 - Auf dem Miro-Board findest du einen groben Tagesplan mit freien Zeitslots und Themenvorschlägen.
 - Du kannst mitentscheiden:
 - „Welche Themen sind mir heute besonders wichtig?“
 - „Welche Fragen möchte ich klären?“
 - „Wie möchte ich am liebsten arbeiten?“
 - Wir ergänzen den Plan gemeinsam und sorgen dafür, dass alle wichtigen Punkte berücksichtigt werden.

Software_the_invisible_beast.de

Software - das unsichtbare Biest



Approaches to Software Development: The Nature of Software - OpenLearn
What are the 10 biggest risks in software development? - Codebots
Need for a Soft Dimension - arXiv

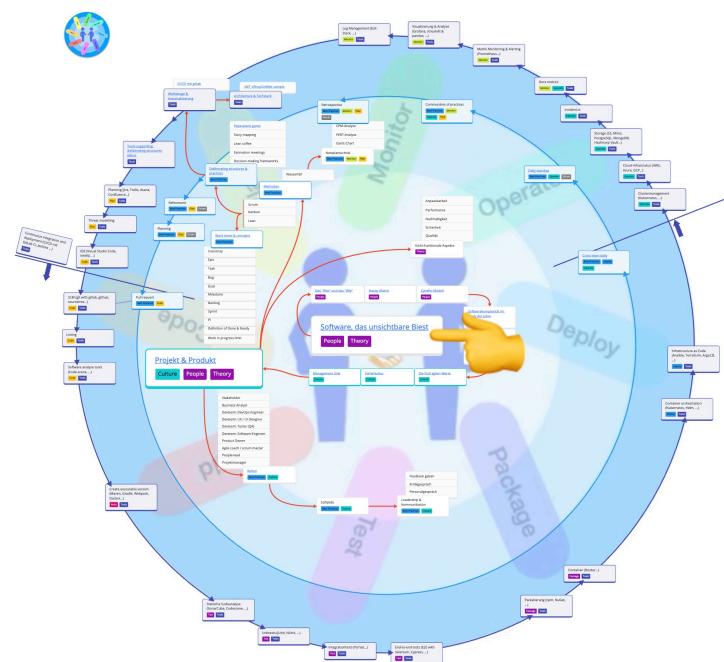
Integrating User eXperience Practices into Software Development Processes: Implications of Subjectivity and Emergent Nature of UX - arXiv

Concept-centric Software Development - arXiv

Rethinking Software

Engineering in the Foundation Model Era: A Curated Catalogue of Challenges in the

Development of Trustworthy FMware - arXiv Lean IT - Wikipedia



[12 Common Software Development Obstacles And How To Tackle Them - Forbes](#)
[Six Sigma in Software Development: Benefits Explained - Six Sigma Online](#)

- **Unsichtbarkeit und fehlende Haptik:** Software ist nicht greifbar, im Gegensatz zu physischen Produkten; ihre Komplexität und Qualität sind nicht direkt sichtbar oder spürbar.
- **Schwierigkeiten bei der Qualitätserkennung:** Qualitätsmerkmale wie Effizienz, Sicherheit und Skalierbarkeit sind für Nutzer schwer erkennbar und messbar, oft nur über die Benutzeroberfläche erlebbar.
- **Hohe Einstiegshürde im Software-Ökosystem:** Abstrakte Architektur, Schnittstellen und Infrastruktur im Hintergrund machen das Verständnis von Softwareprodukten für Laien schwierig.
- **Wahrnehmung der Entwicklungs-Komplexität:** Der Entwicklungsaufwand für Software wird oft unterschätzt, da Nutzer die internen Prozesse und Fachkenntnisse nicht sehen.
- **Fazit:** Die Unsichtbarkeit und fehlende direkte Wahrnehmung schaffen hohe Barrieren, um Software intuitiv und auf sensorischer Ebene zu verstehen und zu bewerten, was sie schwer zugänglich und bewertbar macht.

Software ist in vielerlei Hinsicht eine „unsichtbare“ Technologie, die sich stark von physischen Produkten unterscheidet, was die Wahrnehmung, die Haptik und die intuitiv nutzbaren Sinne betrifft. Diese Unsichtbarkeit führt zu Herausforderungen auf mehreren Ebenen, insbesondere hinsichtlich der Benutzung, der Qualität und des Ökosystems, in dem sie eingebettet ist. Anders als bei physischen Objekten fehlt bei Software häufig der direkte sensorische Zugang – das Anfassen, das Sehen, das Hören – wodurch auch die Komplexität und das Entwicklungsniveau für Laien weniger sichtbar sind.

1. Fehlende Haptik und Sensorik

Im Gegensatz zu greifbaren Produkten wie Fahrrädern oder Raketen, bei denen die Komplexität der Konstruktion und die Qualität der Materialien unmittelbar sichtbar oder spürbar sind, fehlt bei Software jegliche Form der Haptik und des sichtbaren Fortschritts. Wenn man ein Fahrrad oder eine Rakete betrachtet, lassen sich oft sofort Rückschlüsse auf den Entwicklungsaufwand ziehen: Die verwendeten Materialien, die Größe, die Art der Verschraubungen und die Anzahl der Einzelteile geben Hinweise darauf, wie aufwendig das Produkt in der Herstellung ist. Software dagegen besteht aus Code und läuft im Verborgenen auf Hardware. Der

Entwicklungsaufwand und die Komplexität des Programmcodes sind unsichtbar; was der Benutzer sieht, ist meist nur ein grafisches Interface – eine Benutzeroberfläche, die die technischen Prozesse im Hintergrund verbirgt.

2. Herausforderungen bei der Benutzung und Sichtbarkeit der Qualität

Bei physischen Produkten lässt sich oft intuitiv erkennen, ob sie funktionieren und wie gut sie verarbeitet sind. Die Qualität eines Sandwichs oder eines Fahrrads ist oft direkt erfahrbar: Ist der Salat frisch, das Brot knusprig, die Schaltung leichtgängig, oder das Material robust? Software jedoch bietet diesen unmittelbaren Zugang nicht. Nutzer sehen und erleben oft nur die Oberflächendarstellung, nicht jedoch die innere Logik, die Struktur, die Sicherheitsmechanismen oder die Effizienz der Algorithmen. Qualitätsmerkmale wie Effizienz, Skalierbarkeit und Sicherheit sind schwer verständlich und messbar, und es bedarf oft tiefer technischer Kenntnisse, um diese Eigenschaften zu bewerten oder gar zu diskutieren.

Der Entwicklungsprozess von Software bleibt unsichtbar: Der Benutzer kann nicht erkennen, „wie weit“ ein Softwareprodukt entwickelt ist oder wo die Qualitätsmerkmale liegen, wie man das vielleicht bei einem physischen Produkt könnte. Häufig können Endnutzer nicht sofort feststellen, ob eine Software gut entwickelt oder sicher ist, und Softwarefehler sind oft erst im Betrieb und unter Last spürbar – eine Eigenschaft, die es umso schwieriger macht, Qualität zu bewerten oder sogar zu verbessern.

3. Einstiegshürde: Abstraktheit und fehlende Intuition im Software-Ökosystem

Das Ökosystem, in das Software eingebettet ist, erhöht ebenfalls die Unsichtbarkeit und die Einstiegshürde. Ein Großteil der Software-Architektur, der zugrunde liegenden Systeme und der Infrastruktur sind für den normalen Benutzer weder sichtbar noch intuitiv verständlich. Unterschiedliche Plattformen, Schnittstellen, Datenbanken und Netzwerkkonfigurationen bilden das Rückgrat von Softwareprodukten, doch sind sie im Hintergrund verborgen und schwer nachvollziehbar. Auch die Werkzeuge und Frameworks, die bei der Entwicklung verwendet werden, sind hochgradig abstrakt und techniklastig, was es für Laien zusätzlich erschwert, den Entstehungsprozess oder die Qualität eines Produkts nachzuvollziehen.

4. Komplexität in der Entwicklung und Wahrnehmung

Im Gegensatz zu einem Fahrrad, dessen Aufbau recht schnell intuitiv zu verstehen ist, wird der Entwicklungsaufwand für Software unterschätzt. Die Komplexität und die Fachkenntnisse, die nötig sind, um robuste, sichere und leistungsfähige Software zu entwickeln, sind für Laien oft nicht erkennbar und daher schwer verständlich. Ein

Programm, das auf den ersten Blick nur aus ein paar Klicks auf einer Benutzeroberfläche besteht, kann intern auf Millionen von Zeilen Code, komplexen Algorithmen und detaillierten Abhängigkeiten beruhen. Die Einstiegshürde ist hier sehr hoch, da allein das Verständnis der Grundbausteine und der eingesetzten Werkzeuge tiefgehendes Wissen und jahrelange Erfahrung voraussetzt.

Fazit

Das Fehlen von Haptik und die Unsichtbarkeit der Softwareinfrastruktur sind wesentliche Gründe, warum Software für viele schwer zugänglich und bewertbar ist. Ohne direkten sensorischen Zugang und die Möglichkeit, Komplexität und Qualität intuitiv wahrzunehmen, ist Software für Laien oft wie eine „Black Box“ – ein Produkt, dessen Entwicklung, Aufwand und Qualitätskriterien erst nach intensiver Beschäftigung und technischem Verständnis klar werden. Dies schafft eine erhebliche Barriere, die es schwer macht, Software auf eine Weise zu diskutieren und zu bewerten, die den gleichen direkten und intuitiven Zugang ermöglicht wie bei greifbaren Produkten.

Unsichtbarkeit und Komplexität spielerisch erleben

Dieses interaktive Programm zeigt, wie Entscheidungen über Programmiersprachen und Frameworks unsichtbare Software-Eigenschaften wie Performance, Sicherheit, Benutzerfreundlichkeit und Entwicklungsaufwand beeinflussen. Der Nutzer wählt über Dropdown-Menüs eine Sprache und ein Framework, und ein Balkendiagramm zeigt die Auswirkungen der Wahl auf die Qualitätseigenschaften, die sonst schwer erfassbar sind. So wird erlebbar, wie technische Entscheidungen Softwarequalität beeinflussen können.

```
In [1]: import ipywidgets as widgets
from IPython.display import display, clear_output
import matplotlib.pyplot as plt

# Software simulation class with visual feedback
class SoftwareSimulation:
    def __init__(self):
        self.performance = 0
        self.security = 0
        self.usability = 0
        self.development_effort = 0

    def update_quality(self, quality, value):
        """Updates the quality attribute based on decision"""
        setattr(self, quality, getattr(self, quality) + value)

    def display_status(self):
        """Displays the current status as a bar chart"""
        qualities = ['Performance', 'Security', 'Usability', 'Development']
        values = [self.performance, self.security, self.usability, self.development_effort]
        plt.bar(qualities, values)
        display(plt.gcf())
        clear_output(wait=True)
```

```

plt.figure(figsize=(8, 4))
plt.bar(qualities, values, color='skyblue')
plt.ylim(0, max(values) + 1)
plt.title("Software Quality Summary")
plt.ylabel("Quality Level")
plt.show()

# Initialize simulation
sim = SoftwareSimulation()

# Interactive widgets
language_dropdown = widgets.Dropdown(
    options=["Python", "C++", "JavaScript"],
    description="Language:"
)

framework_dropdown = widgets.Dropdown(
    options=["Django", "Flask", "React"],
    description="Framework:"
)

submit_button = widgets.Button(description="Submit")
# output = widgets.Output()

# Function to handle button click
def on_submit_button_clicked(b):
    language = language_dropdown.value
    framework = framework_dropdown.value

    # Update quality based on language choice
    if language == "Python":
        sim.update_quality("usability", 2)
        sim.update_quality("development_effort", 1)
    elif language == "C++":
        sim.update_quality("performance", 3)
        sim.update_quality("development_effort", 2)
    elif language == "JavaScript":
        sim.update_quality("usability", 2)
        sim.update_quality("security", -1)

    # Update quality based on framework choice
    if framework == "Django":
        sim.update_quality("security", 2)
        sim.update_quality("development_effort", 1)
    elif framework == "Flask":
        sim.update_quality("usability", 1)
        sim.update_quality("development_effort", 1)
    elif framework == "React":
        sim.update_quality("usability", 2)
        sim.update_quality("security", -1)

    # Display updated status
    sim.display_status()

# Link button click to function
submit_button.on_click(on_submit_button_clicked)

```

```
# Display the interactive elements
display(language_dropdown, framework_dropdown, submit_button)
```

```
Dropdown(description='Language:', options=('Python', 'C++', 'JavaScript'),
value='Python')
Dropdown(description='Framework:', options=('Django', 'Flask', 'React'),
value='Django')
Button(description='Submit', style=ButtonStyle())
```



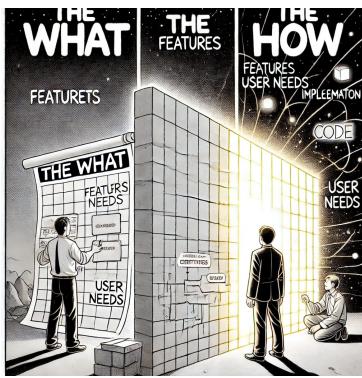
Code-Erklärung

Das Programm verwendet die Klasse `SoftwareSimulation`, um vier Qualitätsmerkmale zu speichern und dynamisch anzupassen. Jedes Dropdown-Menü löst eine Aktualisierung des Balkendiagramms aus und zeigt so die Qualitätseigenschaften live an. Dies ermöglicht eine visuelle und intuitive Rückmeldung, die die „unsichtbaren“ Aspekte der Softwareentwicklung veranschaulicht.

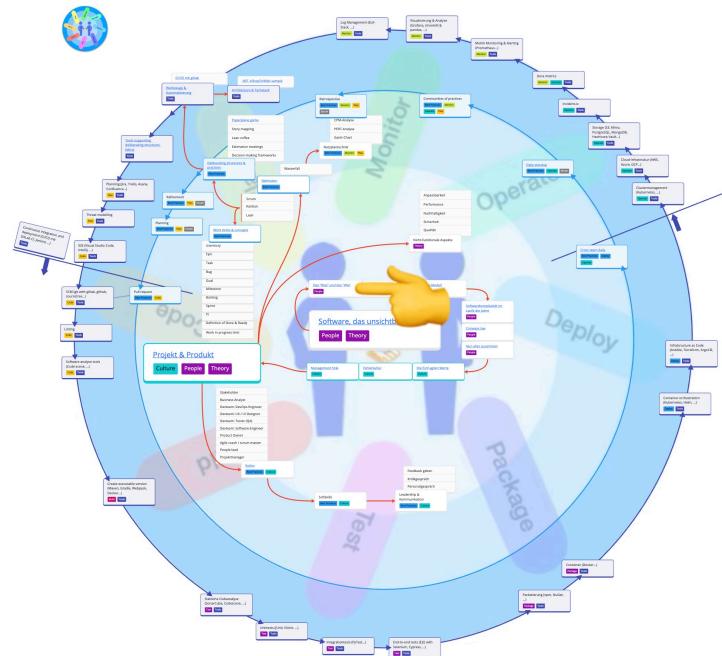
What_and_How.de

[Das unsichtbare Biest](#)

Das "Was" und das "Wie"



Requirements Vs Implementation: Keep Them Apart - DConsulted
 Software Requirements Specifications: Best Practices and SRS - AltexSoft
 SDLC Guide: Requirement Analysis in Software Engineering - Stratoflow
 Why You Need to Understand Software Requirements as a Software Engineer - freeCodeCamp
 Chapter 3: Requirements – Software Engineering: A Modern Approach
 Software Requirements Engineering - Software Mind
 Requirements quality research: a harmonized theory, evaluation, and roadmap - SpringerLink
 Interface-Based Software Requirements Analysis - SpringerLink
 Measuring and improving software requirements elicitation in a small-sized software organization - SpringerLink
 Recommendation systems-based software requirements elicitation process: a systematic literature review - Journal of Engineering and Applied Science

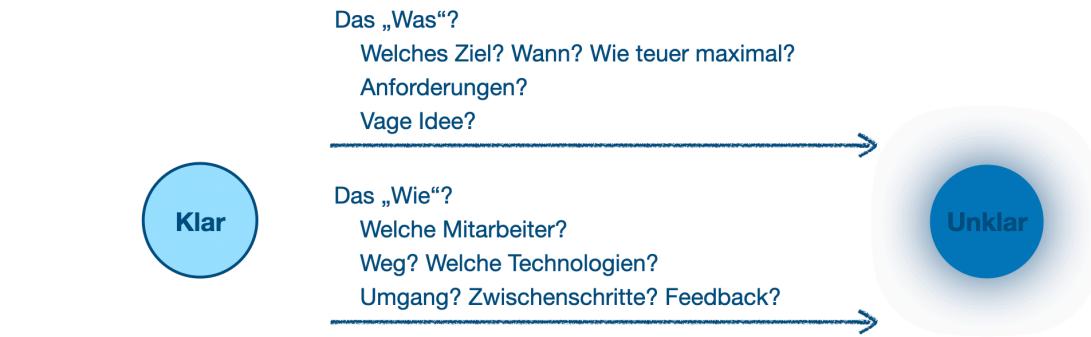


Entkopplung von „Was“ (Anforderungen) und „Wie“ (Umsetzung) in der Softwareentwicklung



Das „Was“ und das „Wie“

Entkopplung von Anforderung und Umsetzung



„Was“ – Anforderungsebene

- **Ziel:** Klare Definition des gewünschten Endergebnisses
- **Zeitplan:** Festlegung von Meilensteinen und Deadlines
- **Budget:** Maximal erlaubte Kosten
- **Anforderungen:** Präzisierung von Funktionen und Nutzerbedürfnissen
- **Klarheit:** Vage Idee oder detailliertes Ziel?

„Wie“ – Umsetzungsebene

- **Team:** Passende Mitarbeiter und Kompetenzen
- **Vorgehensweise:** Auswahl der Methodik (z. B. agil oder Wasserfall)
- **Technologien:** Festlegung von Tools und Plattformen
- **Risiken:** Umgang mit technischen Unsicherheiten (z. B. Prototyping)
- **Zwischenschritte & Feedback:** Milestones und regelmäßige Überprüfungen

Vorteile der Entkopplung

- **Flexibilität:** Änderungen im „Was“ oder „Wie“ unabhängig voneinander möglich
- **Risikominimierung:** Risiken können gezielt adressiert werden
- **Klarheit und Verantwortung:** Klare Verantwortlichkeiten und Kommunikationswege

Die Entkopplung von Anforderung („Was“) und Umsetzung („Wie“) in der Softwareentwicklung ist essenziell, um Flexibilität zu schaffen und eine klare Aufteilung der Verantwortlichkeiten zu ermöglichen. Diese Entkopplung erlaubt es, dass sich die Anforderungen und das Verständnis des Ziels unabhängig vom technischen Lösungsweg weiterentwickeln können. Dadurch wird die Entwicklung weniger starr und ermöglicht eine iterative Anpassung, die sowohl Unsicherheiten in den Anforderungen als auch technische Risiken berücksichtigt.

Das „Was“ – Anforderungsebene

Auf der Ebene des „Was“ werden strategische und geschäftliche Fragen geklärt, die die Grundlage für das Projekt bilden. Hier geht es darum, die Motivation und die Erwartungen hinter der Software klar zu verstehen, ohne sich sofort in die technischen Details zu vertiefen. Typische Fragen auf dieser Ebene sind:

- **Welches Ziel soll erreicht werden?**

Hier geht es darum, das gewünschte Endergebnis zu definieren, also den Nutzen oder die Problemlösung, die die Software bieten soll. Ein klares Ziel hilft dabei, den Wert des Projekts zu verstehen und sicherzustellen, dass alle Beteiligten eine gemeinsame Vision haben.

- **Wann soll das Ziel erreicht werden?**

Diese Frage hilft, eine Zeitleiste zu setzen und Meilensteine zu definieren. Ein realistischer Zeitplan unterstützt das Projektmanagement und ermöglicht es, priorisierte Anforderungen zuerst zu implementieren, falls Zeitdruck besteht.

- **Wie teuer darf die Entwicklung maximal sein?**

Das Budget beeinflusst die verfügbaren Ressourcen und hilft dabei, die Machbarkeit abzuschätzen. Es stellt sicher, dass das Projekt innerhalb der finanziellen Rahmenbedingungen bleibt und dass Entscheidungen zu Funktionen und Prioritäten getroffen werden können, falls Kompromisse notwendig sind.

- **Welche Anforderungen sind konkret?**

Hier werden die Hauptanforderungen spezifiziert, um den Projektumfang abzugrenzen. Dies könnte die Definition von Benutzerrollen, Sicherheitsanforderungen, Performance-Zielen oder spezifischen Funktionalitäten umfassen.

- **Handelt es sich um eine vage Idee oder ist das Ziel detailliert ausgearbeitet?**

Wenn das „Was“ nur als vage Idee vorliegt, ist eine agile, iterativ-inkrementelle Vorgehensweise empfehlenswert, um die Anforderungen im Laufe der Zeit zu konkretisieren. Eine detaillierte Anforderung hingegen kann präzisere Planungen und möglicherweise einen stärker strukturierten Ansatz ermöglichen.

Die Beantwortung dieser Fragen auf der Anforderungsebene legt den Fokus auf das „Was“ und gibt der Entwicklung klare, geschäftsgesetzte Rahmenbedingungen.

Das „Wie“ – Umsetzungsebene

Sobald das „Was“ in seinen Grundzügen geklärt ist, wird die Umsetzungsebene („Wie“) relevant. Hier wird der technische Lösungsweg detailliert und der Plan erstellt, wie die Anforderungen erfüllt werden können. Die Fragen auf dieser Ebene helfen, die praktische Umsetzung zu organisieren und Risiken in Bezug auf die Machbarkeit und Effizienz zu steuern.

- **Welche Mitarbeiter und Kompetenzen werden benötigt?**

Die Auswahl der Teammitglieder oder die Definition benötigter Kompetenzen ist entscheidend, um die Umsetzung effektiv zu gestalten. Die richtigen Fähigkeiten im Team ermöglichen es, die Anforderungen optimal zu erfüllen und technische Hürden schneller zu überwinden.

- **Welcher Weg soll eingeschlagen werden?**

Diese Frage betrifft den gesamten Prozess der Softwareentwicklung, also ob beispielsweise ein agiler Ansatz wie Scrum oder ein klassischer Wasserfall-Ansatz gewählt wird. Die Methodik sollte zum Projekt passen und hilft, Arbeitsabläufe zu strukturieren und effizient zu gestalten.

- **Welche Technologien und Werkzeuge werden eingesetzt?**

Die Auswahl der Technologien (z. B. Programmiersprachen, Frameworks, Plattformen) ist von zentraler Bedeutung, da sie die Architektur, Entwicklungszeit und Wartbarkeit beeinflusst. Dies betrifft auch Werkzeuge für das Projektmanagement, die Versionskontrolle und die Qualitätssicherung.

- **Wie wird mit technischen Risiken umgegangen?**

Hierzu zählen mögliche Risiken im Zusammenhang mit neuen Technologien, komplexen Architekturentscheidungen oder Integrationsproblemen. Eine Risikobewertung und die Identifikation von Experimenten (z. B. Prototyping oder Proof-of-Concepts) können helfen, technische Unsicherheiten zu verringern.

- **Gibt es Zwischenschritte und wie wird Feedback gesammelt?**

Die Definition von Zwischenschritten und Milestones ist wichtig, um den Fortschritt zu überwachen und die Einhaltung der Anforderungen sicherzustellen. Regelmäßiges Feedback (z. B. durch Stakeholder-Reviews oder Usability-Tests) stellt sicher, dass das Projekt auf Kurs bleibt und Anpassungen frühzeitig vorgenommen werden können.

Vorteile der Entkopplung von „Was“ und „Wie“

Durch eine klare Trennung dieser beiden Ebenen wird sichergestellt, dass das Team auf der Anforderungsebene flexibel auf Änderungen reagieren kann, ohne die technische Umsetzung zu beeinträchtigen. Genauso können auf der Umsetzungsebene technische Herausforderungen und Lösungen diskutiert werden, ohne dass die Gesamtanforderungen infrage gestellt werden. Hier sind einige wichtige Vorteile:

- **Flexibilität bei Anforderungen und technischem Weg:** Die Entwicklung bleibt anpassbar und kann auf geänderte Geschäftsanforderungen oder technologische Fortschritte reagieren.
- **Risikominimierung:** Durch die Entkopplung kann das Team spezifische Risiken auf jeder Ebene isoliert betrachten und gezielt adressieren.
- **Klarheit und Verantwortung:** Jedes Teammitglied weiß, ob es für das „Was“

oder das „Wie“ verantwortlich ist, wodurch Kommunikations- und Entscheidungslinien klarer werden.

Zusammengefasst erlaubt die Entkopplung, dass Änderungen in den Anforderungen nicht sofort die technische Umsetzung beeinflussen und technische Innovationen sich auf die bestmögliche Umsetzung konzentrieren, ohne die Ziele aus den Augen zu verlieren.

Was und wie als Diagramm



Aufbau des Diagramms

- **X-Achse:** Klarheit des „Wie“ (links = unklar, rechts = klar)
- **Y-Achse:** Klarheit des „Was“ (unten = unklar, oben = klar)

Interpretation der Position

- **Rechts oben:** Klare Anforderungen & klare Umsetzung → stabiler, strukturierter Ansatz
- **Oben links:** Klare Anforderungen, unklare Umsetzung → Prototyping, iterative Klärung des technischen Wegs
- **Unten rechts:** Unklare Anforderungen, klare Umsetzung → MVPs, iteratives Feedback zur Schärfung der Anforderungen
- **Links unten:** Unklare Anforderungen & unklare Umsetzung → explorativer, agiler Ansatz mit intensivem Kundenfeedback

Nutzen

- **Zeigt an**, wo Klärungsbedarf besteht (Anforderungen oder Lösungsweg)

- **Hilft bei der Wahl des Entwicklungsansatzes** (strukturiert oder flexibel)
- **Unterstützt Risikomanagement** durch frühe Erkennung von Unsicherheiten

Ein **Diagramm zur Visualisierung der Klarheit in den Dimensionen „Was“** (Anforderungen und Ziel) und „**Wie**“ (technischer Lösungsweg) kann genutzt werden, um auf einen Blick den aktuellen Status eines Projekts in Bezug auf die Anforderungen und die technische Umsetzung zu erfassen. Es hilft Projektteams, Unsicherheiten frühzeitig zu erkennen und den Entwicklungsprozess optimal zu steuern.

Aufbau des Diagramms

Das Diagramm basiert auf zwei Achsen, die jeweils die Klarheit des „Was“ und des „Wie“ abbilden:

1. X-Achse: Klarheit des „Wie“

Die X-Achse verläuft von links („unklar“) nach rechts („klar“) und zeigt, wie gut der technische Lösungsweg, also das „Wie“, definiert ist. Hier wird visualisiert, ob der technische Ansatz, die genutzten Technologien, der Aufbau der Architektur und die notwendigen Schritte zur Umsetzung klar und präzise festgelegt sind.

2. Y-Achse: Klarheit des „Was“

Die Y-Achse verläuft von unten („unklar“) nach oben („klar“) und gibt an, wie genau die Anforderungen und das Ziel des Projekts definiert sind, also das „Was“. Dies betrifft Fragen nach den gewünschten Funktionen, den Benutzerbedürfnissen, den zu erreichenden Ergebnissen und dem übergeordneten Ziel des Projekts.

Skalen und Positionierung

• Position auf der X-Achse (Klarheit des „Wie“):

Projekte, die weiter rechts auf der X-Achse liegen, haben eine präzise Vorstellung darüber, wie die Umsetzung erfolgen soll. Der technische Lösungsweg ist klar und gut durchdacht, und es besteht weniger Bedarf an experimentellen Ansätzen oder Prototypen. Ein Projektpunkt weiter links zeigt hingegen an, dass Unsicherheiten hinsichtlich der technischen Umsetzung bestehen. Hier könnten Prototyping, technische Experimente oder ein iterativer Ansatz sinnvoll sein, um die Klarheit zu erhöhen.

• Position auf der Y-Achse (Klarheit des „Was“):

Ein Punkt, der weiter oben auf der Y-Achse liegt, deutet darauf hin, dass das Projektteam ein klares Verständnis der Anforderungen und des gewünschten Ergebnisses hat. Die Ziele und Funktionalitäten sind eindeutig definiert, und es besteht Konsens über das „Was“. Ein Punkt weiter unten zeigt, dass noch Unsicherheiten bezüglich der Anforderungen bestehen. Hier ist eine iterativ-inkrementelle Entwicklung, möglicherweise mit kontinuierlichem

Kundenfeedback, hilfreich, um die Anforderungen im Verlauf des Projekts zu konkretisieren.

Interpretation der Position im Diagramm

Die Position eines Projekts im Diagramm gibt Hinweise darauf, wie viel Arbeit und Planung in die beiden Dimensionen investiert werden muss und welche Art von Entwicklungsansatz sich eignet:

- **Klarheit in beiden Dimensionen (rechts oben):**

Ein Projekt, das sich weiter rechts oben im Diagramm befindet, zeigt eine hohe Klarheit in Bezug auf Anforderungen und technische Umsetzung. Sowohl das Ziel als auch der technische Weg sind gut definiert, was eine stabile und strukturierte Vorgehensweise erlaubt. Ein klassischer Wasserfall-Ansatz oder ein geplanter agiler Ansatz kann hier geeignet sein, da die Anforderungen stabil sind und der technische Weg klar vorgegeben ist.

- **Hohe Klarheit im „Was“, aber Unsicherheit im „Wie“ (oben links):**

Wenn der Punkt des Projekts weiter oben, aber links im Diagramm liegt, sind die Anforderungen klar, jedoch besteht Unsicherheit in der technischen Umsetzung. In solchen Fällen können technische Experimente, Prototyping oder ein risikobasiertes Vorgehen sinnvoll sein, um die beste technische Lösung zu finden. Hier kann ein iterativer, agiler Prozess hilfreich sein, um Lösungen zu testen und schrittweise die Klarheit im „Wie“ zu erhöhen.

- **Hohe Klarheit im „Wie“, aber Unsicherheit im „Was“ (unten rechts):**

Ein Punkt weiter rechts, aber unten, zeigt, dass der technische Lösungsweg klar ist, während die Anforderungen noch unklar sind. Hier kann es sich anbieten, die Anforderungen schrittweise zu konkretisieren und durch Feedbackschleifen und frühe Tests am Benutzer die Bedürfnisse besser zu verstehen. Ein Ansatz mit Minimum Viable Products (MVPs) oder kleinen, iterativen Verbesserungen hilft hier, das „Was“ schrittweise zu schärfen.

- **Unsicherheit in beiden Dimensionen (links unten):**

Befindet sich das Projekt links unten, sind sowohl die Anforderungen als auch der technische Lösungsweg noch unklar. Ein stark explorativer und agiler Ansatz ist hier notwendig, um sowohl das „Was“ als auch das „Wie“ schrittweise zu definieren. In diesem Fall sind Methoden wie Design Thinking, intensives Prototyping und kontinuierliches Kundenfeedback sehr hilfreich, um die Klarheit beider Dimensionen zu verbessern.

Anpassung des Entwicklungsansatzes

Die Position des Projektpunkts im Diagramm zeigt somit an, welche Dimension noch weiter geklärt werden muss und wie flexibel oder strukturiert der Entwicklungsprozess gestaltet werden sollte. Projekte weiter oben und rechts im

Diagramm sind bereit für eine planbare Umsetzung, während Projekte mit einer Position weiter unten oder links auf Flexibilität und iterative Klärung setzen sollten.

Nutzen des Diagramms

Dieses Diagramm bietet einen schnellen Überblick über den Klarheitsgrad eines Projekts in den Dimensionen „Was“ und „Wie“. Es ermöglicht den Projektteams, auf einen Blick zu erkennen, in welchen Bereichen Unsicherheiten bestehen und welche methodischen Ansätze geeignet sind, um die Klarheit zu erhöhen. So wird die Planung und Umsetzung gezielt an die spezifischen Bedürfnisse des Projekts angepasst, und Risiken können frühzeitig identifiziert und minimiert werden.

Das "Was" und das "Wie" spielerisch erleben

Ein kleines interaktives Programm, in dem der Benutzer durch die Dimensionen „Was“ und „Wie“ navigiert und seine Antworten bewertet, um zu sehen, in welchem Bereich des Diagramms sein Projekt liegt. Am Ende wird automatisch eine Empfehlung für den Entwicklungsansatz angezeigt.

Hier ist ein einfaches Beispiel für ein solches Programm:

1. **Setup:** Erstelle ein interaktives Widget, das dem Benutzer Fragen zur Klarheit des „Was“ und „Wie“ stellt.
2. **Bewertung:** Lasse den Benutzer für jede Frage Punkte von 1 bis 5 vergeben, wobei 1 „sehr unklar“ und 5 „sehr klar“ bedeutet.
3. **Berechnung und Feedback:** Basierend auf den Eingaben wird das Projekt im Diagramm positioniert und eine passende Empfehlung für den Entwicklungsansatz angezeigt.
4. **Ergebnis visualisieren:** Ein Diagramm zeigt den Punkt für das Projekt an der berechneten Position im Klarheitsdiagramm.

Hier ist ein Beispielcode für die Umsetzung:

```
In [2]: import ipywidgets as widgets
from IPython.display import display
import matplotlib.pyplot as plt

# Interaktive Fragen erstellen
question_what = widgets.IntSlider(value=3, min=1, max=5, step=1, description="Was")
question_how = widgets.IntSlider(value=3, min=1, max=5, step=1, description="Wie")
submit_button = widgets.Button(description="Projekt positionieren")

# Anzeige der Fragen
display(question_what, question_how, submit_button)

# Funktion zur Berechnung und Ausgabe der Empfehlung
def evaluate_project(event):
    # Werte des „Was“ und „Wie“ erfassen
```

```

what_score = question_what.value
how_score = question_how.value

# Diagramm zeichnen
plt.figure(figsize=(6, 6))
plt.plot(what_score, how_score, 'bo', markersize=10)
plt.xlim(1, 5)
plt.ylim(1, 5)
plt.xlabel("Klarheit des 'Wie''")
plt.ylabel("Klarheit des 'Was''")
plt.title("Projektposition im Klarheitsdiagramm")
plt.grid(True)
plt.show()

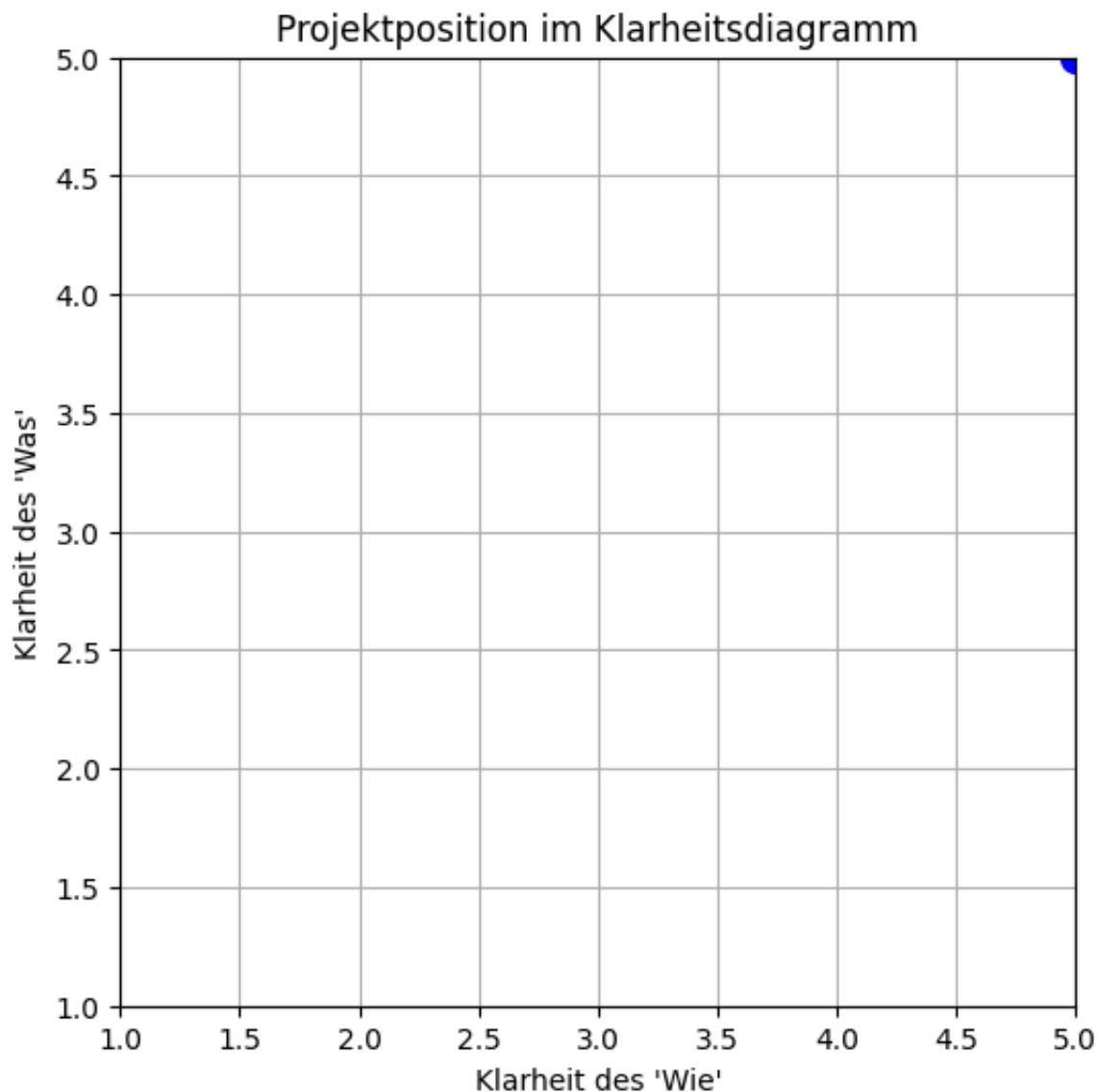
# Empfehlung basierend auf der Position
if what_score >= 4 and how_score >= 4:
    recommendation = "Stabiler, strukturierter Ansatz empfohlen."
elif what_score >= 4 and how_score < 4:
    recommendation = "Iteratives Prototyping für technische Klärung s"
elif what_score < 4 and how_score >= 4:
    recommendation = "Iterative Klärung der Anforderungen mit MVPs un"
else:
    recommendation = "Explorativer Ansatz mit intensivem Feedback emp"

# Ausgabe der Empfehlung
print("Empfehlung:", recommendation)

# Button-Click-Event verknüpfen
submit_button.on_click(evaluate_project)

IntSlider(value=3, description="Klarheit des 'Was'", max=5, min=1)
IntSlider(value=3, description="Klarheit des 'Wie'", max=5, min=1)
Button(description='Projekt positionieren', style=ButtonStyle())

```



Empfehlung: Stabiler, strukturierter Ansatz empfohlen.

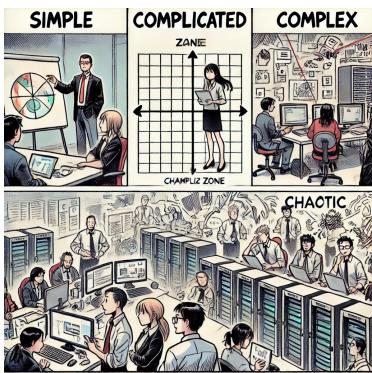
Erläuterung des Codes:

- **Widgets:** Der Benutzer kann die Klarheit des „Was“ und „Wie“ mithilfe von Schieberegbern bewerten.
- **Positionierung:** Der Code platziert die Projektposition auf einem Diagramm mit der Skala von 1 bis 5 auf beiden Achsen.
- **Empfehlung:** Basierend auf der Position des Projekts wird eine Empfehlung für den Entwicklungsansatz angezeigt.

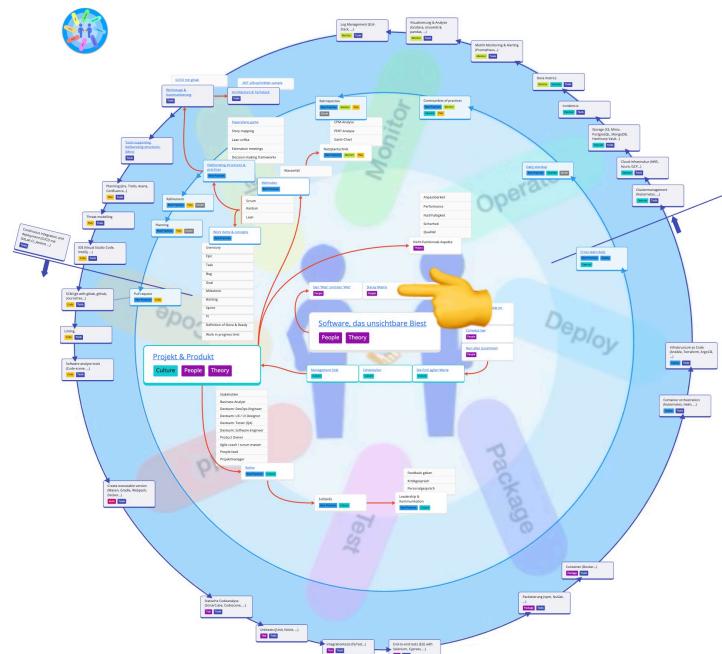
Stacey_Matrix.de

[Das unsichtbare Biest / Das "Was" und das "Wie"](#)

Stacey Matrix



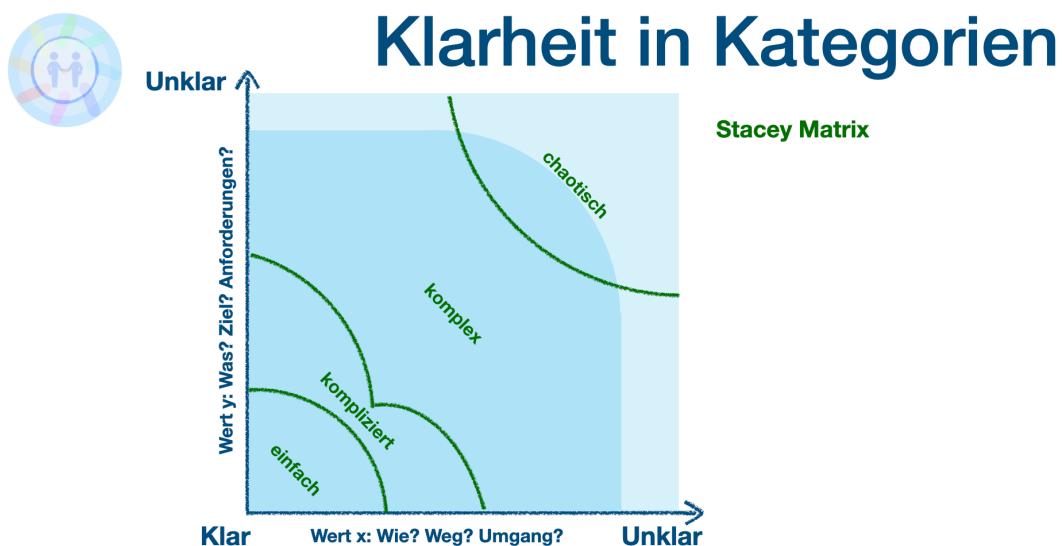
Die Stacey-Matrix einfach erklärt - Projekte leicht gemacht
 Stacey Matrix • einfach erklärt • Ansätze für jeweilige Projekte - Studyflix
 Stacey-Matrix - kurz erklärt - projektmagazin
 Die Stacey-Matrix - Bundesverwaltungsamt
 Stacey Matrix - Erklärung (deutsch) - Digitale Neuordnung
 Stacey Matrix: Definition und Bedeutung im Projektmanagement! - Asana
 Stacey-Matrix: Ganz einfach die richtige Projektmanagement-Methode finden - Erfolgreich Projekte Leiten
 Stacey Matrix: So wählst Du Deine Methode für Dein Projekt - factro
 Stacey-Matrix: Tool für effektives Projektmanagement - Personio
 Stacey Matrix: Definition und Bedeutung im Projektmanagement - Qvest





Quelle: https://en.wikipedia.org/wiki/Ralph_D._Stacey (9.11.2024)

Definition: Tool zur Einordnung von Projekten nach Klarheit der Anforderungen und Gewissheit der Lösung.



- **Achsen:**
 - **Anforderungen** (What): Von klar bis unklar.
 - **Lösung** (How): Von sicher bis unsicher.
- **Kategorien:**
 - **Einfach (Simple)**: Klarheit über Anforderungen und Lösung.
 - **Vorgehen**: Standardisierte Prozesse, Wasserfallmodell.
 - **Anwendung**: Routinelösungen, z. B. Verwaltungssysteme.
 - **Kompliziert (Complicated)**: Anforderungen klar, Lösung erfordert Expertise.
 - **Vorgehen**: Experten-Analyse, Scrum, Kanban.

- **Anwendung:** Neue Features in bestehenden Systemen.
- **Komplex (Complex):** Anforderungen und Lösung unklar.
 - **Vorgehen:** Agile Methoden, Iterationen, Feedback.
 - **Anwendung:** Innovative Produkte, z. B. neue Apps.
- **Chaotisch (Chaos):** Anforderungen und Lösung völlig unklar.
 - **Vorgehen:** Kleine Experimente, Fail-fast.
 - **Anwendung:** Bahnbrechende Technologien, hohe Unsicherheit.

Die Stacey Matrix ist ein Management- und Entscheidungstool, das komplexe Probleme und Unsicherheiten visualisiert und aufzeigt, wie unterschiedliche Management- und Herangehensweisen zu verschiedenen Arten von Problemen passen. Entwickelt wurde sie von Ralph Stacey, einem britischen Professor für Management, um Organisationen zu helfen, ihre Projekte und Entscheidungen besser einzuordnen und geeignete Ansätze auszuwählen. Die Matrix ist besonders nützlich in Bereichen wie der Softwareentwicklung, wo Unsicherheiten in Anforderungen und technischen Lösungen häufig vorkommen.

Aufbau der Stacey Matrix

Die Stacey Matrix visualisiert Problemfelder entlang zweier Achsen:

1. **Grad der Klarheit der Anforderungen (What):** Reicht von „Klarheit“ bis „Unklarheit“.
2. **Grad der Gewissheit über die Lösung (How):** Reicht von „Sicher“ bis „Unsicher“.

Diese beiden Dimensionen bilden ein Koordinatensystem, das in vier grundlegende Kategorien unterteilt ist:

1. **Einfach (Simple):** Klarheit über die Anforderungen und die Lösung. Vorgehensweise und Resultat sind vorhersehbar und leicht zu managen.
2. **Kompliziert (Complicated):** Anforderungen sind klar, aber die Lösung ist schwieriger und erfordert mehr Fachwissen oder Analyse. Experten können durch Analysen Lösungen finden.
3. **Komplex (Complex):** Sowohl die Anforderungen als auch die Lösung sind unsicher. Die Projekteigenschaften ändern sich dynamisch und sind oft schwer zu prognostizieren. Häufig wird iteratives und exploratives Arbeiten benötigt.
4. **Chaotisch (Chaos):** Sowohl Anforderungen als auch die Lösung sind völlig unklar. Es gibt keine klare Richtung, und ein strukturiertes Vorgehen ist kaum möglich. Hier wird experimentelles Arbeiten empfohlen, bis sich eine Richtung ergibt.

Zusätzlich gibt es noch den **Bereich des Ungewissen** oder **Zonen der Anarchie** („Zone of Anarchy“) am äußeren Rand, wo Projekte außerhalb sinnvoller Entscheidungsgrundlagen liegen.

Anwendung auf Softwareprodukte und deren Entwicklung

Die Stacey Matrix hat für Softwareprodukte erhebliche Auswirkungen, da Softwareentwicklungsprojekte oft von Unsicherheiten geprägt sind. Dies gilt insbesondere für Produkte, bei denen Anforderungen und technische Lösungen noch nicht ausgereift sind, sich dynamisch verändern oder Neuland betreten wird. Folgende Auswirkungen können je nach Position innerhalb der Matrix entstehen:

1. Einfach (Simple)

- **Merkmale:** Anforderungen sind stabil und klar definiert; technische Lösung ist bekannt und bewährt.
- **Vorgehen:** Projekte in diesem Bereich können leicht durch standardisierte Prozesse, wie **Wasserfallmodelle** oder **stark strukturierte agile Prozesse**, abgewickelt werden.
- **Auswirkung auf Softwareprodukte:** Optimale Planung und Strukturierung möglich; geringe bis keine Iterationen nötig. Ideal für Software mit wiederkehrenden Funktionen, z. B. interne Verwaltungssysteme oder Prozessautomatisierungen, bei denen nur wenig Innovation erforderlich ist.

2. Kompliziert (Complicated)

- **Merkmale:** Anforderungen sind klar, die Lösung erfordert jedoch spezielle technische Expertise.
- **Vorgehen:** Hier eignet sich ein analytisches Vorgehen mit Expertenteams oder gezielte iterative Methoden. Frameworks wie **Scrum** oder **Kanban** können genutzt werden, wobei technische Experten eine große Rolle spielen.
- **Auswirkung auf Softwareprodukte:** Der Einsatz von Fachwissen minimiert Risiken. Produkte in diesem Bereich erfordern möglicherweise neue Funktionen oder technische Anpassungen, die komplex, aber nicht unlösbar sind. Beispiele wären neue Features in einer bestehenden Anwendung, die nur von einem spezialisierten Team umgesetzt werden können.

3. Komplex (Complex)

- **Merkmale:** Anforderungen und technische Lösung sind unklar; Feedback und iteratives Lernen sind entscheidend.
- **Vorgehen:** Agile Methoden wie **Scrum** oder **Lean Development** sind besonders effektiv. Häufige **Iterationen** und **Feedback-Loops** sind entscheidend, um Lösungen zu validieren und Anforderungen dynamisch anzupassen.
- **Auswirkung auf Softwareprodukte:** Hohe Anpassungsfähigkeit und Flexibilität notwendig. Produkte in diesem Bereich sind oft innovativ, wie neue Apps, Plattformen oder digitale Services, die kontinuierlich auf Markt- und Kundenfeedback reagieren müssen. Teams arbeiten empirisch, führen Experimente durch und passen die Lösung schrittweise an. Agile Prinzipien bieten eine gute Möglichkeit, den ständigen Wandel zu managen.

4. Chaotisch (Chaos)

- **Merkmale:** Es gibt weder Klarheit über Anforderungen noch über die technische Lösung.
- **Vorgehen:** In dieser Zone ist es oft am besten, kleine Experimente durchzuführen, um Orientierung zu gewinnen. Ein *Fail-fast*-Ansatz kann sinnvoll sein, um schnell herauszufinden, was funktioniert und was nicht.
- **Auswirkung auf Softwarereprodukte:** Bei Softwareprodukten in dieser Zone kann es sich um bahnbrechende, radikal neue Technologien handeln, deren Marktpotenzial und technische Umsetzung unklar sind. Solche Projekte sollten möglichst risikominimierend durchgeführt werden, indem Experimente und Tests frühzeitig validiert werden. Projekte könnten schnell eingestellt oder drastisch umstrukturiert werden, wenn sich eine Richtung als unbrauchbar erweist.

Fazit

Die Stacey Matrix ist ein wertvolles Tool, um die richtige Projekt- und Produktstrategie in der Softwareentwicklung zu bestimmen. Indem sie hilft, Komplexität und Unsicherheit zu identifizieren, ermöglicht sie es, die Herangehensweise anzupassen und Ressourcen optimal einzusetzen. Für Softwareentwickler und Projektmanager ist die Matrix eine nützliche Orientierungshilfe, um zu entscheiden, wann standardisierte Prozesse und wann exploratives, iteratives Arbeiten erforderlich ist.

Stacey Matrix spielerisch erleben

Ein interaktives Programm in einem Jupyter Notebook könnte die Stacey Matrix auf spielerische Weise vermitteln, indem es den Benutzer dazu auffordert, verschiedene Szenarien in Bezug auf Anforderungs- und Lösungssicherheit zu bewerten. Abhängig von den Antworten des Benutzers wird das Projekt in eine Kategorie der Matrix eingeordnet und eine geeignete Vorgehensweise vorgeschlagen.

Hier ein Beispiel:

```
In [ ]: import ipywidgets as widgets
from IPython.display import display, clear_output

# Labels für Anforderungen und Lösungssicherheit
anforderungen_label = widgets.Label("Anforderungen:")
loesung_label = widgets.Label("Lösungssicherheit:")

# Dropdown-Widgets für Anforderungen und Lösungssicherheit ohne Beschreibung
anforderungen_dropdown = widgets.Dropdown(
    options=["Klar", "Unklar", "Sehr unklar"],
    layout=widgets.Layout(width='300px')
)
loesung_dropdown = widgets.Dropdown(
    options=["Sicher", "Unsicher", "Sehr unsicher"],
```

```

        layout=widgets.Layout(width='300px')
    )

# Button-Widget zur Bestätigung
button = widgets.Button(description="Analyse durchführen", layout=widgets.Layout(width='200px'))

# Ausgabe-Widget
output = widgets.Output()

# Funktion zur Analyse basierend auf der Stacey Matrix
def analyze_project(button):
    with output:
        clear_output()
        anforderungen = anforderungen_dropdown.value.lower()
        loesung = loesung_dropdown.value.lower()

        if anforderungen == "klar" and loesung == "sicher":
            result = "Einfach (Simple) – Standardprozess wie Wasserfallmodell"
        elif anforderungen == "klar" and loesung == "unsicher":
            result = "Kompliziert (Complicated) – Experten-Analyse oder Konsens"
        elif anforderungen == "unklar" and loesung == "unsicher":
            result = "Komplex (Complex) – Agile Methoden, Scrum oder Lean"
        elif anforderungen == "sehr unklar" or loesung == "sehr unsicher":
            result = "Chaotisch (Chaos) – Experimentieren und fail-fast-Ändern"
        else:
            result = "Zone der Anarchie – Experimentieren und Risikomanagement"

        print("Empfohlener Ansatz:", result)

# Button mit der Funktion verknüpfen
button.on_click(analyze_project)

# Widgets und Ausgabe anzeigen
display(anforderungen_label, anforderungen_dropdown)
display(loesung_label, loesung_dropdown)
display(button)
display(output)

```

Label(value='Anforderungen:')
 Dropdown(layout=Layout(width='300px'), options=('Klar', 'Unklar', 'Sehr unklar'), value='Klar')
 Label(value='Lösungssicherheit:')
 Dropdown(layout=Layout(width='300px'), options=('Sicher', 'Unsicher', 'Sehr unsicher'), value='Sicher')
 Button(description='Analyse durchführen', layout=Layout(width='200px'), style=ButtonStyle())
 Output()

Erläuterung des Codes

- Dropdown-Menüs:** Die Dropdowns `anforderungen_dropdown` und `loesung_dropdown` lassen den Benutzer die Anforderungen und Lösungssicherheit auswählen.
- Button-Widget:** Der Button „Analyse durchführen“ startet die Analyse.
- Funktion `analyze_project`:** Diese Funktion liest die Auswahl, führt die

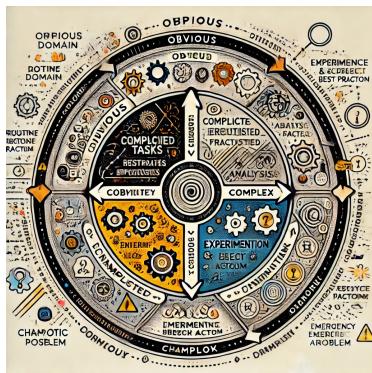
Entscheidung basierend auf der Stacey Matrix durch und gibt die Empfehlung im output -Widget aus.

4. **Anzeige:** Alle Widgets und das Ausgabe-Widget werden am Ende mit `display` zusammen angezeigt.

Cynefin_model.de

Das unsichtbare Biest / Das "Was" und das "Wie" / Stacey Matrix

Cynefin-Modell in der Softwareentwicklung



Cynefin-Framework – Wikipedia

Das Cynefin-Framework einfach erklärt - Projekte leicht gemacht

Cynefin - Cynefin.io

Das Cynefin Framework: Definition und Erklärung! –

Asana

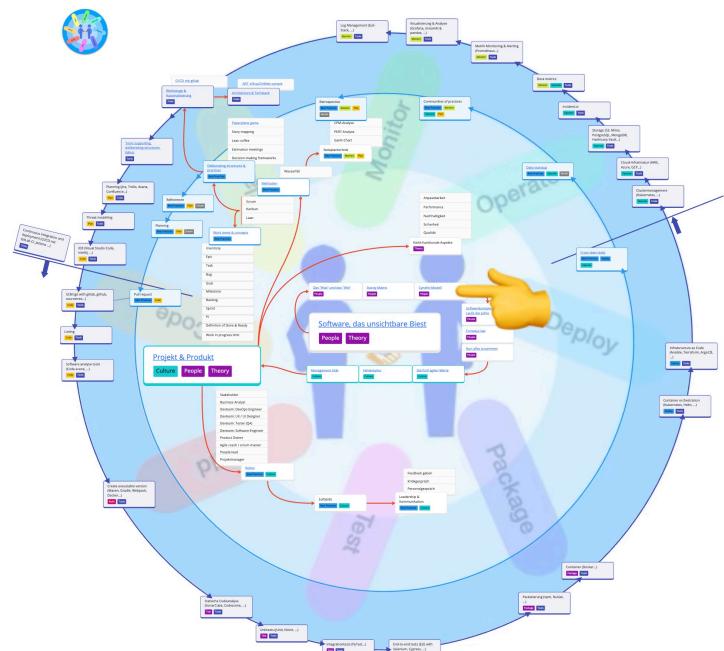
Das Cynefin Framework: Bereit für einen neuen Blick auf Veränderung? - Agile Scrum Group

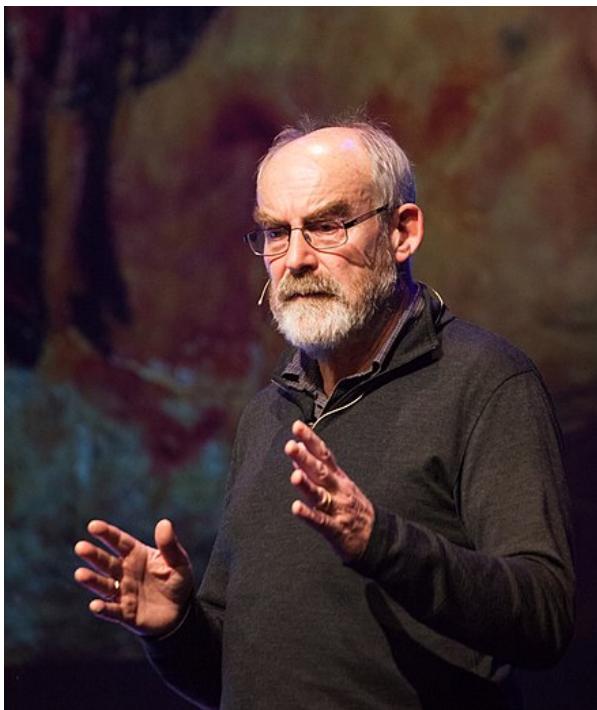
Scrum Group

CYNEFIN Modell nach Dave Snowden von Dr. Patrick Fritz

Cynefin Framework - Einsatz, Tipps & Vorlage | Consulting LIFE Cynefin Framework:

Cynefin Framework:
Definition & Erläuterung -
PURE Consultant
Das Cynefin-Modell - Dr.
Andreas Kolmer
Cynefin-Framework -
Entscheidungsfindung und





Quelle: https://en.wikipedia.org/wiki/Dave_Snowden (9.11.2024)

Das **Cynefin-Modell** (ausgesprochen: kuh-NEV-in) ist ein Framework, das ursprünglich vom walisischen Forscher Dave Snowden entwickelt wurde und hilft, verschiedene Arten von Problemen zu kategorisieren und darauf aufbauend die geeigneten Entscheidungsstrategien und Handlungsweisen zu wählen. Im Kontext der Softwareentwicklung bietet das Modell einen nützlichen Rahmen, um komplexe Entscheidungen zu treffen und geeignete Methoden zur Problemlösung und Projektsteuerung zu wählen.

Cynefin ist ein walisisches Wort, das üblicherweise im Deutschen mit 'Lebensraum' oder 'Platz' übersetzt wird, obwohl diese Übersetzung nicht seine volle Bedeutung vermitteln kann. Eine vollständige Übersetzung des Wortes würde aussagen, dass wir alle mehrere Vergangenheiten haben, derer wir nur teilweise bewusst sein können: kulturelle, religiöse, geographische, stammesgeschichtliche usw.



1. Einfach (Obvious)

- *Merkmale*: Ursache-Wirkung klar, Best Practices vorhanden
- *Vorgehen*: „Erkennen – Kategorisieren – Reagieren“
- *Beispiele*: Wartungsaufgaben, Einhaltung von Coding-Standards

2. Kompliziert (Complicated)

- *Merkmale*: Ursache-Wirkung nicht sofort klar, Fachwissen erforderlich
- *Vorgehen*: „Erkennen – Analysieren – Reagieren“
- *Beispiele*: Architekturdesign, Performance-Optimierung

3. Komplex (Complex)

- *Merkmale*: Ursache-Wirkung retrospektiv erkennbar, Experimentation notwendig
- *Vorgehen*: „Probieren – Erkennen – Reagieren“
- *Beispiele*: Entwicklung neuer Features, Prototypen-Tests

4. Chaotisch (Chaotic)

- *Merkmale*: Keine klare Ursache-Wirkung, schnelle Reaktion erforderlich
- *Vorgehen*: „Handeln – Erkennen – Reagieren“
- *Beispiele*: Notfall-Bugfixes, Sicherheitsverletzungen

5. Unordentlich (Disorder)

- *Merkmale*: Kategorie unklar, Diagnosephase nötig
- *Beispiele*: Unklare Anfangsphase von Projekten

Nutzen im Projektmanagement

- *Methodenanpassung*: Wahl passender Methoden für die jeweilige Kategorie
- *Risiko-Management*: Kategorisierung zur besseren Risikoeinschätzung
- *Agilität*: Iterative Methoden im komplexen Bereich

Das Modell unterscheidet fünf Bereiche, die durch die Natur und das Verständnis der Probleme und Systeme geprägt sind:

1. Einfach (Obvious):

- Probleme in diesem Bereich sind klar und einfach zu verstehen. Ursache und Wirkung sind offensichtlich und die Lösung kann auf bewährte Best Practices zurückgreifen. In der Softwareentwicklung betrifft dies oft routinemäßige Aufgaben, wie das Debuggen von gut bekannten Fehlern, die Standardimplementierung von bewährten Mustern oder das Anwenden von Coding-Standards.
- **Vorgehensweise**: „Sense – Categorize – Respond“ (Erkennen – Kategorisieren – Reagieren).
- **Beispiele**: Standardisierte Wartungsaufgaben, Einhaltung von Coding-

Guidelines und Durchführung von klar definierten Testfällen.

2. Kompliziert (Complicated):

- Hier sind die Beziehungen zwischen Ursache und Wirkung zwar nicht sofort offensichtlich, aber durch Analyse herauszufinden. Für diese Art von Problemen ist Fachwissen erforderlich. Softwarearchitektur-Design, die Optimierung von Algorithmen oder das Einführen neuer Frameworks fallen oft in diesen Bereich, da Expertise und Erfahrung erforderlich sind.
- **Vorgehensweise:** „Sense – Analyze – Respond“ (Erkennen – Analysieren – Reagieren).
- **Beispiele:** Design-Entscheidungen zur Skalierbarkeit oder Performance-Optimierung, Auswahl eines geeigneten Technologie-Stacks, Einführung neuer Frameworks.

3. Komplex (Complex):

- In diesem Bereich ist Ursache und Wirkung erst im Nachhinein (retrospektiv) erkennbar. In der Softwareentwicklung trifft dies oft auf die Implementierung neuer, innovativer Features zu, die zuvor in dieser Form noch nicht umgesetzt wurden. Iterative Ansätze wie agile Methoden oder Scrum eignen sich gut für komplexe Umgebungen, da sie durch kurze Iterationen und regelmäßige Rückmeldungen helfen, auf Veränderungen zu reagieren und den Entwicklungsprozess anzupassen.
- **Vorgehensweise:** „Probe – Sense – Respond“ (Probieren – Erkennen – Reagieren).
- **Beispiele:** Entwicklung neuartiger Features, das Testen von Prototypen oder Proof of Concepts, Umstellungen auf neue Softwarearchitekturen.

4. Chaotisch (Chaotic):

- In chaotischen Situationen gibt es keine Beziehung zwischen Ursache und Wirkung. Hier steht das Problem oft plötzlich und unvorhergesehen auf der Agenda, etwa bei Krisen oder akuten Zwischenfällen. In der Softwareentwicklung tritt dieser Zustand bei schwerwiegenden Bugs in Produktivsystemen auf, die schnelles Eingreifen und unmittelbare Lösungsansätze erfordern.
- **Vorgehensweise:** „Act – Sense – Respond“ (Handeln – Erkennen – Reagieren).
- **Beispiele:** Notfall-Reparaturen, schnelle Bugfixes, Reaktionen auf Sicherheitsverletzungen.

5. Unordentlich (Aporetisch/Disorder):

- Diese Kategorie beschreibt einen Zustand, in dem unklar ist, welche der vier vorherigen Kategorien zutrifft. Es handelt sich um eine Zwischenzone, in der das Problem noch nicht vollständig verstanden wird. Hier wird oft eine Diagnosephase benötigt, um festzustellen, in welche Kategorie das Problem tatsächlich gehört.

- **Beispiele:** Anfangsphasen von Projekten oder Anforderungen, bei denen nicht klar ist, wie der Umfang oder die genaue Problemstellung aussieht.

Anwendung des Cynefin-Modells in der Softwareentwicklung

In der Praxis kann das Cynefin-Modell Softwareentwicklungsteams helfen, Situationen und Herausforderungen besser einzuordnen und darauf basierend die beste Handlungsweise auszuwählen:

- **Team-Zusammensetzung und Entscheidungsfindung:** Die Bereiche des Cynefin-Modells bieten Orientierung für die Auswahl von Methoden und Ressourcen. In komplexen Bereichen eignen sich beispielsweise interdisziplinäre Teams und agile Methoden, während in einfachen und komplizierten Bereichen ein erfahrener Experte für klare, analytische Entscheidungen geeignet ist.
- **Methodenanpassung:** Projekte können in verschiedenen Phasen in unterschiedliche Kategorien fallen. Ein neues, unerprobtes Feature kann komplex beginnen und wird möglicherweise nach mehreren Iterationen und Tests zu einer komplizierten oder einfachen Aufgabe. Das Modell hilft dabei, den Entwicklungsansatz dynamisch an den Bereich anzupassen, in dem sich das Team befindet.
- **Risiko-Management und Planung:** Durch das frühzeitige Erkennen, in welcher Kategorie sich eine Aufgabe oder ein Problem befindet, können Risiken im Projektmanagement besser eingeordnet werden. Chaotische oder komplexe Bereiche erfordern beispielsweise andere Planungs- und Risikomanagement-Strategien als einfache oder komplizierte Aufgaben.
- **Agilität und Experimentation:** Im Bereich der komplexen Probleme zeigt das Modell, warum Experimente und Feedback-Schleifen entscheidend sind. Teams sollten iterativ vorgehen und die Möglichkeit zur Anpassung bewahren, um die besten Lösungen für unklare Probleme zu finden.

Insgesamt ist das Cynefin-Modell ein mächtiges Werkzeug in der Softwareentwicklung, das Entscheidungsträgern dabei hilft, Probleme zu strukturieren und systematisch zu lösen.

Cynefin-Modell spielerisch kennenlernen

Eine spielerische Möglichkeit zur Erkundung des Cynefin-Modells könnte eine interaktive Zuordnungsaufgabe sein, bei der der Benutzer Beispiele zu den richtigen Kategorien des Modells zuordnen muss. Nach der Zuordnung kann das Programm die Eingaben des Benutzers überprüfen und Feedback geben.

Hier ein Beispiel:

```
In [2]: import ipywidgets as widgets
from IPython.display import display, clear_output

# Beschreibungen für jedes Cynefin-Feld
questions = {
    "Erfordert tiefere Analyse, wie etwa das Design einer neuen Softwarearchitektur": "Unklare Anforderungen, bei denen der Bereich noch nicht identifiziert ist.",
    "Die Entwicklung neuartiger Features, die regelmäßige Feedback-Schleifen erfordern": "Die Entwicklung neuartiger Features, die regelmäßige Feedback-Schleifen erfordern.",
    "Ein Beispiel für Routineaufgaben, wie das Debuggen einfacher Fehler": "Ein Beispiel für Routineaufgaben, wie das Debuggen einfacher Fehler.",
    "Notfall-Bugfixes oder Sicherheitsverletzungen, die sofortiges Eingreifen benötigen": "Notfall-Bugfixes oder Sicherheitsverletzungen, die sofortiges Eingreifen benötigen."
}

# Dropdown-Menüs für die Antworten ohne descriptions, stattdessen mit Labels
options = ["Einfach", "Kompliziert", "Komplex", "Chaotisch", "Unordentlich"]
dropdowns = {question: widgets.Dropdown(options=options, layout=widgets.Layout(width="150px")) for question in questions}

# Überprüfungsfunction
def check_answers(b):
    clear_output()
    score = 0
    total = len(questions)
    for question, dropdown in dropdowns.items():
        answer = dropdown.value
        if answer == questions[question]:
            score += 1
            print(f"Korrekt: {question}")
        else:
            print(f"Falsch: {question} (Ihre Antwort: {answer}, Richtig: {questions[question]})")
    print(f"\nIhr Ergebnis: {score}/{total} richtig.")

# Button zum Überprüfen der Antworten
check_button = widgets.Button(description="Überprüfen")
check_button.on_click(check_answers)

# Anzeige der Labels, Dropdowns und des Buttons
for question, dropdown in dropdowns.items():
    display(widgets.Label(value=question)) # Frage als Label
    display(dropdown) # Dropdown-Menü darunter
    display(check_button)
```

Korrekt: Erfordert tiefere Analyse, wie etwa das Design einer neuen Softwarearchitektur.

Korrekt: Unklare Anforderungen, bei denen der Bereich noch nicht identifiziert ist.

Korrekt: Die Entwicklung neuartiger Features, die regelmäßige Feedback-Schleifen erfordern.

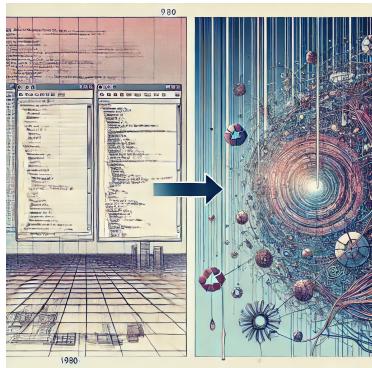
Korrekt: Ein Beispiel für Routineaufgaben, wie das Debuggen einfacher Fehler.

Korrekt: Notfall-Bugfixes oder Sicherheitsverletzungen, die sofortiges Eingreifen benötigen.

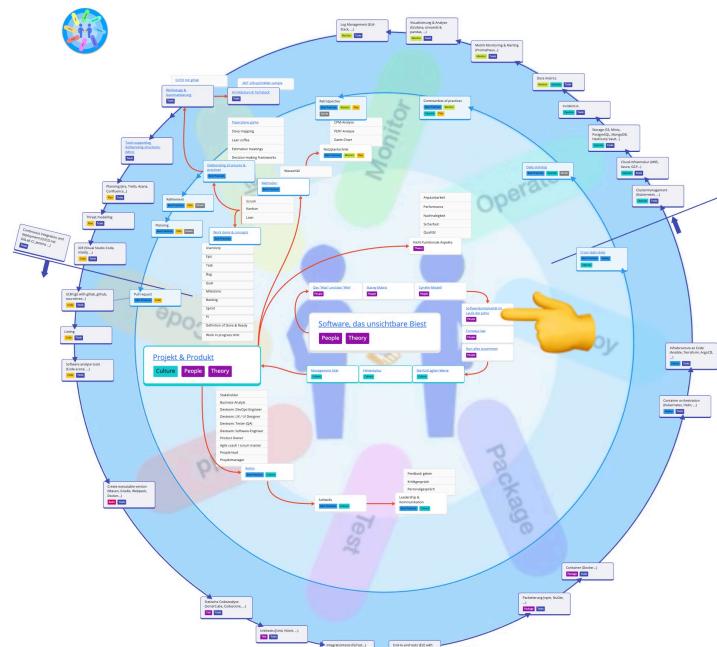
Ihr Ergebnis: 5/5 richtig.

Complexity_over_the_years.de

Komplexität von Software im Laufe der Jahre



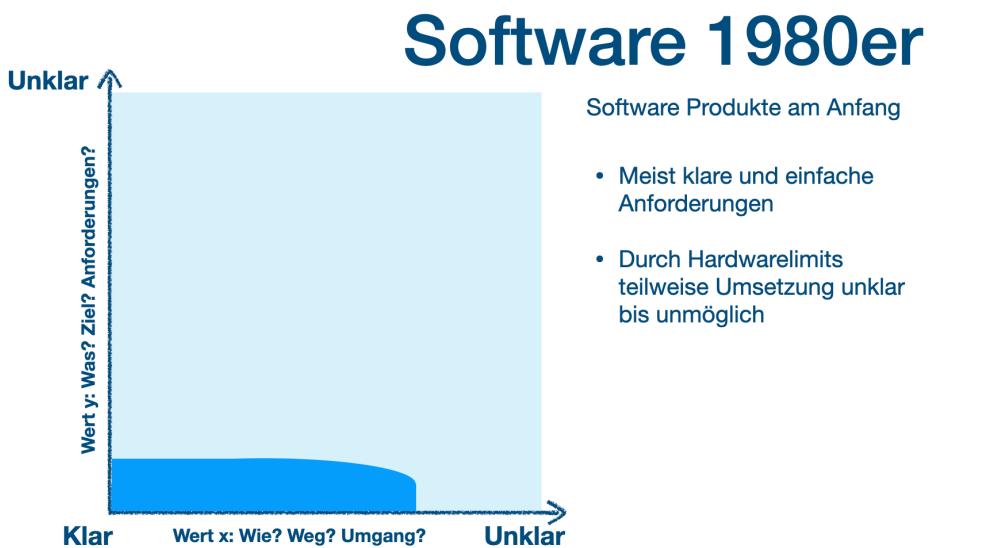
Programming Complexity - Wikipedia
Why Embracing Complexity Is the Real Challenge in Software Today - MIT Technology Review
How and Why We End Up with Complex Methods: A Multi-Language Study - SpringerLink
Measuring Software Complexity: What Metrics to Use? - The Valuable Dev Research Trends in Structural Software Complexity - arXiv.org
Historical Survey: Crises, Crises, Everywhere; Any End to This ... - SpringerLink
Comprehensive Model-Driven Complexity Metrics for Software Systems ... - IEEE Xplore
Structural Epochs in the Complexity of Software over Time - IEEE Xplore
A Survey on Metric of Software Complexity - IEEE Xplore



1980er Jahre: Personal Computer-Ära

Was: Desktop-Software (z. B. Textverarbeitung, Spiele) für Einzelplatzanwendungen

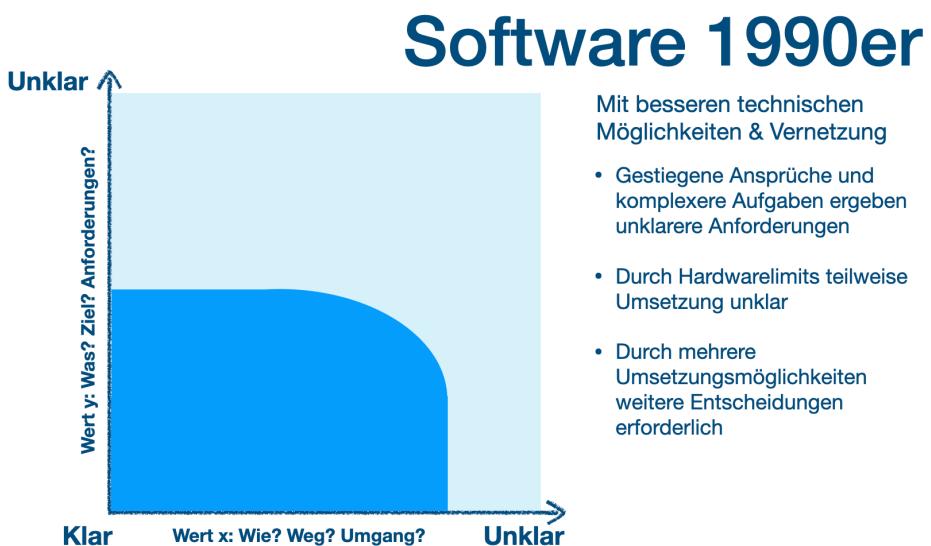
Wie: Monolithische Entwicklung, einfache Sprachen (C, Pascal), geringe Teamarbeit



1990er Jahre: Internet & Client-Server-Architektur

Was: Netzwerk- und Webanwendungen, z. B. Browser und Client-Server-Modelle

Wie: Objektorientierte Programmierung (Java, C++), erste Versionierungstools (CVS), größere Teams



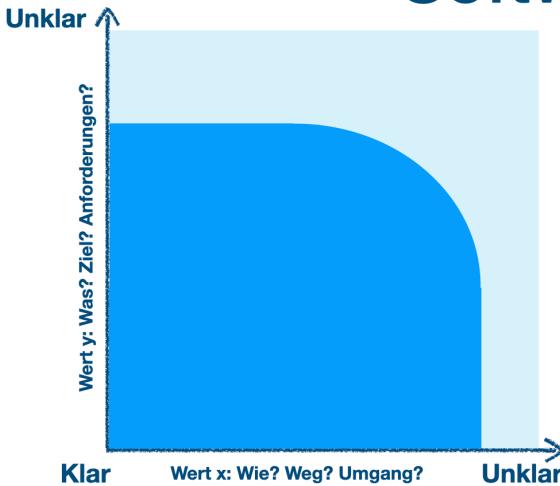
2000er Jahre: Web 2.0 und verteilte Systeme

Was: Interaktive, skalierbare Webanwendungen (soziale Netzwerke, E-Commerce)

Wie: Agile Methoden (Scrum), Frameworks (Ruby on Rails), Cloud-Computing



Software 2000er



Mit mobilen Geräte & mobilen Netzen

- Weiter gestiegene Ansprüche ergeben noch unklarere Anforderungen
- Fast verschwundene Hardwarelimits
- Eine Vielzahl an Werkzeugen, Bibliotheken
- Eine Vielzahl von Spezialisierungen bei Mitarbeitern

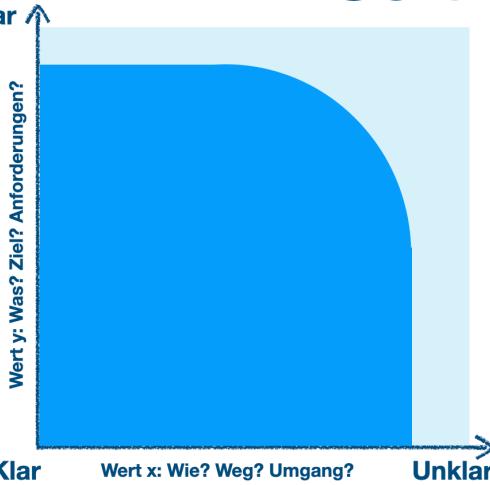
2010er Jahre: Microservices & DevOps

Was: Skalierbare, unabhängige Microservices und mobile Anwendungen

Wie: DevOps, CI/CD, Container (Docker, Kubernetes), Observability-Tools



Software 2010er



Durch cloudbasierte und z.B. künstlich intelligente Anwendungen

- Weiter gestiegene Ansprüche, noch unklarere Anforderungen
- Enormen Datenmengen, weitere Anforderungen, z.B. BigData, Geodaten
- Weitere Spezialisierung von Personal und Technik

2020er Jahre: Künstliche Intelligenz & Automatisierung

Was: KI-basierte Anwendungen (z. B. Chatbots, Bilderkennung), IoT und Edge-Computing

Wie: KI-Frameworks (TensorFlow, PyTorch), MLOps, automatisierte Sicherheit (Zero-Trust)

Seit den 1980er Jahren hat sich die Komplexität von Software in vielen Dimensionen kontinuierlich erhöht, was sowohl die Funktionalität und Anwendungsfälle ("Was") als

auch die Methoden und Werkzeuge zur Entwicklung und Verwaltung dieser Software ("Wie") betrifft. Hier ist ein Überblick, wie sich die Softwarelandschaft seitdem verändert hat.

1980er Jahre: Beginn der Personal Computer-Ära

Was:

- Software wurde in den 1980ern primär für Desktop-Umgebungen und einfache Benutzeranwendungen entwickelt, z. B. Textverarbeitungsprogramme, Tabellenkalkulationen und einfache Spiele.
- Komplexität zeigte sich hauptsächlich in der Funktionalität und Benutzerfreundlichkeit. Ziel war es, Software für den individuellen Gebrauch auf Personal Computern zugänglich zu machen.

Wie:

- Programmiersprachen wie C, Pascal und BASIC waren dominant. C wurde vor allem im System- und Betriebssystembereich verwendet, Pascal häufig in der Bildung.
- Die Softwareentwicklung erfolgte meist monolithisch: Ein einzelnes Programm wurde als Ganzes kompiliert und ausgeführt.
- Versionskontrolle und Zusammenarbeit standen noch am Anfang. Meistens arbeitete ein Entwickler oder ein kleines Team an einem Programm, ohne ausgereifte Methoden zur Teamkoordination.

Beispiel:

- MS-DOS, ein relativ einfaches Betriebssystem, bot eine textbasierte Schnittstelle ohne Multitasking und ohne moderne Speicherverwaltung. Die Komplexität lag primär in der Hardwareinteraktion und Benutzerfreundlichkeit.

1990er Jahre: Aufkommen des Internets und der Client-Server-Architektur

Was:

- Die Softwareentwicklung verlagerte sich von reinen Desktop-Anwendungen hin zu Netzwerkanwendungen, was durch das Aufkommen des Internets und den Client-Server-Modellen angetrieben wurde.
- Neue Arten von Software wie Webbrowser (z. B. Netscape) und Client-Server-Anwendungen in Unternehmen führten zu einer erhöhten Komplexität.
- Auch Datenbanken und serverseitige Applikationen gewannen an Bedeutung.

Wie:

- Die Einführung objektorientierter Programmierung (OOP) in Sprachen wie Java

und C++ erleichterte die Modellierung komplexer Anwendungen.

- Die Strukturierung in Client-Server-Modelle führte dazu, dass Applikationen in mehrere Komponenten (Client und Server) unterteilt wurden, die miteinander kommunizierten.
- Es entstanden erste Versionierungstools wie CVS, die die Zusammenarbeit in Teams förderten.

Beispiel:

- Netscape Navigator ermöglichte den Zugang zum World Wide Web, wobei eine Client-Server-Architektur eingesetzt wurde. Die Komplexität lag nun in der Netzwerkkommunikation und der Handhabung dynamischer Inhalte.

2000er Jahre: Web 2.0 und verteilte Systeme

Was:

- Der Begriff "Web 2.0" steht für die Entwicklung interaktiver, dynamischer Webanwendungen. Plattformen wie soziale Netzwerke, E-Commerce, und cloudbasierte Software gewannen an Bedeutung.
- Die zunehmende Vernetzung und Globalisierung erforderten Skalierbarkeit und Hochverfügbarkeit, die durch verteilte Systeme ermöglicht wurden.
- Die Software entwickelte sich von einfachen Webseiten zu komplexen Webanwendungen mit Echtzeit-Interaktion und Personalisierung.

Wie:

- Sprachen wie JavaScript und Frameworks wie Ruby on Rails ermöglichen die Entwicklung dynamischer und interaktiver Webanwendungen.
- Die Verwendung von Datenbanken und Caching-Systemen zur Speicherung und schnellen Bereitstellung großer Datenmengen wurde Standard.
- Methoden wie Agile und Scrum setzten sich durch, um die Entwicklung in schnell veränderlichen Umfeldern zu unterstützen.
- Cloud-Computing-Anbieter wie Amazon Web Services (AWS) machten es möglich, skalierbare und verteilte Systeme zu betreiben.

Beispiel:

- Facebook und andere soziale Netzwerke entwickelten sich zu komplexen Webplattformen, die Benutzerinteraktionen in Echtzeit verarbeiten mussten. Die Herausforderung bestand darin, sowohl Personalisierung als auch Skalierbarkeit zu gewährleisten.

2010er Jahre: Microservices und DevOps

Was:

- Die Anwendungen wurden durch die fortschreitende Digitalisierung noch umfangreicher, mit einer Vielzahl von Funktionen und Integrationen (z. B. Zahlungssysteme, Benachrichtigungen).
- Mobile Apps traten verstärkt in den Vordergrund, was zu einer Diversifizierung der Plattformen (iOS, Android, Web) führte.
- Anwendungen wurden in immer kleinere Einheiten aufgeteilt (Microservices), um die Komplexität beherrschbar zu halten und eine schnellere Skalierung und Wartung zu ermöglichen.

Wie:

- DevOps und CI/CD (Continuous Integration/Continuous Deployment) wurden als Antwort auf die Notwendigkeit entwickelt, schnell und zuverlässig Updates bereitzustellen.
- Container-Technologien wie Docker und Kubernetes ermöglichen eine plattformunabhängige Bereitstellung von Anwendungen.
- Es entstanden Observability-Tools wie Prometheus und Grafana, um den Zustand verteilter Systeme zu überwachen und Fehler schneller zu finden.

Beispiel:

- Netflix führte die Microservices-Architektur ein, die es ermöglichte, dass verschiedene Entwicklerteams unabhängig voneinander an einzelnen Teilen des Systems arbeiten konnten. Die Herausforderung bestand darin, die Vielzahl der Services orchestrieren und überwachen zu können.

2020er Jahre: Künstliche Intelligenz und Automatisierung

Was:

- Die Komplexität der Software zeigt sich zunehmend in der Integration von Künstlicher Intelligenz (KI), maschinellem Lernen (ML) und Automatisierung. Software wird heute "intelligenter" und kann Entscheidungen auf Basis von Daten treffen.
- Anwendungen wie Chatbots, automatische Bilderkennung und Sprachverarbeitung sind allgegenwärtig.
- Edge-Computing ermöglicht die Verarbeitung von Daten am Rand des Netzwerks, was besonders für IoT-Geräte wichtig ist.

Wie:

- KI-Frameworks wie TensorFlow und PyTorch sowie cloudbasierte ML-Plattformen erleichtern die Entwicklung und Bereitstellung von Modellen.
- MLOps, eine Kombination aus DevOps und Machine Learning, wird eingeführt, um ML-Modelle kontinuierlich zu trainieren und zu aktualisieren.
- Sicherheitskonzepte wie Zero-Trust und automatisierte Sicherheitsscans sind notwendiger, da sich Angriffsflächen durch vernetzte und KI-gesteuerte Systeme

vergrößert haben.

Beispiel:

- Sprachassistenten wie Alexa oder Google Assistant kombinieren KI, NLP und Cloud-Computing, um Sprachbefehle zu verstehen und darauf zu reagieren. Die Herausforderung liegt hier in der Komplexität der Datenverarbeitung und dem Datenschutz, da sensible Informationen verarbeitet werden.

Fazit

Die Softwareentwicklung hat sich über die Jahrzehnte erheblich weiterentwickelt:

- **Was:** Die Art der Anwendungen wurde immer komplexer, von Einzelplatzanwendungen auf PCs bis hin zu KI-getriebenen, global vernetzten Anwendungen mit Echtzeitverarbeitung.
- **Wie:** Es wurden neue Paradigmen, Architekturen und Tools eingeführt, um diese zunehmende Komplexität handhaben zu können. Die Methoden zur Entwicklung, Bereitstellung und Wartung von Software wurden immer spezialisierter, um mit der Geschwindigkeit und dem Umfang moderner Anwendungen Schritt zu halten.

Conways_law.de

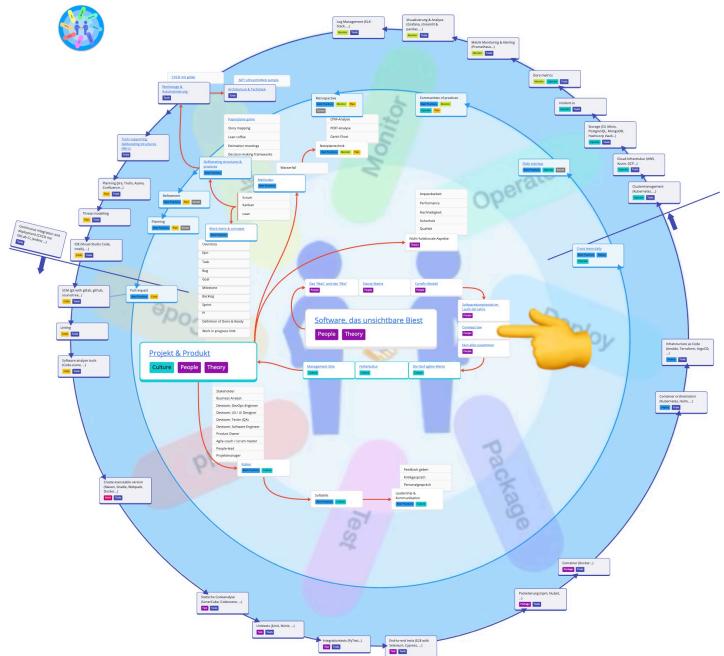
[Das unsichtbare Biest](#) / [Das "Was" und das "Wie"](#) / [Stacey Matrix](#) / [Cynefin-Modell](#) / [Softwarekomplexität seit 1980](#)

Conways law



[Conway's Law - Wikipedia](#)
[Conway's Law - Martin Fowler](#)
[Was ist das Gesetz von](#)

Conway? - t2informatik
 Conway's Law einfach erklärt (deutsch) - Digitale Neuordnung
 Conway's Law (Conway's Gesetz): Definition, Erkenntnisse & Tipps - PURE Consultant
 Conway's Law: Die Auswirkungen auf die Organisationsstruktur und Softwareentwicklung - Digitales Institut
 Das Gesetz von Conway - elmar-dott.com
 Gesetz von Conway | Definition, Nutzen, Weiterentwicklung - projektmagazin
 Conway's Law – Organisation und Systeme perfekt aufeinander abstimmen - Palladio Consulting
 What is Conway's Law? - Smartpedia - t2informatik



"Any organization that designs a system will inevitably produce a design whose structure is a copy of the organization's communication structure."
Melvin E. Conway



Quelle: <https://www.facebook.com/flexbasedigital> (9.11.2024)

Conways Gesetz besagt, dass die Struktur von Softwareprodukten oft die Kommunikationsstrukturen der Teams widerspiegeln, die sie entwickeln. Formuliert wurde es in den 1960er Jahren vom Informatiker Melvin Conway und kann zusammengefasst werden als:

"Organisationen, die Systeme entwerfen, sind gezwungen, Systeme zu entwerfen, die Abbilder ihrer Kommunikationsstrukturen sind."

Auswirkungen auf Softwareprodukte:

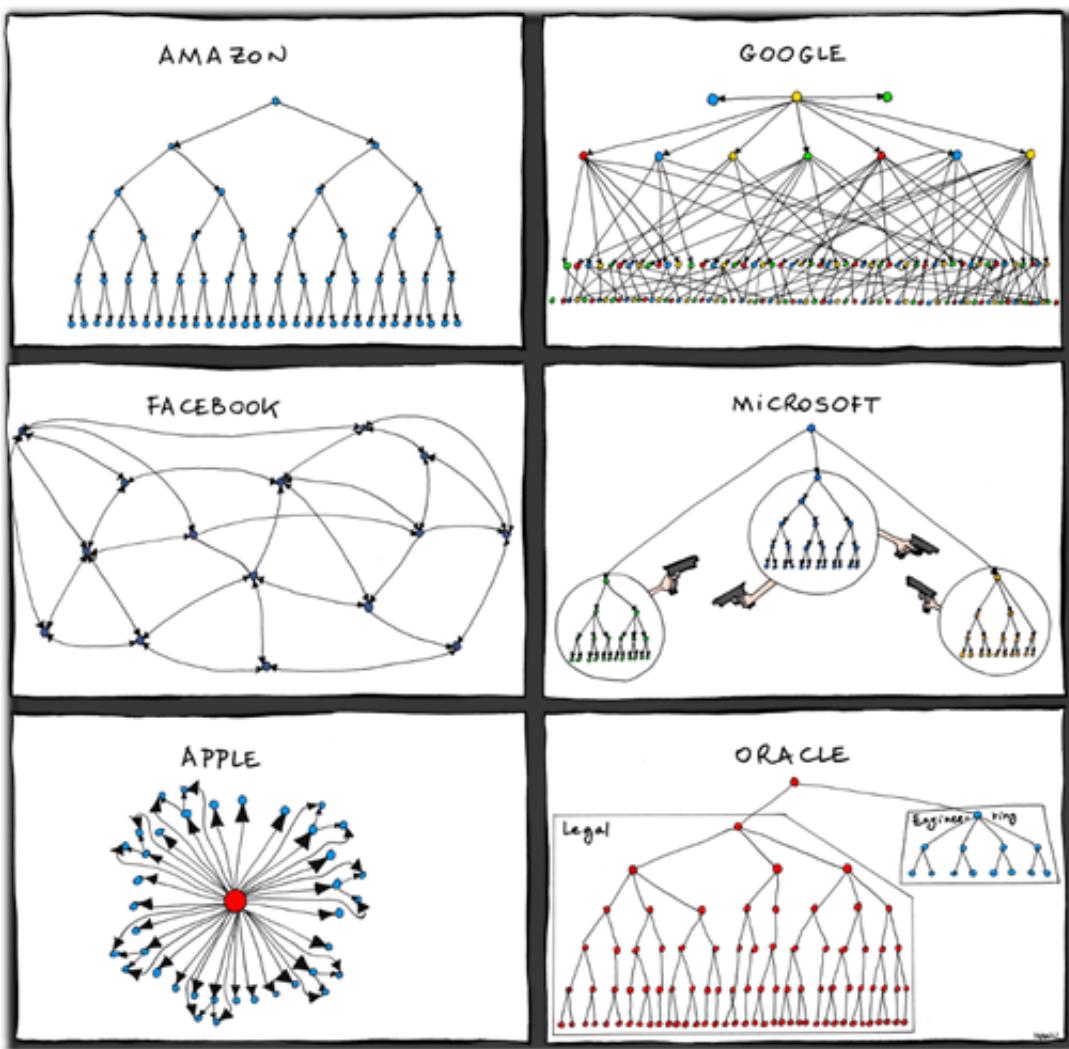
- **Architekturelle Abhangigkeiten:** Isolierte Module und Silos, die schwerfallig und komplex sein konnen.
- **Wartbarkeit und technische Schuld:** Schwierige Anpassungen und Risiko technischer Schuld durch enge Abhangigkeiten.
- **Innovation und Geschwindigkeit:** Starre Strukturen hemmen Innovation; agile Kommunikation fordert Flexibilitat.
- **Microservices und DevOps:** Unterstutzen das Gesetz durch autonom agierende, funktionsbergreifende Teams.
- **Nutzererfahrung und Produktqualitat:** Schlechte Abstimmung zwischen Teams kann zu inkonsistenter Nutzererfahrung fuhren.

Strategien zur positiven Nutzung:

- **Cross-funktionale Teams:** Teams mit allen notigen Kompetenzen minimieren Abhangigkeiten.
- **Optimierte Kommunikation:** Regelmiger Austausch und Meeting-Formate wie "Cross Team Daily".
- **Produktorientierte Teamstrukturen:** Teams entsprechend der Softwaremodule organisieren.

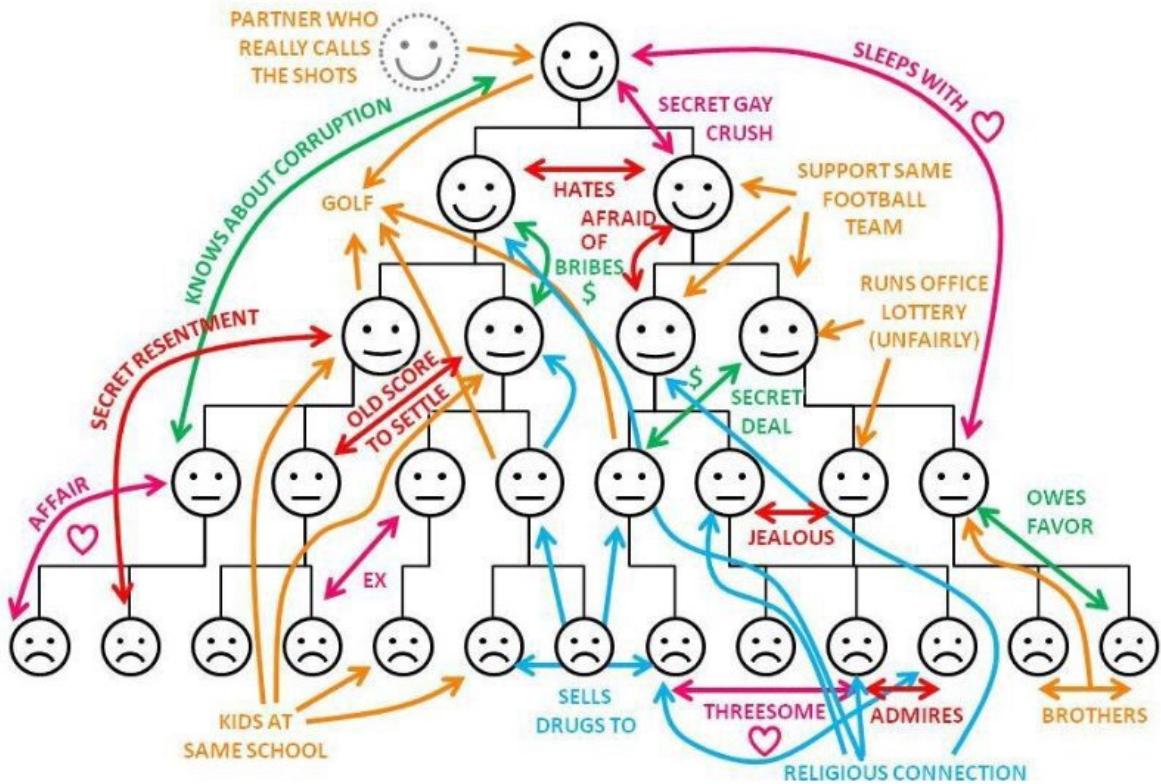
Conways Gesetz in Bildern

Beispiele (mit Augenzwinkern)



Quelle: <https://www.kassenzone.de/conways-law/> (9.11.2024)

Realität versus Planung (mit Augenzwinkern)



Quelle: <https://www.dcme.nu/conways-law/> (9.11.2024)

Das bedeutet, dass sich in Softwarearchitekturen oft dieselben Hierarchien, Schnittstellen und Abgrenzungen finden lassen, die in den Entwicklungsabteilungen existieren. Eine Firma mit stark separierten Abteilungen (z. B. Frontend- und Backend-Teams) neigt also dazu, auch ihre Software in deutlich abgegrenzte Frontend- und Backend-Komponenten aufzuteilen.

Auswirkungen von Conways Gesetz auf Softwareprodukte

1. Architekturelle Abhängigkeiten:

- Wenn Teams stark voneinander getrennt arbeiten und nur wenig miteinander kommunizieren, entstehen oft „Silos“ in der Software. Diese Silos sind Module oder Services, die kaum Schnittstellen zu anderen Teilen des Systems haben und eher isoliert arbeiten.
- Ein aufgeteiltes Team kann dazu führen, dass Systeme schwerfällig sind und die Schnittstellen zwischen Modulen komplex und starr. Dies wird in monolithischen Architekturen oft sichtbar.

2. Wartbarkeit und technische Schuld:

- Systeme, die stark nach den Teamstrukturen aufgebaut sind, haben oft komplizierte Abhängigkeiten. Es kann sein, dass Änderungen oder Erweiterungen der Software erschwert werden, weil es schwierig ist, das Gesamtbild zu durchschauen.
- Technische Schuld kann entstehen, wenn über Jahre hinweg

Softwarestrukturen lediglich den Kommunikationsstrukturen folgen, statt modular und agil auf neue Anforderungen reagieren zu können.

3. Innovation und Geschwindigkeit:

- Kommunikationsstruktur hat einen Einfluss auf die Innovationsfähigkeit eines Teams. In einem stark fragmentierten Unternehmen, in dem sich Teams selten austauschen, kommt es zu weniger Innovation. Die Modularität wird oft nicht genutzt, und die Geschwindigkeit, in der ein Team auf Feedback reagieren kann, sinkt.
- Ist die Kommunikation agil und offen, sind auch die Produkte oft flexibler und modular aufgebaut, sodass neue Ideen und Features schneller umgesetzt werden können.

4. Microservices und DevOps:

- Conways Gesetz hat in der Microservices- und DevOps-Ära an Bedeutung gewonnen, da diese Ansätze selbstorganisierende, funktionsübergreifende Teams fördern. Jedes Team kann Verantwortung für einen bestimmten Service übernehmen, was zu einer modulareren und wartbaren Architektur führt.
- Die unabhängigen Microservices spiegeln dabei die Kommunikationsstruktur der Teams wider und sind in sich geschlossene Einheiten, die autonom entwickelt und skaliert werden können.

5. Einfluss auf Nutzererfahrung und Produktqualität:

- Ein schlecht abgestimmtes Team-Setup kann die Nutzererfahrung beeinflussen. Wenn z. B. das Frontend-Team nicht eng mit dem Backend-Team kommuniziert, können unterschiedliche Interpretationen der Anwendungslogik oder der Benutzerführung entstehen, was die Qualität und Konsistenz beeinträchtigen kann.

Praktische Ansätze, um Conways Gesetz positiv zu nutzen

- **Cross-funktionale Teams aufbauen:** Teams sollten alle notwendigen Kompetenzen abdecken, um eigenverantwortlich Features oder Module zu entwickeln. Damit wird die Softwarearchitektur modularer, und die Abhängigkeiten zwischen Teams und Modulen werden minimiert.
- **Kommunikationswege optimieren:** Regelmäßiger Austausch, z. B. in Form von „Cross Team Dailys“ (wie in deinem Fall geplant), kann helfen, einheitliche Ziele zu verfolgen und Blocker zu identifizieren, die die Produktentwicklung behindern.
- **Teamstrukturen nach dem Produktdesign aufbauen:** Statt die Teamstrukturen willkürlich zu gestalten, kann eine Ausrichtung an den geplanten Softwaremodulen sinnvoll sein. So wird sichergestellt, dass die Architektur der Software stabil und konsistent bleibt, während sie sich den Benutzeranforderungen anpasst.

Conways Gesetz zu berücksichtigen, bedeutet also, die Teamstrukturen bewusst so zu gestalten, dass sie die gewünschte Produktarchitektur und Qualität begünstigen.

Conways law spielerisch kennenlernen

Ein spielerisches Programm könnte Teams und Softwaremodule simulieren, um Conways Gesetz praktisch erfahrbar zu machen. Dabei könnte eine interaktive Simulation helfen, die zeigt, wie verschiedene Teamstrukturen die resultierende Softwarearchitektur beeinflussen. Am Ende könnte man den Zusammenhang visualisieren und durch einen Code-Anwendungsfall verdeutlichen.

Hier ein Beispiel:

```
In [15]: import networkx as nx
import matplotlib.pyplot as plt

# Definiere die Teams und die ihnen zugeordneten Module
teams = {
    "Team A": ["Module 1", "Module 2"],
    "Team B": ["Module 3"],
    "Team C": ["Module 4", "Module 5"]
}

# Definiere die Kommunikationsstruktur zwischen den Teams
communication = {
    "Team A": ["Team B"],
    "Team B": ["Team C"],
    "Team C": []
}

# Definiere die Abhängigkeiten zwischen Modulen
module_dependencies = {
    "Module 1": ["Module 2", "Module 3"],
    "Module 2": ["Module 3"],
    "Module 3": ["Module 5"],
    "Module 4": [],
    "Module 5": []
}

# Visualisierung der Teamkommunikation und Modulabhängigkeiten
def visualize_conways_law(teams, communication, module_dependencies):
    G = nx.DiGraph()

    # Füge Teams als separate Knoten hinzu
    for team in teams:
        G.add_node(team, color="lightcoral") # Teamknoten

    # Füge Module und Zuordnung zu den Teams hinzu
    for team, modules in teams.items():
        for module in modules:
            G.add_node(module, color="skyblue") # Moduleknoten
            G.add_edge(team, module, style="dotted", color="lightcoral")
```

```

# Füge Abhängigkeiten zwischen Modulen hinzu (solid lines)
solid_edges = []
for module, dependencies in module_dependencies.items():
    for dependency in dependencies:
        solid_edges.append((dependency, module))

# Füge Kommunikationslinien zwischen Teams hinzu (dashed lines)
dashed_edges = []
for team, neighbors in communication.items():
    for neighbor in neighbors:
        dashed_edges.append((team, neighbor))

# Visualisierung der Struktur
pos = nx.spring_layout(G)

# Zeichne die Teams als separate Knoten
team_nodes = [node for node in G.nodes if node in teams]
module_nodes = [node for node in G.nodes if node not in teams]

nx.draw_networkx_nodes(G, pos, nodelist=team_nodes, node_size=2500,
nx.draw_networkx_nodes(G, pos, nodelist=module_nodes, node_size=2000,

# Zeichne die Labels für Teams und Module
nx.draw_networkx_labels(G, pos, font_size=10, font_weight="bold")

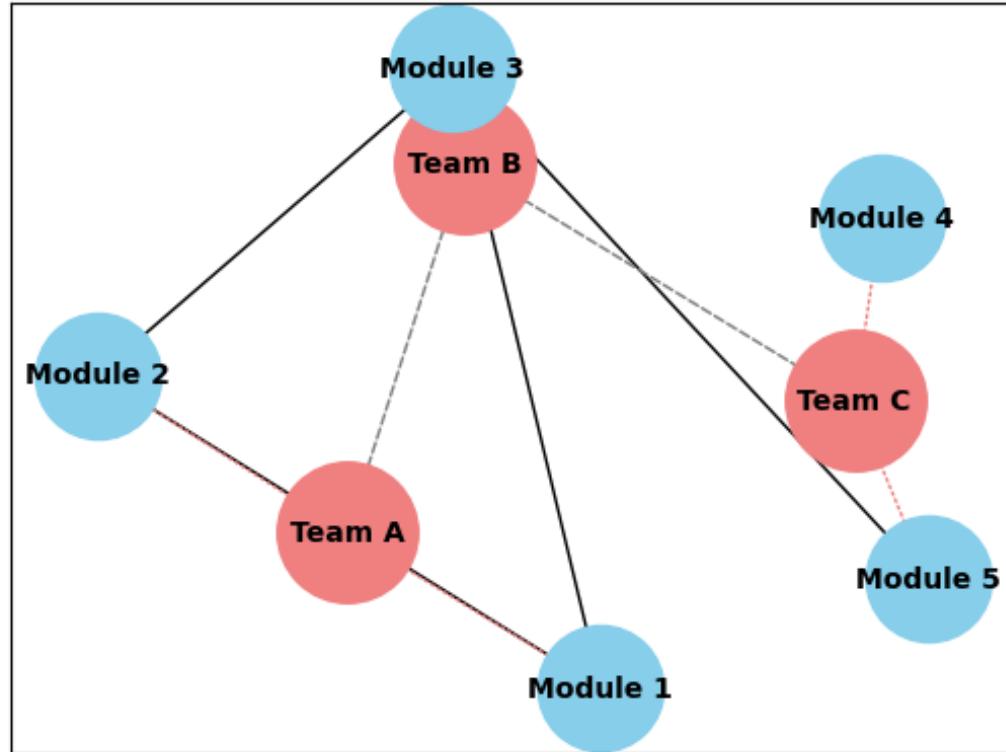
# Zeichne die verschiedenen Kantentypen (Team-Kommunikation und Modul-
nx.draw_networkx_edges(G, pos, edgelist=solid_edges, style="solid",
nx.draw_networkx_edges(G, pos, edgelist=dashed_edges, style="dashed",

# Zeichne die Verbindungen zwischen Teams und ihren Modulen (dotted l
dotted_edges = [(team, module) for team in teams for module in teams[
nx.draw_networkx_edges(G, pos, edgelist=dotted_edges, style="dotted",

plt.title("Conway's Law: Teams, Team Communication, and Module Depend
plt.show()

visualize_conways_law(teams, communication, module_dependencies)

```



Erklärung der Visualisierung

- **Teams (lightcoral):** Teams sind als separate Knoten dargestellt.
- **Module (skyblue):** Module sind ebenfalls als separate Knoten sichtbar.
- **Team-Kommunikation (gestrichelte, graue Linien):** Verbindungen zwischen Teams zeigen, welche Teams miteinander kommunizieren.
- **Modulabhängigkeiten (durchgezogene, schwarze Linien):** Zeigt Abhängigkeiten zwischen Modulen an, die unterschiedliche Teams betreffen können.
- **Team-Modul-Zuordnung (gepunktete, hellrote Linien):** Verbindet Teams mit ihren Modulen, um die Teamstruktur visuell darzustellen.

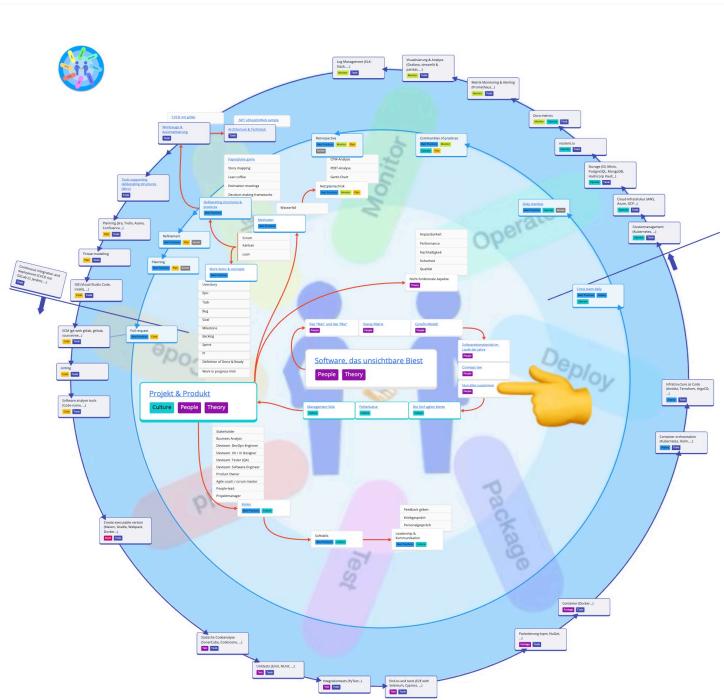
Conclusion.de

[Das unsichtbare Biest](#) / [Das "Was"](#) und das "Wie" / [Stacey Matrix](#) / [Cynefin-Modell](#) / [Softwarekomplexität seit 1980](#) / [Conways law](#)

Zusammenfassung Theorie



Die Stacey-Matrix einfach erklärt – Projekte leicht gemacht
 Cynefin Modell und Stacey Matrix – Doppler Management
 Conway's Law - Wikipedia
 Stacey Matrix: Definition und Bedeutung im Projektmanagement - Qvest



Kontinuierliche Bereitstellung von Mehrwert für Kunden und wirtschaftlichem Nutzen in einer dynamischen Welt nenne ich **Software Product Mastering** in einem

- **Ganzheitlichen Ansatz:** Kulturelle, praktische und technische Integration; vereint DevOps und agile Methoden.

Es leitet sich u.a. ab aus:

- **Stacey Matrix:** Unterstützung in komplexen und chaotischen Umgebungen durch agile und anpassungsfähige Strukturen.
- **Cynefin-Modell:** Navigation zwischen komplizierten und komplexen Domänen; fördert kontinuierliches Experimentieren und Lernen.
- **Conway's Law:** Anpassung der Teamstrukturen für direkte Kommunikation und kundenorientierte Software-Entwicklung.

Gesamtheitliches Ziel: Strukturelle Flexibilität und dynamische Anpassungsfähigkeit, um nachhaltigen Mehrwert zu schaffen.

Es stellt dabei eine Erweiterung dessen, was man gemeinhin unter DevOps-Bewegung versteht, dar. Da DevOps gerne mit Automatisierung verwechselt wird, bzw. darauf reduziert wird, wähle ich hier ganz ausdrücklich einen anderen Begriff.

Die Fähigkeit, Software-Produkte in einer dynamischen Welt kontinuierlich mit Mehrwert für den Kunden und wirtschaftlichem Nutzen anzubieten nenne ich "Software Product Mastering". Dieser ganzheitliche Ansatz integriert kulturelle, praktische und technische Ebenen, inspiriert von DevOps und agilen Methoden, um agile und anpassungsfähige Strukturen zu schaffen.

In Bezug auf die **Stacey Matrix** erkennt Software Product Mastering, dass Software-Entwicklung selten im einfachen oder komplizierten Bereich stattfindet. Durch die Kombination von agilen und DevOps-Methoden wird es möglich, effektiv in komplexen und chaotischen Umgebungen zu agieren, in denen sich Anforderungen und Technologien häufig ändern.

Das **Cynefin-Modell** bietet ebenfalls eine nützliche Perspektive: Software Product Mastering nutzt agile Ansätze, um dynamisch zwischen den Domänen des Komplizierten und Komplexen zu navigieren und schafft Strukturen, die die Erkennung und Anpassung an neue Bedingungen fördern. So kann in einer komplexen Umgebung durch kontinuierliches Experimentieren und Lernen (etwa mittels DevOps und Continuous Delivery) ein optimaler Weg zu nachhaltigem Mehrwert gefunden werden.

Conway's Law zeigt, dass Software-Systeme die Kommunikationsstrukturen der Organisation widerspiegeln, die sie erstellt haben. Ein zentraler Aspekt des Software Product Masterings ist es daher, agile und DevOps-Methoden zu verwenden, um Team- und Organisationsstrukturen anzupassen. Ziel ist, eine Struktur zu schaffen, die enge, direkte Kommunikation und Zusammenarbeit zwischen den Teams ermöglicht und die Entwicklung von Software-Systemen fördert, die anpassungsfähig und kundenorientiert sind.

In Summe ist Software Product Mastering ein umfassender Ansatz, der flexibel auf die Komplexität und Dynamik der modernen Software-Entwicklung reagiert, strukturelle Flexibilität schafft und die kontinuierliche Bereitstellung von Mehrwert für den Kunden unterstützt.

Zusammenwirken der Stacey Matrix und des Cynefin Modells

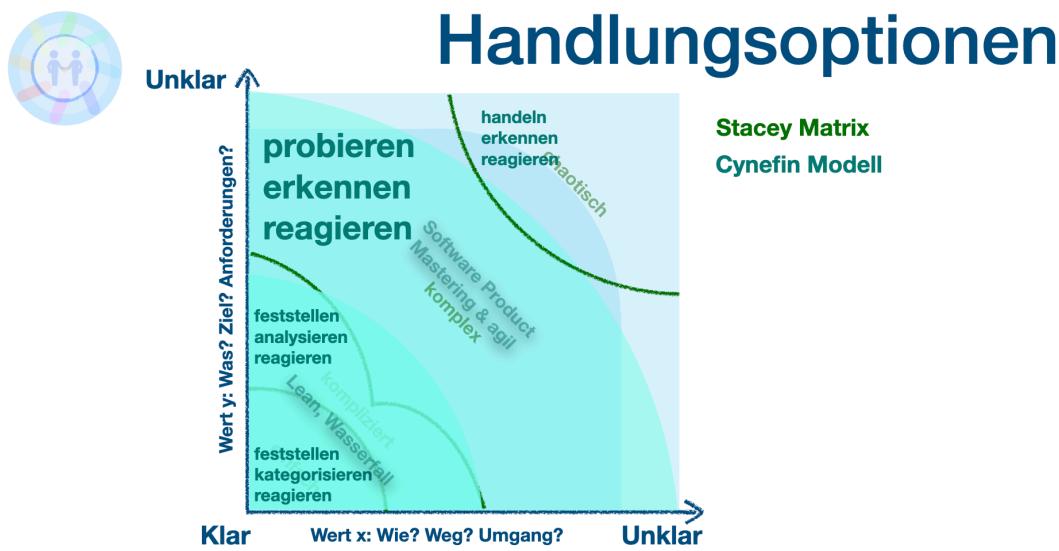


- **Ziel:** Kombination der Modelle hilft, das „Was“ (Ziel/Anforderung) vom „Wie“

(Vorgehen/Methode) zu trennen und Softwareprojekte handhabbarer zu machen.

1. Zusammenwirken der Modelle

- **Stacey Matrix (Was):**
 - Analysiert die Klarheit des Ziels und der Anforderungen.
 - Zeigt, ob Anforderungen stabil oder unsicher sind.
 - **Cynefin Modell (Wie):**
 - Definiert das passende Vorgehen je nach Komplexitätsgrad.
 - Gibt Orientierung für strukturiertes, iteratives oder experimentelles Arbeiten.
-



2. Trennung von „Was“ und „Wie“

- **Was (Ziel):** Die Stacey Matrix bestimmt, wie klar die Anforderungen sind; dynamische Projekte brauchen Flexibilität.
- **Wie (Vorgehen):** Das Cynefin Modell passt Methoden an die Komplexität an; einfache Projekte nutzen bewährte Methoden, komplexe Projekte setzen auf Experimente.

3. Anwendung auf Softwareentwicklung

- **Klarheit im „Was“:** Ein stabiles Ziel ermöglicht Effizienz; bei dynamischen Anforderungen hilft Agilität.
- **Flexibilität im „Wie“:** Cynefin passt das Vorgehen dynamisch an; komplexe Projekte profitieren von iterativen Ansätzen.
- **Wichtig:** Die Trennung von Ziel und Methode macht Softwareprojekte greifbarer und erleichtert die Reaktion auf Veränderungen.

Beide Modelle bieten verschiedene Perspektiven auf komplexe Systeme, aber sie ergänzen sich hervorragend. Wenn wir die beiden Modelle kombinieren, können wir

sowohl die Ziele und Anforderungen eines Projekts (das "Was") als auch das passende Vorgehen und die Methoden (das "Wie") besser verstehen. Das Zusammenspiel dieser beiden Perspektiven hilft, das "unsichtbare Biest" Softwareentwicklung strukturierter und kontrollierter zu handhaben.

1. Die Stacey Matrix: Das "Was" klären

Die Stacey Matrix ist besonders hilfreich, um die Anforderungen und Erwartungen in einem Projekt zu präzisieren und herauszufinden, wie stabil oder variabel sie sind. Dabei unterscheidet sie den Sicherheitsgrad über das Ziel:

- **Klare Anforderungen:** Wenn das Ziel eindeutig und gut verständlich ist, bedeutet das, dass wir eine stabile Vorstellung davon haben, was am Ende erreicht werden soll. Hier können bewährte Methoden angewandt werden, um die Anforderungen umzusetzen.
- **Unsichere Anforderungen:** Wenn die Ziele jedoch unscharf oder stark veränderlich sind, befinden wir uns in einem Bereich, der mehr Flexibilität und Offenheit für Änderungen erfordert. Hier braucht es eine explorative, iterative Vorgehensweise.

Diese Analyse der Anforderungen oder des „Was“ ist wichtig, um festzustellen, wie präzise das Zielbild des Projekts ist. Projekte, deren Anforderungen oft wechseln oder unklar definiert sind, können nicht einfach mit strikten Plänen und fixen Methoden umgesetzt werden. Die Stacey Matrix schafft also Klarheit darüber, ob ein Projekt stabil und gut planbar oder dynamisch und flexibler angegangen werden muss.

2. Das Cynefin Modell: Das "Wie" anpassen

Das Cynefin Modell legt den Fokus stärker auf die Herangehensweise und bietet Orientierung für das Vorgehen, also das „Wie“. Cynefin unterscheidet zwischen verschiedenen Problemtypen und zeigt, wie sich Ursache-Wirkungs-Zusammenhänge in verschiedenen Situationen darstellen:

- **Einfach und kompliziert:** In diesen Domänen sind die Zusammenhänge klar oder zumindest analysierbar. Hier können Teams etablierte oder bewährte Methoden anwenden, und das Vorgehen kann stark strukturiert und vorhersehbar sein.
- **Komplex und chaotisch:** Hier gibt es entweder keine offensichtlichen Zusammenhänge oder sie sind schwer zu erkennen. In komplexen und chaotischen Projekten müssen wir iterativ vorgehen und experimentieren, da sich die Lösungen nur durch Versuch und Irrtum erschließen lassen.

Die Cynefin-Domänen helfen dabei, das Vorgehen der Situation anzupassen. Gerade in komplexen Softwareprojekten, in denen viel Unsicherheit herrscht, ermöglicht das Cynefin Modell, flexibel und responsiv zu arbeiten. Im Gegensatz zur Stacey Matrix,

die mehr auf die Zieldefinition fokussiert ist, beleuchtet Cynefin, **wie** man sich durch das Projekt bewegt und welche Methoden sich in unterschiedlichen Komplexitätsgraden als effektiv erweisen.

Trennung von "Was" und "Wie" für ein effektives Vorgehen

Eine klare Trennung von Ziel (Was) und Methode (Wie) ist entscheidend für ein erfolgreiches Projektmanagement, insbesondere in der Softwareentwicklung, wo oft Unsicherheit und Komplexität herrschen. Wenn man diese Aspekte getrennt betrachtet, können Teams klarer arbeiten, weil jeder Bereich spezifisch auf seine Anforderungen eingeht und nicht mit einem anderen vermischt wird.

1. Das "Was" (Ziel) definieren

Die Stacey Matrix hilft dabei, das Ziel eines Projekts – das "Was" – besser zu verstehen und einzuschätzen. Wenn das Ziel klar ist und die Anforderungen stabil bleiben, kann man auf bewährte Vorgehensweisen setzen und effizienter arbeiten. Ein klares Ziel bedeutet, dass die Anforderungen und Erwartungen gut dokumentiert und allgemein anerkannt sind.

Bei dynamischen oder unsicheren Anforderungen, die häufig in innovativen Softwareprojekten auftreten, erlaubt die Stacey Matrix hingegen, die Unsicherheit in der Zieldefinition sichtbar zu machen. Projekte, die eher vage Ziele haben oder deren Anforderungen oft wechseln, sollten daher keine strikt linearen Planungen verfolgen. Stattdessen eignen sich hier agile Methoden, die Flexibilität und Anpassungsfähigkeit ermöglichen.

2. Das "Wie" (Vorgehen) festlegen

Parallel zur Definition des Ziels hilft das Cynefin Modell, das richtige Vorgehen – das "Wie" – zu bestimmen. Dies bedeutet, dass das Team basierend auf der Komplexität der Aufgaben und der Umgebung entscheidet, wie das Projekt durchgeführt wird. In Situationen, die vorhersehbar oder klar sind, können Routineprozesse und Best Practices angewendet werden. Dies ist besonders hilfreich, wenn das Projektziel klar und gut umrissen ist.

In komplexen und chaotischen Situationen empfiehlt das Cynefin Modell jedoch einen iterativen und experimentellen Ansatz. Anstatt sich auf eine starre Planung zu verlassen, arbeitet das Team hier mit Experimenten und kontinuierlichem Feedback, um das Projekt voranzutreiben. Diese Flexibilität ist entscheidend, um mit Unsicherheit und Komplexität umzugehen, die typischerweise bei der Entwicklung komplexer Software auftreten.

Anwendung auf Software als "unsichtbares Biest"

Software wird oft als "unsichtbares Biest" betrachtet, weil sie schwer fassbar und komplex ist. Die Anforderungen an Software können schnell wechseln, und ihre inneren Strukturen und Logiken bleiben oft verborgen. Durch die Kombination der Stacey Matrix und des Cynefin Modells kann man Softwareprojekte handhabbarer und strukturierter gestalten, indem man die Trennung des "Was" und des "Wie" nutzt.

1. Klarheit im "Was" schaffen

Wenn man die Stacey Matrix anwendet, kann man besser erkennen, wie stabil oder unklar das Ziel des Projekts ist. Diese Klarheit hilft allen Beteiligten, ein gemeinsames Zielbild zu entwickeln und die Erwartungen auszurichten. Ein Projekt mit stabilen Anforderungen erlaubt es, Prozesse effizienter zu gestalten und auf bekannte Methoden zu setzen.

In Projekten, die dynamische Anforderungen haben, kann die Stacey Matrix jedoch darauf hinweisen, dass eine agile Vorgehensweise notwendig ist. Agilität ermöglicht eine flexible Anpassung an wechselnde Anforderungen und macht das unsichtbare Biest Software handhabbarer, da sie kontinuierlich überprüft und verfeinert wird.

2. Flexibilität im "Wie" durch Cynefin

Das Cynefin Modell ermöglicht es, die Methoden und das Vorgehen dynamisch anzupassen, je nach Komplexität des Problems. In Bereichen mit hoher Unsicherheit oder Dynamik (z. B. wenn neue Technologien oder unklare Zielvorgaben im Spiel sind) hilft ein experimentelles und iteratives Vorgehen, um sich an die Komplexität des Systems anzupassen.

Ein komplexes Softwareprojekt kann oft nicht vollständig durchgeplant werden, da viele unbekannte Variablen existieren. Das Cynefin Modell bietet hier Orientierung, indem es aufzeigt, dass iteratives Vorgehen und Experimente effektiver sind als starre Pläne. Dadurch wird das "unsichtbare Biest" Software greifbarer, weil das Team flexibler und reaktionsfähiger arbeitet.

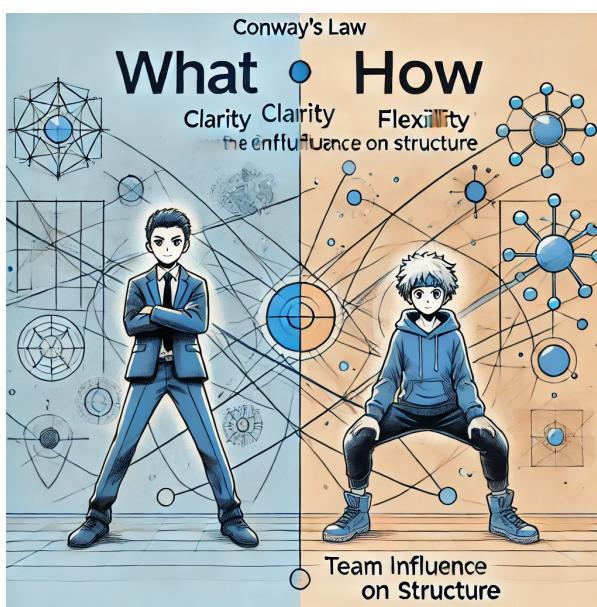
3. Warum die Trennung von "Was" und "Wie" wichtig ist

Indem das Ziel ("Was") von der Methode ("Wie") getrennt wird, schaffen die Modelle ein klareres Verständnis des Projekts und erlauben, dass beide Ebenen spezifisch angesprochen werden. Die Anforderungen (Was) werden unabhängig von der Methodik (Wie) definiert und können sich weiterentwickeln, ohne das Vorgehen zu beeinflussen. Gleichzeitig passt sich das Vorgehen an die Komplexität der jeweiligen Situation an und ist somit flexibler, wenn unvorhergesehene Probleme auftauchen.

Fazit

Das Zusammenwirken der Stacey Matrix und des Cynefin Modells hilft, das "Was" und "Wie" klar voneinander zu trennen und auf die spezifischen Anforderungen eines Projekts abzustimmen. Diese Trennung ist besonders in der Softwareentwicklung hilfreich, wo Anforderungen oft dynamisch und komplex sind. Die Kombination dieser Modelle bietet eine flexible Struktur, um das "unsichtbare Biest" Softwareentwicklung greifbarer zu machen und methodisch vorzugehen. So wird sichergestellt, dass das Zielbild präzise bleibt, während die Umsetzung flexibel auf die Komplexität der Software reagieren kann.

Bedeutung von Conways Gesetz für Klarheit im "Was" und Flexibilität im "Wie"



- **Conways Gesetz:** Systeme spiegeln die Kommunikationsstrukturen ihrer Organisation wider – beeinflusst Design und Umsetzung von Software.

1. Klarheit im "Was" (Stacey Matrix)

- **Gemeinsame Ziele:** Enge, abgestimmte Kommunikation hilft, Anforderungen und Zielbilder klar zu definieren.
- **Vermeidung von Silos:** Integrierte Kommunikation verhindert isolierte Zielvorstellungen und Missverständnisse.
- **Regelmäßiges Feedback:** Schafft Klarheit und erlaubt kontinuierliche Anpassungen an das "Was".

2. Flexibilität im "Wie" (Cynefin Modell)

- **Interdisziplinäre Teams:** Fördern autonome Anpassung und flexibles Vorgehen.
- **Reduzierte Bürokratie:** Offenheit und flache Strukturen erlauben schnelle

- **Reaktionen und iterative Arbeitsweisen.**
- **Experimentierfreudige Kultur:** Erlaubt dynamisches Anpassen des Vorgehens an komplexe Herausforderungen.

3. Passende Kultur als Schlüssel

- **Transparenz und Offenheit:** Unterstützen eine klare Definition des "Was" und Flexibilität im "Wie".
- **Flache Hierarchien:** Fördern Selbstorganisation und schnelle Anpassungen.
- **Agile Prinzipien:** Ermöglichen iteratives Feedback und Anpassungsfähigkeit.

Fazit

Eine passende Kultur, die Kommunikation und Flexibilität fördert, ist entscheidend, um das "Was" klar zu definieren und das "Wie" flexibel anzupassen – dies hilft, komplexe Softwareprojekte erfolgreich zu meistern.

Conways Gesetz besagt, dass Systeme, einschließlich Software, letztlich die Kommunikationsstrukturen der Organisation widerspiegeln, die sie entworfen hat. Dieses Prinzip hat tiefgreifende Auswirkungen auf die Gestaltung von Software und die Organisation von Teams. Wenn Conways Gesetz in Bezug auf die Stacey Matrix und das Cynefin Modell betrachtet wird, wird deutlich, wie wichtig eine geeignete Kultur und Arbeitsweise ist, um sowohl Klarheit im "Was" als auch Flexibilität im "Wie" zu ermöglichen.

Bedeutung von Conways Gesetz für das "Was" und "Wie"

Conways Gesetz impliziert, dass die Art und Weise, wie Teams kommunizieren und strukturiert sind, direkten Einfluss auf das Design, die Architektur und den Erfolg der entwickelten Software hat. Das bedeutet:

- **Wenn Anforderungen unklar sind,** können Kommunikationsprobleme die Unsicherheit verstärken und dazu führen, dass das Zielbild verschwimmt oder Fehlinterpretationen entstehen.
- **Wenn die Methode zu starr ist,** können organisatorische Barrieren die Flexibilität blockieren, was zu ineffizienten Prozessen oder ungeeigneten Lösungen führt.

Um den Anforderungen komplexer Projekte gerecht zu werden, ist es also entscheidend, eine Kultur und Arbeitsweise zu etablieren, die das "Was" und das "Wie" systematisch angeht.

1. Conways Gesetz und Klarheit im "Was" (Stacey Matrix)

Conways Gesetz zeigt, dass die Kommunikationsstrukturen einer Organisation bestimmen, wie Anforderungen verstanden und umgesetzt werden. Eine klare

Kommunikation und gemeinsame Zieldefinition sind daher notwendig, um Missverständnisse oder Inkonsistenzen zu vermeiden:

- **Zusammenarbeit und klare Ziele:** Wenn Teams eng und klar strukturiert zusammenarbeiten, kann das "Was" des Projekts – die Anforderungen und das Zielbild – besser definiert und vermittelt werden.
- **Vermeidung von Silos:** Eine fragmentierte Struktur kann zu unterschiedlichen Zielvorstellungen und Missverständnissen führen, da isolierte Teams oft ihre eigenen Perspektiven und Prioritäten entwickeln. Durch eine integrierte Kultur werden dagegen klare und abgestimmte Anforderungen gefördert.
- **Feedback-Mechanismen:** Regelmäßiges Feedback zwischen Teams schafft mehr Klarheit im "Was", da Anforderungen kontinuierlich überprüft und angepasst werden können.

Ohne eine geeignete Kultur, die abteilungsübergreifende Kommunikation und gemeinsame Ziele fördert, entstehen eher inkohärente Systeme, die den Anforderungen oft nicht gerecht werden.

2. Conways Gesetz und Flexibilität im "Wie" (Cynefin Modell)

Ebenso beeinflusst Conways Gesetz die Flexibilität im "Wie" – also das Vorgehen und die Methodik zur Lösung komplexer Probleme. Starre Kommunikationsstrukturen können Flexibilität und Anpassungsfähigkeit behindern:

- **Förderung interdisziplinärer Teams:** Flexibilität im "Wie" setzt voraus, dass Teams autonom und interdisziplinär arbeiten, um die Methoden anzupassen, wenn neue Herausforderungen auftreten. Diese Struktur fördert ein dynamisches und anpassungsfähiges Vorgehen.
- **Vermeidung von Bürokratie:** Wenn Organisationen ihre Kommunikationsstrukturen zu stark formalisiert haben, werden Entscheidungsprozesse langwierig, und Innovationen werden gehemmt. Eine offene, flexible Kultur ermöglicht es Teams dagegen, sich agil anzupassen und iterativ zu arbeiten.
- **Experimentierfreudigkeit:** In komplexen Projekten ist es wichtig, experimentelle Ansätze zuzulassen. Eine Organisation, die dies durch ihre Struktur und Kultur fördert, erlaubt es Teams, das Vorgehen dynamisch zu ändern und die besten Methoden für unklare Probleme zu finden.

Eine starre Kultur behindert die Reaktion auf Veränderungen. Flexibilität im "Wie" entsteht erst dann, wenn Teams durch offene Kommunikationsstrukturen unterstützt werden.

Die Rolle einer geeigneten Kultur für das "Was" und "Wie"

Eine Kultur, die sowohl Klarheit im "Was" als auch Flexibilität im "Wie" fördert, hilft, Conways Gesetz proaktiv zu nutzen, um Softwareprojekte erfolgreicher zu gestalten.

- **Transparenz und Offenheit:** Wenn Teams auf einer gemeinsamen Informationsbasis arbeiten und regelmäßig kommunizieren, wird das "Was" klarer und stabiler. Gleichzeitig können sie flexibel auf neue Erkenntnisse reagieren und das "Wie" dynamisch anpassen.
- **Kollaboration und flache Hierarchien:** Flache Hierarchien fördern die Selbstorganisation der Teams, was sich positiv auf die Flexibilität auswirkt. Die Kommunikation verläuft effizienter und ermöglicht schnellere Anpassungen an neue Anforderungen oder unerwartete Herausforderungen.
- **Agile Werte und Prinzipien:** Agile Arbeitsweisen fördern genau diese Art von Kultur, indem sie Selbstorganisation und iterative Feedbackzyklen in den Vordergrund stellen. Diese Werte schaffen Raum für eine klare Definition des „Was“ und die nötige Anpassungsfähigkeit im „Wie“.

Fazit: Warum eine passende Kultur entscheidend ist

Conways Gesetz zeigt, dass eine Organisation durch ihre Kommunikationsstruktur Einfluss auf die Art und Weise nimmt, wie ihre Systeme gestaltet und entwickelt werden. Eine klare, offene und flexible Kultur ist daher essentiell, um:

1. **Das "Was" zu klären:** Die Struktur und Kommunikation der Organisation sollten darauf ausgerichtet sein, eine klare Definition der Anforderungen zu fördern und durch regelmäßiges Feedback Anpassungen zu ermöglichen.
2. **Das "Wie" flexibel zu gestalten:** Die Kultur sollte agile und interdisziplinäre Teams unterstützen, die in der Lage sind, ihr Vorgehen dynamisch an die jeweilige Komplexität des Problems anzupassen.

In der Softwareentwicklung hilft eine solche Kultur, das "unsichtbare Biest" in kontrollierbare Schritte zu zerlegen, indem Klarheit im Ziel und Anpassungsfähigkeit im Vorgehen geschaffen werden.

Project_Products.de

[Das unsichtbare Biest](#) / [Das "Was" und das "Wie"](#) / [Stacey Matrix](#) / [Cynefin-Modell](#) / [Softwarekomplexität seit 1980](#) / [Conways law](#) / [Zusammenfassung Theorie](#)

Software-Projekte und Software-Produkte



Difference between Project and Product - GeeksforGeeks

Product Software vs. Project Software: Understanding the Differences - DEV Community

10 Key Differences Between Projects Vs Products - Walter Associates

Software Projects vs Software Products - Blog @romenrg

Product vs Project – Similar Words but Very Different Approaches - Futurice

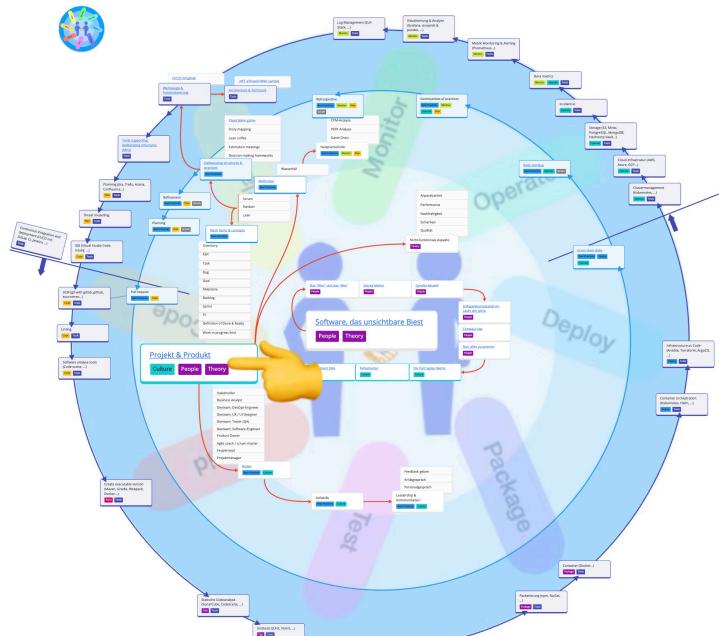
Software Project VS Other types of Project - Exnrt

Projects and Products - What is the Difference and How You Can't Have One Without the Other - Fullstack

Product vs Project: How They Differ, and Why Elon Needs To ... - Avion

The Differences Between IT Projects And Other Types of Projects ... - ScrumTeam.pro

Products vs Projects – On Software Development in the Enterprise - Idea 11



Softwareprojekt

- Zeitlich begrenzt, hat festen Start und Abschluss
- Zielgerichtet auf ein spezifisches Ergebnis (z. B. Feature oder Version)
- Einmalig, oft nicht wiederholbar in dieser Form
- Budget und Ressourcen festgelegt
- Projektmanagement-Methoden (z. B. Agile, Wasserfall)
- Endet mit Lieferung eines abgeschlossenen Ergebnisses

Softwareprodukt

- Langfristiger Lebenszyklus, kontinuierliche Entwicklung
- Ziel: dauerhafter Nutzen und Wert für Kunden
- Regelmäßige Updates und Wartung (Versionierung)
- Kundenzentriert, orientiert an Feedback und Marktbedürfnissen
- Produktmanagement und strategische Planung
- Monetarisierungsmodell, um Einnahmen zu generieren

Hauptunterschiede

- Projekte: zeitlich begrenzt, spezifisches Ziel, endet mit Fertigstellung
- Produkte: langfristig, kontinuierliche Verbesserung, Kunden- und Marktfokus

Softwareprojekte und Softwareprodukte sind zwei eng verwandte, aber unterschiedliche Konzepte in der Softwareentwicklung, die sich in Zielen, zeitlichem Rahmen, Herangehensweise und Resultaten unterscheiden. Hier ist eine detaillierte Erklärung der beiden Begriffe und der Unterschiede:

1. Softwareprojekte

Ein **Softwareprojekt** ist ein zeitlich begrenztes Vorhaben zur Entwicklung oder Anpassung von Software mit einem klaren Ziel und definierten Anforderungen. Es kann Teil eines größeren Programms oder Produkts sein, ist jedoch als eigenständige Initiative organisiert, die in der Regel durch Projektmanagement gesteuert wird.

Merkmale eines Softwareprojekts:

- **Zielorientierung:** Jedes Projekt hat ein spezifisches Ziel, wie z. B. die Entwicklung eines neuen Features, die Einführung einer Anwendung oder die Anpassung einer bestehenden Lösung.
- **Zeitliche Begrenzung:** Projekte haben feste Start- und Endtermine und werden in der Regel in Phasen organisiert (Initiierung, Planung, Durchführung, Abschluss).
- **Einzigartigkeit:** Ein Softwareprojekt wird meist nur einmal in dieser spezifischen Form durchgeführt, da das Ziel, die Anforderungen und die Bedingungen für jedes Projekt unterschiedlich sind.
- **Budget und Ressourcen:** Projekte haben ein zugewiesenes Budget und spezifische Ressourcen (Mitarbeiter, Werkzeuge), die für die Erreichung des Ziels vorgesehen sind.
- **Projektmanagement-Methoden:** Für die Durchführung und Steuerung eines Projekts werden oft Methoden wie Agile, Scrum, Kanban oder klassisches Wasserfall-Projektmanagement verwendet, je nach Komplexität und Anforderungen.
- **Endergebnis:** Das Endergebnis eines Projekts ist eine konkrete, abgeschlossene

Version eines Softwareelements oder einer gesamten Anwendung, das spezifische Anforderungen erfüllt.

2. Softwarereprodukte

Ein **Softwareprodukt** hingegen ist eine Softwareanwendung oder -lösung, die für einen bestimmten Markt oder eine bestimmte Zielgruppe entwickelt wird und über einen längeren Zeitraum gepflegt, erweitert und aktualisiert wird. Der Fokus liegt hier auf der Bereitstellung von langfristigem Wert für Benutzer oder Kunden.

Merkmale eines Softwareprodukts:

- **Langfristiger Lebenszyklus:** Softwarereprodukte werden entwickelt, um langfristig am Markt verfügbar zu sein und kontinuierlich Nutzen für die Anwender zu stiften.
- **Versionierung und Wartung:** Produkte werden regelmäßig aktualisiert, um Fehler zu beheben, neue Funktionen hinzuzufügen und sich an Marktanforderungen anzupassen. Dies erfordert kontinuierliche Wartung, Verbesserung und Support.
- **Wertorientierung:** Ein Softwareprodukt zielt darauf ab, dauerhaften Nutzen zu bieten, indem es für spezifische Probleme der Zielgruppe Lösungen bereitstellt. Der Wert kann wirtschaftlicher oder funktionaler Natur sein.
- **Kundenzentrierung:** Im Mittelpunkt der Produktentwicklung stehen die Bedürfnisse der Kunden und Anwender. Die Entwicklung neuer Funktionen oder Verbesserungen orientiert sich am Feedback der Nutzer und an Marktentwicklungen.
- **Produktmanagement:** Produkte haben in der Regel ein dediziertes Produktmanagement-Team, das die Strategie und den Roadmap-Plan für das Produkt definiert, um den langfristigen Erfolg sicherzustellen.
- **Revenue und Marktausrichtung:** Viele Softwarereprodukte haben ein Geschäftsmodell (z. B. Abonnement, Lizenzierung), um Umsatz zu generieren, und müssen sich gegen Wettbewerber im Markt behaupten.

3. Unterschied zwischen Softwareprojekten und Softwareprodukten

Faktor	Softwareprojekt	Softwareprodukt
Ziel	Erreichen eines spezifischen Ziels, z. B. ein Feature oder eine Version	Bereitstellung langfristigen Nutzens und Wertschöpfung für Kunden
Zeithorizont	Zeitlich begrenzt, mit festem Start und Abschluss	Unbefristet, kontinuierliche Weiterentwicklung

Kundenfokus	Fokus auf das Projektziel, oft unabhängig von langfristigen Kundenbedürfnissen	Kunden- und marktorientiert, regelmäßiges Feedback fließt ein
Management	Wird durch Projektmanagement-Methoden gesteuert	Durch Produktmanagement, das Markt und Nutzerbedürfnisse im Blick hat
Finanzierung	Budget wird projektbezogen geplant und zugewiesen	Einnahmen durch verschiedene Monetarisierungsmodelle
Ergebnis	Ein konkretes und meist abgeschlossenes Produkt (z. B. eine Version)	Ein fortlaufend verfügbares Produkt, das regelmäßig aktualisiert wird

Zusammengefasst:

Der wesentliche Unterschied liegt in der **zeitlichen Begrenzung und Zielsetzung**: Während ein **Softwareprojekt** ein abgeschlossenes Vorhaben mit klarem Ziel ist, dient ein **Softwareprodukt** der fortlaufenden Bereitstellung von Nutzen für den Markt. Ein Projekt endet mit der Erfüllung seines spezifischen Ziels, wohingegen ein Produkt in einem permanenten Prozess der Verbesserung und Anpassung bleibt, solange es auf dem Markt existiert.

Unterschiede spielerisch kennenlernen

Hier ist ein Codebeispiel, das den Unterschied zwischen einem Softwareprojekt und einem Softwareprodukt spielerisch verdeutlicht. Die Idee ist, zwei einfache Aufgaben zu simulieren: eine, die wie ein abgeschlossenes Projekt funktioniert (mit einer festen Zielvorgabe), und eine, die einem Produkt mit kontinuierlicher Verbesserung ähnelt (mit einem Feedbackprozess).

```
In [1]: # Import necessary libraries
import time
import random
import matplotlib.pyplot as plt

# Set random seed for reproducibility
random.seed(42)

# Function simulating a software project with a clear end goal
def software_project(target_feature):
    print("Software Project Simulation: Develop feature")
    completed = False
    progress = 0
    goal = 100 # represents 100% completion

    # Simulate progress towards a fixed goal
    while not completed:
        progress += random.randint(10, 30) # Simulate work chunks
        time.sleep(0.5)
```

```

# Cap progress at the goal and complete the project
if progress >= goal:
    progress = goal
    completed = True
print(f"Project Progress: {progress}%")

print(f"\nProject Completed! Feature '{target_feature}' is fully deve

# Function simulating a software product with continuous updates and improvement
def software_product(initial_version, updates=5):
    print("Software Product Simulation: Continuous Improvement")
    version = initial_version
    user_feedback_scores = []
    version_scores = []

    # Simulate continuous improvement process
    for i in range(updates):
        print(f"\nVersion {version} Released")

        # Get random feedback from users (simulating feedback loop)
        feedback = random.randint(1, 10)
        user_feedback_scores.append(feedback)
        version_scores.append(version)
        print(f"User Feedback: {feedback}/10")

        # Update the version and simulate improvement
        version += 0.1 # Increment version
        print(f"Product updated to version {version:.1f}")
        time.sleep(0.5)

    # Plot feedback over versions
    plt.plot(version_scores, user_feedback_scores, marker='o')
    plt.title("User Feedback on Software Product over Versions")
    plt.xlabel("Product Version")
    plt.ylabel("User Feedback Score")
    plt.show()
    print("Product continues to evolve based on feedback and improvements")

# Run simulations
print("----Simulating Software Project----")
software_product("Login System")

print("\n----Simulating Software Product----")
software_product(1.0)

```

----Simulating Software Project----

Software Project Simulation: Develop feature

Project Progress: 30%

Project Progress: 43%

Project Progress: 53%

Project Progress: 71%

Project Progress: 88%

Project Progress: 100%

Project Completed! Feature 'Login System' is fully developed.

----Simulating Software Product----

Software Product Simulation: Continuous Improvement

Version 1.0 Released

User Feedback: 3/10

Product updated to version 1.1

Version 1.1 Released

User Feedback: 2/10

Product updated to version 1.2

Version 1.2000000000000002 Released

User Feedback: 9/10

Product updated to version 1.3

Version 1.3000000000000003 Released

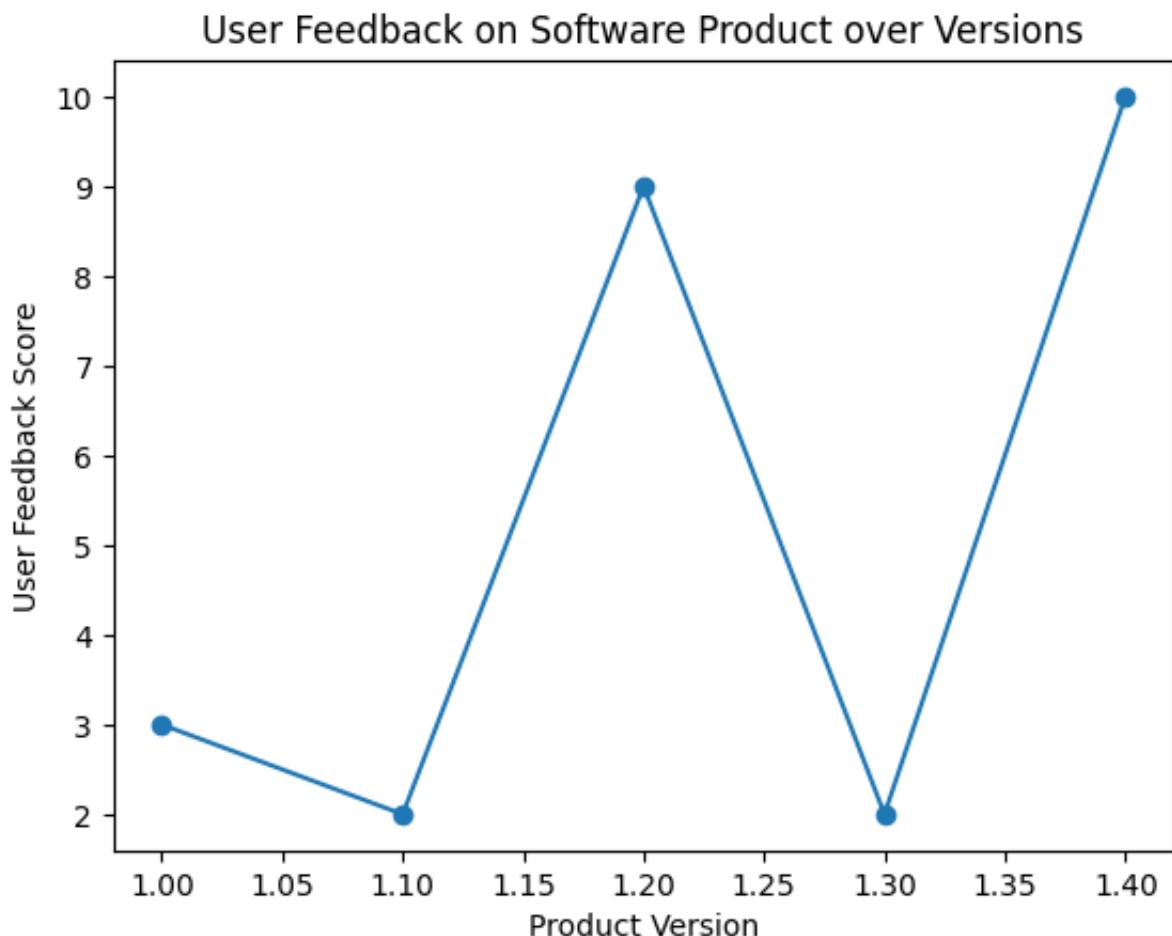
User Feedback: 2/10

Product updated to version 1.4

Version 1.4000000000000004 Released

User Feedback: 10/10

Product updated to version 1.5



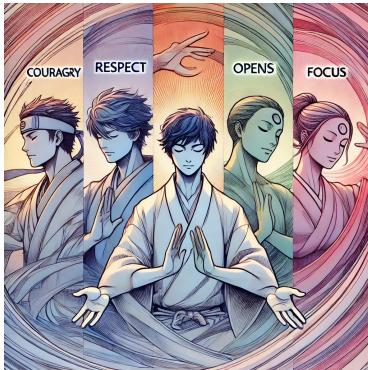
Erläuterung des Codes

1. **Softwareprojekt:** Ein festes Ziel (z. B. "Login System") wird erreicht, indem die Simulation Fortschrittsschritte ausführt, bis 100% erreicht sind. Sobald das Ziel erreicht ist, wird das Projekt als abgeschlossen betrachtet.
2. **Softwareprodukt:** In dieser Simulation wird eine anfängliche Produktversion kontinuierlich aktualisiert, basierend auf zufälligem Benutzerfeedback. Die Version wird iterativ verbessert, und das Feedback wird auf einem Diagramm dargestellt, um den fortlaufenden Charakter eines Produkts zu verdeutlichen.

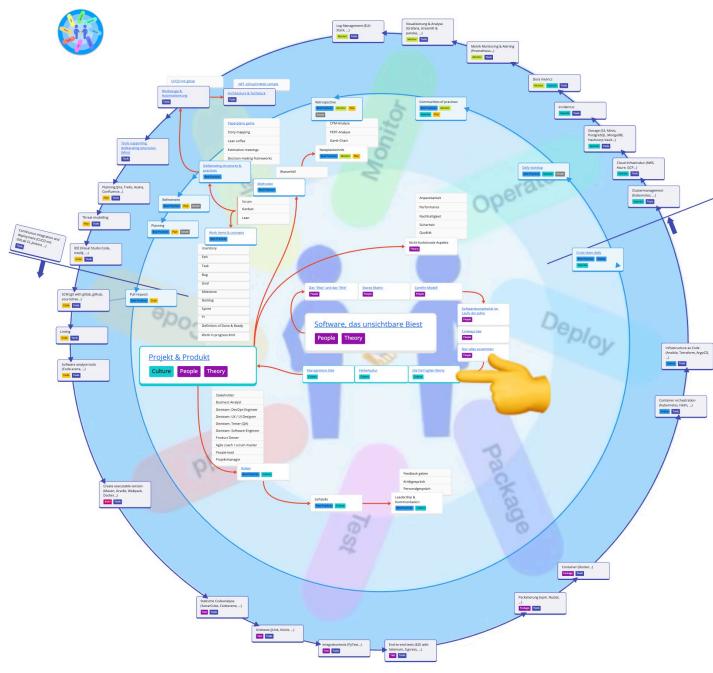
Agile_Values.de

[Das unsichtbare Biest](#) / [Das "Was" und das "Wie"](#) / [Stacey Matrix](#) / [Cynefin-Modell](#) / [Softwarekomplexität seit 1980](#) / [Conways law](#) / [Zusammenfassung Theorie](#)

Die fünf agilen Werte



The Agile Manifesto - Agile Alliance Agile Manifesto and Values - Atlassian



Die fünf agilen Werte sind essenzielle Prinzipien, die dabei helfen, ein vertrauensvolles und produktives Arbeitsumfeld zu schaffen.

- **Mut (Courage):**

- Herausforderungen offen angehen
- Verantwortung übernehmen
- Neue Wege ausprobieren

- **Respekt (Respect):**

- Gleichwertige Behandlung im Team
- Ideen und Beiträge schätzen
- Sicheres Umfeld für Meinungsäußerung schaffen

- **Fokus (Focus):**

- Konzentration auf die wichtigsten Aufgaben
- Ziele erreichen
- Ablenkungen minimieren

- **Transparenz (Openness):**

- Offenheit in der Kommunikation
- Klarer Umgang mit Hindernissen, Risiken und Fortschritten
- Vertrauen und gemeinsame Ausrichtung fördern

- **Verbindlichkeit (Commitment):**

- Verpflichtung zu Zielen und Aufgaben
- Verantwortung übernehmen
- Gemeinsame Zielverfolgung im Team

Beschreibung

Die fünf agilen Werte sind essenzielle Prinzipien, die den Rahmen für agile Methoden wie Scrum oder Kanban bilden. Sie helfen dabei, ein vertrauensvolles und produktives Arbeitsumfeld zu schaffen. Hier eine kurze Beschreibung der einzelnen Werte:

1. **Mut (Courage):** Mut bedeutet, Herausforderungen offen anzugehen und auch schwierige Entscheidungen zu treffen. Im agilen Kontext geht es darum, neue Wege auszuprobieren, Verantwortung zu übernehmen und sich gegenseitig in schwierigen Situationen zu unterstützen.
2. **Respekt (Respect):** Respekt bezieht sich auf den Umgang miteinander. Jeder im Team wird als gleichwertig betrachtet, und alle Meinungen, Ideen und Beiträge werden geschätzt. Es geht darum, Anerkennung für die Leistungen anderer zu zeigen und ein Umfeld zu schaffen, in dem sich jeder sicher fühlt, seine Gedanken zu äußern.
3. **Fokus (Focus):** Fokus bedeutet, sich auf die wichtigsten Aufgaben zu konzentrieren und sicherzustellen, dass die gesetzten Ziele erreicht werden. Im agilen Ansatz hilft Fokus dabei, Ablenkungen zu minimieren und die Aufmerksamkeit auf das zu richten, was dem Kunden den größten Mehrwert bringt.
4. **Transparenz (Openness):** Transparenz fördert Offenheit in der Kommunikation und im Umgang mit Informationen. Alle Teammitglieder und Stakeholder sollten stets über den aktuellen Stand des Projekts informiert sein. Hindernisse, Risiken und Fortschritte werden klar und ehrlich kommuniziert, um Vertrauen und eine gemeinsame Ausrichtung zu fördern.
5. **Verbindlichkeit (Commitment):** Verbindlichkeit bedeutet, dass sich jedes Teammitglied zu den Zielen und Aufgaben verpflichtet, die im Rahmen eines Sprints oder einer Iteration festgelegt wurden. Es geht darum, Verantwortung zu übernehmen und sicherzustellen, dass das Team gemeinsam auf die definierten Ziele hinarbeitet.

Umsetzung der Werte

Mut (Courage)

- **Do's:**
 - Probleme oder Hindernisse offen ansprechen.
 - Risiken eingehen, um innovative Lösungen zu finden.
 - Neue, unkonventionelle Ideen einbringen und ausprobieren.
- **Don'ts:**

- Sich davor scheuen, Fehler zuzugeben oder anzusprechen.
- Konflikten aus dem Weg gehen, anstatt sie zu lösen.
- Veränderungen meiden aus Angst vor dem Scheitern.

Respekt (Respect)

- **Do's:**
 - Aufmerksam zuhören und andere Meinungen ernst nehmen.
 - Teammitglieder unterstützen, wenn sie Hilfe benötigen.
 - Feedback wertschätzend und konstruktiv geben.
- **Don'ts:**
 - Ideen anderer abwerten oder ignorieren.
 - Teammitglieder öffentlich kritisieren oder bloßstellen.
 - Eigenes Ego über die Zusammenarbeit im Team stellen.

Fokus (Focus)

- **Do's:**
 - Prioritäten klar setzen und sich auf die wichtigsten Aufgaben konzentrieren.
 - Ablenkungen minimieren, um Ziele effizient zu erreichen.
 - Klare Ziele und Ergebnisse für jede Iteration/Sprint definieren.
- **Don'ts:**
 - Zu viele parallele Aufgaben übernehmen, die den Fokus verwässern.
 - Häufige Unterbrechungen im Arbeitsprozess zulassen.
 - Zeit auf Aufgaben verschwenden, die keinen Mehrwert bringen.

Transparenz (Openness)

- **Do's:**
 - Den aktuellen Projektstand regelmäßig mit dem Team teilen.
 - Herausforderungen und Risiken frühzeitig ansprechen.
 - Ehrlich über Fortschritte und Probleme kommunizieren.
- **Don'ts:**
 - Informationen zurückhalten, die für das Team relevant sind.
 - Fehler oder Hindernisse verschweigen, um Konflikte zu vermeiden.
 - Wichtige Entscheidungen hinter verschlossenen Türen treffen.

Verbindlichkeit (Commitment)

- **Do's:**
 - Zu den vereinbarten Zielen und Aufgaben stehen.
 - Verantwortung für das eigene Arbeitspaket übernehmen.

- Die Sprint-Ziele gemeinsam im Team erarbeiten und verfolgen.
- **Don'ts:**
 - Aufgaben zusagen und diese dann nicht rechtzeitig abschließen.
 - Versprechen machen, die man nicht halten kann.
 - Das Team im Stich lassen, wenn es auf Unterstützung angewiesen ist.

Diese **Do's** und **Don'ts** helfen dabei, die agilen Werte im Alltag lebendig zu machen und die Zusammenarbeit im Team nachhaltig zu verbessern.

Die fünf Dysfunktionen eines Teams

Die fünf Dysfunktionen eines Teams, wie von Patrick Lencioni beschrieben, lassen sich gut mit den agilen Werten in Beziehung setzen. Diese Dysfunktionen stehen oft im direkten Gegensatz zu den agilen Werten und verdeutlichen, wie wichtig die Umsetzung der Werte für eine effektive Teamarbeit ist.

Hier ist eine Übersicht, wie sich die fünf Dysfunktionen auf die agilen Werte beziehen:

1. Abwesenheit von Vertrauen (Absence of Trust)

- **Bezug zu den agilen Werten:**
 - **Mut:** Ohne Vertrauen trauen sich Teammitglieder nicht, offen über Schwächen oder Fehler zu sprechen.
 - **Transparenz:** Misstrauen führt dazu, dass Informationen zurückgehalten werden, was die Offenheit im Team behindert.
 - **Respekt:** Vertrauen erfordert gegenseitigen Respekt – ohne diesen entsteht ein Umfeld von Skepsis und Misstrauen.
- **Do's:** Teammitglieder ermutigen, sich gegenseitig zu vertrauen, Fehler offen zuzugeben, und ein unterstützendes Umfeld schaffen.
- **Don'ts:** Misstrauen zulassen, durch fehlende Offenheit oder mangelnde Anerkennung von Beiträgen.

2. Angst vor Konflikten (Fear of Conflict)

- **Bezug zu den agilen Werten:**
 - **Mut:** Es erfordert Mut, Konflikte direkt anzusprechen und konstruktiv zu lösen.
 - **Respekt:** Respekt hilft dabei, Konflikte auf gesunde Weise zu führen, ohne die Beziehung zu den Teammitgliedern zu belasten.
 - **Verbindlichkeit:** In Teams, die Konflikte vermeiden, wird es schwer, klare Verbindlichkeiten zu schaffen, weil Probleme nie offen diskutiert werden.
- **Do's:** Offenheit für Diskussionen und konstruktive Auseinandersetzungen fördern, um Probleme zu lösen.

- **Don'ts:** Konflikten aus dem Weg gehen, schwierige Gespräche vermeiden oder Probleme unter den Teppich kehren.

3. Mangel an Engagement (Lack of Commitment)

- **Bezug zu den agilen Werten:**
 - **Fokus:** Ohne klare Verbindlichkeiten und Engagement fehlt der Fokus auf gemeinsame Ziele.
 - **Verbindlichkeit:** Teams ohne Engagement setzen sich nicht wirklich für vereinbarte Ziele und Aufgaben ein.
 - **Transparenz:** Fehlt das Engagement, ist es auch schwer, klare, offene Kommunikation über Ziele und Verantwortlichkeiten sicherzustellen.
- **Do's:** Sicherstellen, dass jedes Teammitglied sich zu den gemeinsamen Zielen und Aufgaben bekennt.
- **Don'ts:** Entscheidungen treffen, ohne das Team einzubinden, oder unklare Ziele setzen, die das Engagement beeinträchtigen.

4. Scheu vor Verantwortung (Avoidance of Accountability)

- **Bezug zu den agilen Werten:**
 - **Verbindlichkeit:** Teams ohne Verantwortungsbewusstsein übernehmen keine Verantwortung für ihre Ziele.
 - **Fokus:** Wenn niemand zur Rechenschaft gezogen wird, wird es schwer, den Fokus auf wichtige Aufgaben aufrechtzuerhalten.
 - **Mut:** Verantwortung zu übernehmen erfordert Mut, besonders bei schwierigen Aufgaben oder Entscheidungen.
- **Do's:** Klare Verantwortlichkeiten definieren und sicherstellen, dass jeder Verantwortung für sein Handeln übernimmt.
- **Don'ts:** Verantwortungslosigkeit tolerieren oder das Team nicht in die Lage versetzen, sich gegenseitig zur Rechenschaft zu ziehen.

5. Ergebnisorientierung (Inattention to Results)

- **Bezug zu den agilen Werten:**
 - **Fokus:** Wenn Teams nicht auf Ergebnisse achten, verlieren sie den Fokus auf das Wesentliche.
 - **Verbindlichkeit:** Ein Team muss sich den Ergebnissen verschreiben und die gemeinsame Zielverfolgung sicherstellen.
 - **Transparenz:** Es ist wichtig, die Ergebnisse und Fortschritte offen zu kommunizieren, um sicherzustellen, dass das Team auf Kurs bleibt.
- **Do's:** Den Fokus auf gemeinsame, messbare Ergebnisse richten und die Fortschritte transparent kommunizieren.

- **Don'ts:** Eigeninteressen über die Teamziele stellen oder Ergebnisse nicht regelmäßig überprüfen.

Zusammenhang zwischen agilen Werten und Dysfunktionen eines Teams:

- **Mut:** Hilft, Vertrauen aufzubauen, Konflikte zu bewältigen und Verantwortung zu übernehmen.
- **Respekt:** Fördert Vertrauen und konstruktive Konfliktlösungen.
- **Fokus:** Steuert das Team in Richtung gemeinsamer Ergebnisse, beugt mangelndem Engagement und Ergebnisorientierung vor.
- **Transparenz:** Schafft Vertrauen und hält das Team auf Kurs, indem Informationen und Ergebnisse klar geteilt werden.
- **Verbindlichkeit:** Sorgt für klare Verantwortlichkeiten und steigert das Engagement für die gemeinsamen Ziele.

Die fünf agilen Werte wirken den Dysfunktionen eines Teams entgegen und stärken die Zusammenarbeit und Leistung in agilen Teams.

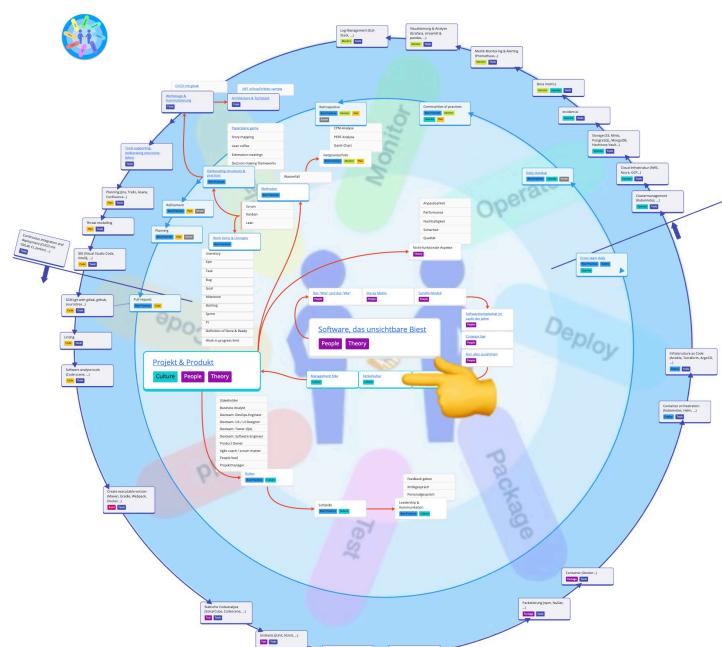
Error_Culture.de

Das unsichtbare Biest / Das "Was" und das "Wie" / Stacey Matrix / Cynefin-Modell / Softwarekomplexität seit 1980 / Conways law / Zusammenfassung Theorie

Fehlerkultur



The Power of Error Culture:
Why Mistakes Lead to Success - HRtalk.de
Positive Error Culture: Why Mistakes Actually Make Sense - OVB
Error Culture in Companies:
How Agility Needs and Promotes It - Sulzer
Positive Error Culture in the Company - Brainbirds Blog
EY: Study on Error Culture



in German Companies
Positive Error Culture: Your Way to More Openness in the Team - Greator
With a Productive Error Culture in the Company to More Innovation - ARTS
Establishing Error Culture in Companies Correctly - Berlitz
How Companies Should Respond to Mistakes - The Global Network for Advanced Management
Transcript: You Messed Up at Work. What Now? - Financial Times

Fehlerkulturen im Allgemeinen:

Verschiedene Arten von Fehlerkulturen wurden beschrieben, darunter:

- **Schuldzuweisungskultur**
- **Vermeidungskultur**
- **Lernkultur**
- **Innovationskultur**
- **Toleranzkultur**
- **Null-Fehler-Kultur**
- **Offene Kultur**

Fehlerkulturen im agilen Kontext:

- Besonders geeignet sind:
 - **Lernkultur:** Unterstützt kontinuierliche Verbesserung und Fehleranalyse.
 - **Innovationskultur:** Erlaubt schnelles Experimentieren und Lernen aus Fehlern.
 - **Offene Kultur:** Fördert Transparenz und offene Kommunikation über Fehler.

Wichtige agile Prinzipien:

- Fokus auf **kontinuierliche Verbesserung, schnelle Anpassung und offene Kommunikation.**

Kontraproduktive Fehlerkulturen:

- **Schuldzuweisungskultur** und **Vermeidungskultur** behindern Agilität durch Angst und fehlende Transparenz.

Beschreibung

Fehlerkultur bezieht sich auf den Umgang einer Organisation oder eines Teams mit

Fehlern, und es gibt unterschiedliche Arten von Fehlerkulturen, die sich vor allem durch den Umgang mit Fehlern und die Reaktion auf diese unterscheiden. Im Folgenden sind einige der bekanntesten Arten von Fehlerkulturen aufgeführt:

1. Schuldzuweisungskultur (Blame Culture):

- In dieser Kultur werden Fehler sofort einer oder mehreren Personen angelastet. Der Fokus liegt auf der Schuldfrage, und es gibt oft Konsequenzen für den Fehlerverursacher. Dies führt häufig zu Angst und einer defensiven Haltung der Mitarbeiter, was Innovation und offene Kommunikation hemmt.

2. Vermeidungskultur (Avoidance Culture):

- Hier werden Fehler vertuscht oder nicht zugegeben, um negative Konsequenzen zu vermeiden. Mitarbeiter versuchen, Fehler vor Vorgesetzten und Kollegen zu verstecken, was zu mangelnder Transparenz und langfristig zu größeren Problemen führen kann.

3. Lernkultur (Learning Culture):

- Diese Fehlerkultur fördert das Lernen aus Fehlern. Fehler werden als Chance zur Verbesserung und Weiterentwicklung gesehen. Der Fokus liegt darauf, die Ursachen des Fehlers zu verstehen und Maßnahmen zu ergreifen, um ähnliche Fehler in der Zukunft zu vermeiden.

4. Innovationskultur (Innovation Culture):

- In dieser Kultur werden Fehler als notwendiger Teil des Innovationsprozesses betrachtet. Es wird ermutigt, Risiken einzugehen und aus Fehlschlägen zu lernen, um neue Ideen und Lösungen zu entwickeln. Solche Kulturen sind oft in Startups oder agilen Teams zu finden.

5. Toleranzkultur (Tolerance Culture):

- In dieser Kultur wird eine gewisse Fehlerhäufigkeit toleriert, solange sie die Arbeitsleistung nicht drastisch beeinträchtigt. Fehler werden akzeptiert, jedoch wird wenig Wert auf eine systematische Fehleranalyse oder das Lernen aus diesen gelegt.

6. Null-Fehler-Kultur (Zero Defect Culture):

- Diese Kultur strebt danach, Fehler zu eliminieren und setzt hohe Standards für Genauigkeit und Präzision. In einer Null-Fehler-Kultur gibt es oft strenge Kontrollen und Qualitätsmanagementsysteme. Dies kann zu hoher Qualität führen, jedoch auch zu Stress und Druck bei den Mitarbeitern.

7. Offene Kultur (Open Culture):

- In dieser Kultur herrscht eine offene Kommunikation über Fehler. Mitarbeiter werden ermutigt, Fehler offen zuzugeben und konstruktive Vorschläge zu deren Behebung zu machen. Der Fokus liegt auf der Teamarbeit und darauf, gemeinsam Lösungen zu finden.

Jede dieser Fehlerkulturen hat Vor- und Nachteile, und der Erfolg hängt oft vom Kontext und den Zielen der Organisation ab. Eine gesunde Fehlerkultur zeichnet sich meist durch Offenheit, Lernbereitschaft und einen konstruktiven Umgang mit Fehlern aus.

Fehlerkultur im agilen Umfeld

Im agilen Kontext eignen sich besonders die **Lernkultur**, die **Innovationskultur** und die **offene Kultur**, da diese Ansätze gut mit den Prinzipien und Werten des agilen Manifests harmonieren. Hier eine genauere Betrachtung, warum diese drei Fehlerkulturen im agilen Umfeld besonders förderlich sind:

1. Lernkultur (Learning Culture):

- **Agile Prinzipien** fördern kontinuierliche Verbesserung und Anpassung. Eine Lernkultur unterstützt diesen Ansatz, da Fehler als Chance zur Verbesserung und Weiterentwicklung betrachtet werden.
- Teams in einem agilen Umfeld reflektieren regelmäßig über ihre Arbeit (z.B. in Retrospektiven) und passen Prozesse an, um aus Fehlern zu lernen und sich stetig zu verbessern.
- Eine Lernkultur ermutigt zur Fehleranalyse, ohne Schuldzuweisungen, und schafft so ein Umfeld, in dem Teams sicher experimentieren und optimieren können.

2. Innovationskultur (Innovation Culture):

- Agilität erfordert oft schnelles Experimentieren und iteratives Vorgehen, um neue Produkte oder Lösungen zu entwickeln. Eine Innovationskultur betrachtet Fehler als natürlichen Bestandteil des Innovationsprozesses.
- Diese Kultur fördert Risikobereitschaft und erlaubt es Teams, neue Ideen auszuprobieren, ohne Angst vor Sanktionen bei Misserfolgen zu haben. Dies ist besonders wichtig in agilen Ansätzen wie Scrum oder Kanban, wo schnelles Feedback und Anpassung entscheidend sind.
- Durch die Bereitschaft, auch aus Fehlschlägen zu lernen, entsteht ein dynamisches Umfeld, das Kreativität und Innovation begünstigt.

3. Offene Kultur (Open Culture):

- Transparenz ist ein zentraler Wert in agilen Teams. Eine offene Fehlerkultur, in der über Fehler gesprochen wird, ohne dass Mitarbeiter Angst vor Schuldzuweisungen haben müssen, unterstützt diesen Transparenzgedanken.
- Agile Teams arbeiten oft eng zusammen und müssen in der Lage sein, offen über Herausforderungen und Fehler zu sprechen, um gemeinsam Lösungen zu finden. Diese offene Kommunikation fördert die Teamdynamik und stärkt das Vertrauen unter den Mitgliedern.
- Die offene Kultur fördert auch eine kontinuierliche Feedbackschleife, die in

agilen Prozessen essenziell ist, um schnell zu reagieren und Fehler zu beheben, bevor sie sich zu größeren Problemen entwickeln.

Warum diese Fehlerkulturen im agilen Kontext besonders geeignet sind:

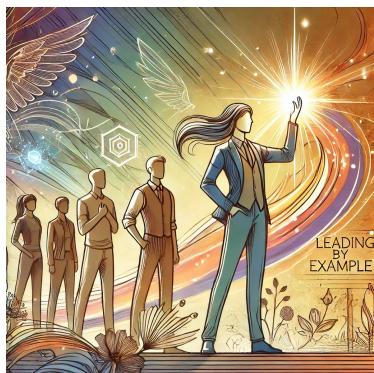
- **Fokus auf kontinuierliche Verbesserung:** Agilität setzt auf kleine, iterative Verbesserungen, die durch Lernen aus Fehlern unterstützt werden.
- **Schnelle Anpassung:** Agiles Arbeiten bedeutet, flexibel auf Veränderungen zu reagieren. Eine Innovations- und Lernkultur erleichtert es, schnell auf Fehler einzugehen und Lösungen zu finden.
- **Offene Kommunikation und Feedback:** Transparenz und Feedback sind Schlüsselaspekte agiler Methoden. Die offene Fehlerkultur erleichtert es, Probleme rechtzeitig zu adressieren und gemeinsam an Verbesserungen zu arbeiten.

Im Gegensatz dazu sind Kulturen wie die **Schuldzuweisungskultur** oder die **Vermeidungskultur** kontraproduktiv für Agilität, da sie Angst vor Fehlern schüren und Innovation sowie Transparenz behindern. Agilität lebt davon, dass Teams sich frei fühlen, Risiken einzugehen, aus Fehlern zu lernen und gemeinsam die bestmöglichen Lösungen zu finden.

Management_styles.de

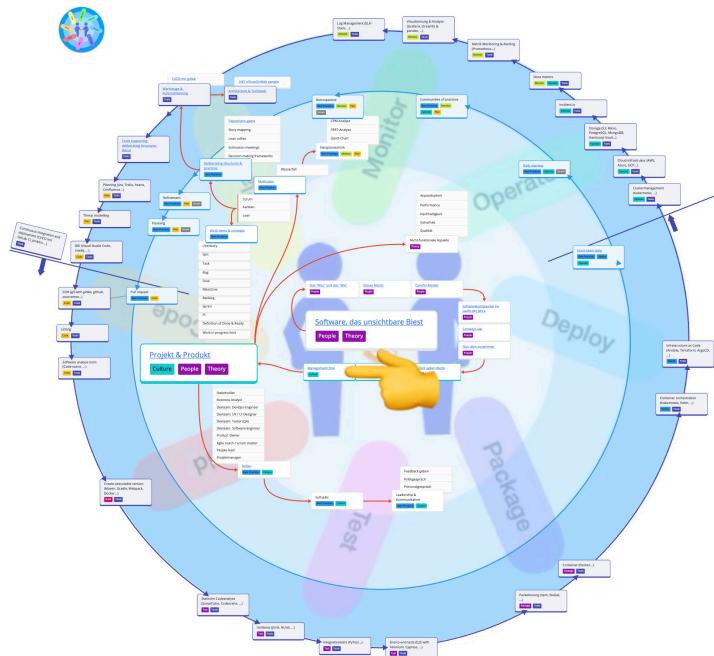
[Das unsichtbare Biest](#) / [Das "Was" und das "Wie"](#) / [Stacey Matrix](#) / [Cynefin-Modell](#) / [Softwarekomplexität seit 1980](#) / [Conways law](#) / [Zusammenfassung Theorie](#)

Management Stile



10 Management Styles Of Effective Leaders – Forbes Advisor

[Führungsstile: Die verschiedenen Modelle und Stile im Vergleich! - Asana](#)
[10 Types of Management Styles For Effective Leadership \[2024\] - Valamis](#)
[9 Different Management Styles and When to Use Them \[2024\] • Asana](#)
[8 Types of Management Styles for Effective Leadership - Indeed](#)
[Der richtige Managementstil zur Unterstützung Ihres Teams \[2024\] • Asana](#)
[Understanding Management Styles: A Comprehensive Overview - Creately](#)
[Effektive Management-Stile für Leader - Smartsheet](#)
[8 Management Styles in Business: Choosing the Best Fit - Coursera](#)
[9 Types of Management Styles and Real-Life Examples - Fellow.app](#)



1. Autoritärer Stil:

- Vorteile: Schnelle Entscheidungen, hilfreich bei unerfahrenen Teams
- Nachteile: Wenig Kreativität und Motivation, hinderlich für Innovation
- Eignung: Selten geeignet, da Softwareentwicklung Flexibilität erfordert

2. Demokratischer Stil:

- Vorteile: Fördert Kreativität und Eigenverantwortung, ideal für agile Methoden
- Nachteile: Langsame Entscheidungen, ungeeignet für schnelle Entscheidungsanforderungen
- Eignung: Sehr geeignet für kollaborative und agile Umgebungen

3. Laissez-faire Stil:

- Vorteile: Hohe Autonomie, fördert kreative Ansätze
- Nachteile: Erfordert hohe Eigenverantwortung, kann chaotisch werden
- Eignung: Geeignet für selbstorganisierte, erfahrene Teams

4. Transformationaler Stil:

- Vorteile: Motivation durch Vision, fördert kontinuierliches Lernen und Innovation
- Nachteile: Erfordert hohe kommunikative Fähigkeiten
- Eignung: Sehr geeignet, fördert Innovationskultur und stetige Verbesserung

5. Situativer Stil:

- Vorteile: Anpassungsfähig je nach Projektphase und Teamdynamik
- Nachteile: Potenzielle Verwirrung bei häufigem Stilwechsel
- Eignung: Gut geeignet, flexibel für verschiedene Projektphasen

6. Servant Leadership:

- Vorteile: Unterstützt agile Ansätze, fördert Vertrauen und Eigenverantwortung
- Nachteile: Kann als Führungsschwäche wahrgenommen werden
- Eignung: Ideal für agile Teams und eine vertrauensvolle Teamkultur

Empfehlung: Demokratischer, transformationaler, situativer und servant leadership Stil sind besonders geeignet, da sie Innovation, Anpassungsfähigkeit und Teamverantwortung in der Softwareentwicklung fördern.

Es gibt eine Vielzahl von Managementstilen, die sich in unterschiedlichen Branchen bewährt haben. Besonders in der Softwareentwicklung spielen einige dieser Stile eine entscheidende Rolle, da sie die Dynamik und Komplexität der Branche adressieren. Hier sind die wichtigsten Managementstile sowie ihre Vor- und Nachteile in der Softwareentwicklung:

1. Autoritärer Managementstil

- **Merkmale:** Der Manager trifft alle Entscheidungen alleine, gibt klare Anweisungen und erwartet, dass das Team diese strikt befolgt. Es herrscht eine klare Hierarchie, und Mitarbeiter haben wenig Entscheidungsfreiheit.
- **Vorteile in der Softwareentwicklung:**
 - Gut für Notfallsituationen, in denen schnelle Entscheidungen erforderlich sind.
 - Geeignet, wenn das Team unerfahren ist und klare Richtlinien benötigt.
- **Nachteile:**
 - Mitarbeiter haben wenig Spielraum für Kreativität und Eigeninitiative.
 - Kann Innovationsfähigkeit und Motivation mindern, was in kreativen Prozessen wie der Softwareentwicklung kontraproduktiv ist.
- **Eignung für Softwareentwicklung:** Dieser Stil eignet sich kaum, da Softwareentwicklung kreative und innovative Ansätze erfordert, die durch strikte Vorgaben und wenig Freiheit eingeschränkt werden könnten. Zudem setzt Softwareentwicklung oft auf flexible und iterative Prozesse, was mit dem autoritären Stil schwer zu vereinbaren ist.

2. Demokratischer oder partizipativer Managementstil

- **Merkmale:** Der Manager bindet das Team in Entscheidungsprozesse ein und fördert die Zusammenarbeit. Feedback und Ideen der Mitarbeiter sind

willkommen und werden berücksichtigt.

- **Vorteile in der Softwareentwicklung:**

- Fördert kreatives Denken und Innovation, da Teammitglieder Ideen einbringen können.
- Steigert die Motivation und das Engagement des Teams, da die Mitarbeiter das Gefühl haben, Teil des Prozesses zu sein.
- Eignet sich für agile Methoden, da Feedback und iterative Verbesserung gefördert werden.

- **Nachteile:**

- Entscheidungen können länger dauern, was den Entwicklungsprozess verlangsamen kann.
- Eignet sich weniger für Situationen, die schnelle und klare Entscheidungen erfordern.

- **Eignung für Softwareentwicklung:** Sehr gut geeignet, vor allem für agile und kollaborative Entwicklungsumgebungen. Agile Methoden wie Scrum und Kanban setzen auf Eigenverantwortung und Mitbestimmung des Teams, was mit diesem Managementstil ideal zusammenpasst.

3. Laissez-faire Managementstil

- **Merkmale:** Der Manager greift kaum in die Arbeit der Mitarbeiter ein und überlässt ihnen weitgehend die Entscheidungsfreiheit. Der Fokus liegt auf Autonomie und Selbstorganisation.

- **Vorteile in der Softwareentwicklung:**

- Hohe Freiheit und Flexibilität ermöglichen kreative Lösungen und Innovationen.
- Besonders geeignet für erfahrene, eigenständige Teams, die ihre Projekte selbst managen können.

- **Nachteile:**

- Ohne klare Führung kann es zu Unsicherheiten und ineffizienten Prozessen kommen.
- Erfordert ein hohes Maß an Disziplin und Eigenverantwortung im Team, was nicht immer gewährleistet ist.

- **Eignung für Softwareentwicklung:** Geeignet für hochqualifizierte und selbstorganisierte Teams, insbesondere in der Produktentwicklung, wo kreative und individuelle Ansätze gefragt sind. In weniger erfahrenen Teams kann dieser Stil jedoch zu Chaos und mangelnder Zielstrebigkeit führen.

4. Transformationaler Managementstil

- **Merkmale:** Der Manager inspiriert und motiviert das Team, große Ziele zu erreichen. Der Fokus liegt auf der Förderung und Weiterentwicklung der Mitarbeiter sowie der Erreichung gemeinsamer Visionen.
- **Vorteile in der Softwareentwicklung:**
 - Motiviert Mitarbeiter durch Visionen und Herausforderungen.
 - Schafft ein Umfeld, in dem Mitarbeiter ihre Fähigkeiten kontinuierlich weiterentwickeln können, was für die Innovationsfähigkeit eines Softwareteams entscheidend ist.
 - Unterstützt die Implementierung von agilen Ansätzen, da Mitarbeiter zur Verbesserung des Produktes inspiriert werden.
- **Nachteile:**
 - Erfordert hohe kommunikative und motivatorische Fähigkeiten des Managers.
 - Kann ohne klare Struktur und Zielsetzung ins Leere laufen.
- **Eignung für Softwareentwicklung:** Sehr gut geeignet, da Softwareentwicklung oft auf Innovation und stetige Verbesserung setzt. Dieser Stil fördert eine Kultur des Lernens und Wachsens, was in dynamischen Projekten von Vorteil ist.

5. Situativer Managementstil

- **Merkmale:** Der Manager passt seinen Führungsstil je nach Situation und Teamdynamik an. Er variiert zwischen autoritären, demokratischen und Laissez-faire-Ansätzen, je nachdem, was die Situation erfordert.
- **Vorteile in der Softwareentwicklung:**
 - Flexibel und anpassungsfähig, was insbesondere in dynamischen Entwicklungsumgebungen wichtig ist.
 - Ermöglicht es, das Team je nach Phase des Projekts anders zu führen – z. B. autoritär in Notfallsituationen und demokratisch in der Konzeptphase.
- **Nachteile:**
 - Kann verwirrend für das Team sein, wenn der Stil zu oft wechselt.
 - Erfordert ein hohes Maß an Beobachtungsgabe und Anpassungsfähigkeit des Managers.
- **Eignung für Softwareentwicklung:** Besonders geeignet, da sich Softwareprojekte häufig in unterschiedlichen Phasen befinden, die unterschiedliche Herangehensweisen erfordern. Ein situativer Führungsstil erlaubt es, auf diese Phasen flexibel zu reagieren und die jeweils beste Methode anzuwenden.

6. Servant Leadership (dienende Führung)

- **Merkmale:** Der Manager sieht sich selbst als Unterstützer des Teams. Der Fokus liegt darauf, Hindernisse zu beseitigen und das Team in seiner Arbeit zu fördern und zu befähigen.
- **Vorteile in der Softwareentwicklung:**
 - Ideal für agile Ansätze, insbesondere Scrum, wo der Manager (z. B. als Scrum Master) dem Team dient und es dabei unterstützt, effizienter zu arbeiten.
 - Fördert eine hohe Motivation und Eigenverantwortung im Team.
 - Schafft eine vertrauensvolle und positive Arbeitsatmosphäre, die für die Zusammenarbeit förderlich ist.
- **Nachteile:**
 - Kann als Führungsschwäche wahrgenommen werden, wenn das Team klare Anweisungen benötigt.
 - Setzt ein hohes Maß an Eigenverantwortung und Reife im Team voraus.
- **Eignung für Softwareentwicklung:** Sehr gut geeignet, besonders für agile Teams, da der Manager als Unterstützer auftritt und das Team zur Selbstorganisation befähigt. Servant Leadership passt gut zur Kultur vieler moderner Softwareunternehmen und fördert Vertrauen und Eigenverantwortung.

Zusammenfassung

In der Softwareentwicklung sind besonders **demokratische, transformative, situative**, und **servant leadership**-Stile empfehlenswert. Diese Stile fördern Innovation, Eigenverantwortung und Anpassungsfähigkeit – alles Eigenschaften, die für die erfolgreiche Entwicklung von Softwareprodukten entscheidend sind. Agile Methoden wie Scrum oder Kanban bauen auf diesen Werten auf und schaffen damit die Basis für eine dynamische und produktive Arbeitsweise, die in der modernen Softwareentwicklung oft gefordert ist.

Leading by example

Allgemein: Manager zeigt durch eigenes Verhalten und Arbeitseinstellung, was er vom Team erwartet. Wichtiger Einflussfaktor in der Softwareentwicklung.

1. **Demokratischer Stil:** Manager inspiriert durch aktives Mitwirken und Verantwortung; stärkt Teamgeist und Eigeninitiative.
2. **Transformationaler Stil:** Manager verkörpert die Werte (Lernen, Transparenz), inspiriert das Team zur Selbstentwicklung und Engagement.
3. **Laissez-faire Stil:** Manager schafft durch Vorbild Orientierung in selbstorganisierten Teams.
4. **Situativer Stil:** Manager passt Verhalten an Projektphasen an; zeigt nötige Einsatzbereitschaft und Planung.

5. **Servant Leadership:** Manager dient dem Team, fördert Vertrauen und Eigenverantwortung durch Unterstützung und Vorbildlichkeit.

Bedeutung für Softwareentwicklung: Fördert Teamkultur, Produktivität und Vertrauen; besonders passend in agilen Umgebungen (Scrum, Kanban) für Eigenverantwortung und stetige Verbesserung.

"Leading by example" ist eine Führungstechnik, die in vielen Managementstilen eine wichtige Rolle spielt und besonders in der Softwareentwicklung von Bedeutung ist. Bei diesem Ansatz zeigt der Manager durch sein eigenes Verhalten und seine eigene Arbeitshaltung, was er von seinem Team erwartet. Dies geht über bloße Anweisungen hinaus und basiert darauf, durch konkrete Aktionen Vorbild für das Team zu sein.

In der Perspektive der verschiedenen Managementstile hat "leading by example" folgende Wirkungen:

1. Demokratischer Managementstil und "Leading by Example"

Im demokratischen Stil hat "leading by example" eine enorme Wirkung, da der Manager nicht nur die Ideen und Vorschläge des Teams berücksichtigt, sondern auch durch eigenes Engagement und Einsatzbereitschaft inspiriert. Wenn ein Manager selbst aktiv am Projekt teilnimmt und bereit ist, Verantwortung zu übernehmen, wird das Team eher geneigt sein, ebenfalls Eigeninitiative zu zeigen und gemeinsam zu agieren. Die gemeinsame Arbeit fördert den Teamgeist und motiviert, da der Manager als Teil des Teams agiert und Verantwortung teilt.

2. Transformationaler Managementstil und "Leading by Example"

Im transformationalen Stil ist "leading by example" oft ein zentrales Element. Transformative Führungskräfte inspirieren ihre Teams durch eine klare Vision und hohe persönliche Integrität, indem sie selbst mit gutem Beispiel vorangehen. Wenn ein Manager hohe Erwartungen an das Team hat, ist es wichtig, dass er selbst die Werte und Prinzipien verkörpert, die er sich wünscht, wie etwa die Bereitschaft zu lernen, Transparenz, Offenheit für Feedback und Innovation. Dies motiviert die Mitarbeiter und schafft ein Umfeld, in dem Teammitglieder das Gefühl haben, eine bedeutungsvolle Arbeit zu leisten, und selbst zu Führungspersönlichkeiten wachsen können.

3. Laissez-faire Managementstil und "Leading by Example"

In einem laissez-faire Umfeld könnte "leading by example" ein unterstützendes Element sein, das Orientierung bietet. Da das Team relativ eigenständig arbeitet, kann der Manager durch sein eigenes Verhalten eine Orientierung bieten, ohne dabei

aktiv Anweisungen zu geben. Dies ist besonders nützlich in selbstorganisierten Teams, wo der Manager durch seine eigene Arbeitsweise und Standards eine Art impliziten Rahmen schafft. Der Manager kann zeigen, wie Herausforderungen angegangen und Probleme gelöst werden, was das Team inspiriert und bei der Selbstorganisation unterstützt.

4. Situativer Managementstil und "Leading by Example"

Im situativen Managementstil passt sich der Manager dem aktuellen Bedarf des Teams an. Hier ist "leading by example" besonders wertvoll, da der Manager je nach Situation durch sein Verhalten zeigen kann, was in der jeweiligen Phase des Projekts notwendig ist. In einer hektischen Phase kann der Manager, indem er z. B. Überstunden macht oder schwierige Aufgaben übernimmt, signalisiert, dass Einsatz gefragt ist. In ruhigeren Phasen kann er wiederum zeigen, wie Reflexion und sorgfältige Planung wichtig sind. "Leading by example" ist im situativen Stil ein flexibles Werkzeug, das die Anpassung des Teams erleichtert und Orientierung bietet.

5. Servant Leadership und "Leading by Example"

Im Servant Leadership ist "leading by example" von zentraler Bedeutung, da der Manager hier als dienender Unterstützer auftritt. Er zeigt durch seine eigene Hilfsbereitschaft, sein Engagement für das Team und seine Bereitschaft, Hindernisse zu beseitigen, was es heißt, eine dienende Rolle zu übernehmen. Indem der Manager selbst für andere da ist und die Bedürfnisse des Teams in den Vordergrund stellt, zeigt er, dass Erfolg durch Zusammenarbeit und gegenseitige Unterstützung erreicht wird. Dies fördert eine Kultur des Vertrauens und der Eigenverantwortung, da der Manager seine Rolle als Vorbild ernst nimmt und echte Integrität zeigt.

Bedeutung von "Leading by Example" in der Softwareentwicklung

Softwareentwicklung ist ein Bereich, in dem Vorbildfunktion eine enorme Wirkung auf die Teamkultur und das Produktivitätsniveau haben kann. Da das Team oft in komplexen, kreativen und anspruchsvollen Projekten arbeitet, ist eine Führungskraft, die durch ihr eigenes Verhalten zeigt, was sie erwartet, von unschätzbarem Wert. Dies kann bedeuten, dass der Manager selbst auf Detailgenauigkeit achtet, iteratives und datengetriebenes Arbeiten unterstützt, agile Werte lebt und kontinuierlich nach Verbesserungen strebt.

Besonders in agilen Entwicklungsumgebungen – wie Scrum oder Kanban – passt "leading by example" perfekt, da die Prinzipien Eigenverantwortung, ständige Verbesserung und offene Kommunikation fördern. Der Manager zeigt, dass er nicht nur darüber spricht, sondern die Werte auch aktiv lebt. Durch "leading by example" können Führungskräfte das Vertrauen und den Respekt ihrer Teams gewinnen und so

eine Kultur des Engagements und der Produktqualität schaffen.

Schlussfolgerung: relevante Pfeiler

Zusammenfassung der Managementpfeiler und "Leading by Example" in der Softwareentwicklung:

1. Autonomie:

- Manager zeigt Eigenverantwortung; fördert Selbstständigkeit und Kreativität.
- Passend für demokratischen und servant-leadership Stil.

2. Mastery:

- Fokus auf ständige Verbesserung und Weiterbildung.
- Transformationaler Stil inspiriert durch persönliches Wachstum.

3. Purpose:

- Manager lebt den Sinn der Arbeit; schafft Verbundenheit zum großen Ganzen.
- Transformationaler und situativer Stil unterstützen das Bewusstsein für Projektziele.

4. Baby Stepping:

- Große Ziele in kleine Schritte unterteilt; kontinuierlicher Fortschritt vorgelebt.
- Situativer Stil flexibel für Anpassungen geeignet.

5. Data Driven:

- Entscheidungen basieren auf Daten; Manager zeigt datengetriebenes Handeln.
- Unterstützt durch servant-leadership.

6. Fehlerkultur:

- Manager zeigt offenen Umgang mit Fehlern, fördert Lern- und Risikobereitschaft.
- Transformationaler und servant-leadership Stil stärken das Vertrauen.

Zusammenfassung: Die Pfeiler fördern ein motiviertes und produktives Arbeitsumfeld. Durch "leading by example" und geeignete Managementstile – servant-leadership, transformationaler und situativer Stil – wird eine positive, engagierte Teamkultur aufgebaut.

Die Managementpfeiler – Autonomie, Mastery, Purpose, Baby Stepping, Data Driven, und Fehlerkultur – lassen sich sehr gut mit "leading by example" und den geeigneten Managementstilen für die Softwareentwicklung kombinieren. Hier ist eine detaillierte Betrachtung, wie diese Prinzipien miteinander in Einklang gebracht werden können:

1. Autonomie

Durch Autonomie ermöglicht du deinem Team, eigenverantwortlich Entscheidungen zu treffen und kreativ zu arbeiten. "Leading by example" spielt hier eine zentrale Rolle: Indem du selbst zeigst, wie du dir vertrauensvoll Freiräume nimmst und Verantwortung übernimmst, vermittelst du deinem Team, dass Eigenverantwortung gewünscht und wertgeschätzt wird. Der demokratische und der servant-leadership Stil eignen sich besonders gut für diesen Ansatz. Wenn der Manager durch Vorbildwirkung zeigt, dass Autonomie gefördert und unterstützt wird, entsteht eine Kultur, in der Mitarbeiter sich befähigt fühlen, selbstständig und proaktiv zu handeln.

2. Mastery (Beherrschung und persönliche Weiterentwicklung)

Mastery beschreibt den Drang, kontinuierlich besser zu werden und neue Fähigkeiten zu entwickeln. Indem du als Manager selbst lernst, dich weiterentwickelst und offen für Feedback bist, zeigst du deinem Team, dass Weiterbildung und Exzellenz zentral sind. Der transformationale Stil passt hervorragend dazu, da du durch Vision und Vorbildfunktion das Team inspirierst, kontinuierlich zu lernen und zu wachsen. Ein Manager, der sich für eigene Weiterbildung einsetzt und die Zeit für Selbstverbesserung einplant, zeigt dem Team, dass dieses Streben nach Meisterschaft wertgeschätzt und unterstützt wird.

3. Purpose (Sinn und Ziel)

Purpose oder das Gefühl eines größeren Sinns ist oft ein starker Motivator, besonders in kreativen und intellektuellen Berufen wie der Softwareentwicklung. Ein Manager, der den Zweck der Arbeit authentisch lebt und klar kommuniziert, schafft ein Vorbild für das Team. Durch den transformationalen und situativen Stil kannst du Purpose in die tägliche Arbeit integrieren und sicherstellen, dass das Team versteht, wie ihre Arbeit zu einem größeren Ganzen beiträgt. Wenn der Manager selbst den Sinn hinter jedem kleinen Schritt erkennt und betont, verstärkt dies das Verständnis und Engagement des Teams für die Projektziele.

4. Baby Stepping (Schrittweises Vorgehen)

Baby Stepping bedeutet, dass große Ziele durch kleine, erreichbare Schritte verfolgt werden. Durch "leading by example" kannst du deinem Team zeigen, dass schrittweises Arbeiten effizient und zielführend ist, indem du selbst kleine, kontinuierliche Fortschritte machst und sie hervorhebst. Der situative Managementstil, in dem der Manager sich flexibel an die Fortschritte des Teams anpasst, ist hier besonders hilfreich. Wenn der Manager realistische, kleine Etappen vorlebt und feiert, entwickelt sich eine Arbeitskultur, in der kontinuierliche Verbesserung im Vordergrund steht und das Team motiviert bleibt.

5. Data Driven (Datenbasierte Entscheidungen)

Ein datenbasierter Ansatz ermöglicht es, Entscheidungen faktenbasiert zu treffen und die besten Lösungen zu finden. Wenn der Manager selbst Daten analysiert und aufgrund dieser Erkenntnisse handelt, etabliert er "data driven" als Standard im Team. Der servant-leadership Stil eignet sich hierfür gut, da der Manager das Team darin unterstützt, Zugang zu relevanten Daten zu haben und diese effektiv zu nutzen. Durch das Vorbild des Managers, der die Entscheidungen anhand von Daten trifft und dies transparent kommuniziert, wird datengetriebenes Handeln zur Norm.

6. Fehlerkultur

Eine konstruktive Fehlerkultur, die darauf basiert, dass Menschen im besten Wissen und Gewissen gehandelt haben, schafft Vertrauen und Offenheit. Indem du selbst Fehler offen zugibst und den Lernprozess daraus transparent machst, lebst du eine vorbildliche Fehlerkultur vor. Der servant-leadership und der transformationale Stil unterstützen dies besonders, da diese Stile Vertrauen und Lernbereitschaft fördern. Ein Manager, der Fehler als Lernchance sieht und aktiv zeigt, wie man konstruktiv mit Fehlern umgeht, baut eine Kultur auf, in der das Team keine Angst hat, Risiken einzugehen und kreativ zu arbeiten.

Zusammenfassung

Mit den Pfeilern "Autonomie," "Mastery," "Purpose," "Baby Stepping," "Data Driven," und "Fehlerkultur" kannst du ein hoch engagiertes, innovatives und produktives Umfeld in der Softwareentwicklung schaffen. Durch "leading by example" und die Anwendung passender Managementstile – insbesondere servant-leadership, transformationale Führung und situatives Management – kannst du diese Werte nachhaltig im Team verankern. Dein Ansatz zeigt, dass du deine Mitarbeiter als kompetente und motivierte Menschen wahrnimmst, die am Ende des Tages einen guten Job machen möchten, was die Grundlage für eine starke und positive Teamkultur ist.

Management Stile spielerisch kennenlernen

Hier ist ein Vorschlag, das die verschiedenen Managementstile und ihre Auswirkungen spielerisch simuliert. Die Simulation nutzt Python, um verschiedene Managementstile anzuwenden und zu zeigen, wie diese sich auf das Teamverhalten (z. B. Produktivität, Motivation und Kreativität) auswirken könnten.

In [1]: `# Management Styles Simulation Notebook`

```
import numpy as np
import matplotlib.pyplot as plt
import random
```

```

# Setting up initial parameters
team_size = 10 # Number of team members
iterations = 10 # Number of project iterations
management_styles = ["Authoritarian", "Democratic", "Laissez-faire", "Tra

# Initial team state (randomized initial motivation, productivity, creativity)
team_state = {
    "motivation": np.random.uniform(0.4, 0.6, team_size),
    "productivity": np.random.uniform(0.4, 0.6, team_size),
    "creativity": np.random.uniform(0.4, 0.6, team_size),
}

# Define how each management style affects the team state
def apply_management_style(style, team_state):
    if style == "Authoritarian":
        team_state["motivation"] -= 0.1
        team_state["productivity"] += 0.1
        team_state["creativity"] -= 0.1
    elif style == "Democratic":
        team_state["motivation"] += 0.1
        team_state["productivity"] += 0.05
        team_state["creativity"] += 0.1
    elif style == "Laissez-faire":
        team_state["motivation"] += 0.05
        team_state["productivity"] -= 0.05
        team_state["creativity"] += 0.15
    elif style == "Transformational":
        team_state["motivation"] += 0.15
        team_state["productivity"] += 0.1
        team_state["creativity"] += 0.1
    elif style == "Situational":
        team_state["motivation"] += np.random.choice([-0.1, 0.1])
        team_state["productivity"] += np.random.choice([-0.1, 0.1])
        team_state["creativity"] += np.random.choice([-0.1, 0.1])
    elif style == "Servant":
        team_state["motivation"] += 0.2
        team_state["productivity"] += 0.05
        team_state["creativity"] += 0.1

    # Ensure values stay between 0 and 1
    for key in team_state:
        team_state[key] = np.clip(team_state[key], 0, 1)
    return team_state

# Simulation function
def run_simulation(style):
    motivation_history, productivity_history, creativity_history = [], []

    for _ in range(iterations):
        apply_management_style(style, team_state)
        motivation_history.append(np.mean(team_state["motivation"]))
        productivity_history.append(np.mean(team_state["productivity"]))
        creativity_history.append(np.mean(team_state["creativity"]))

    return motivation_history, productivity_history, creativity_history

# Plotting function

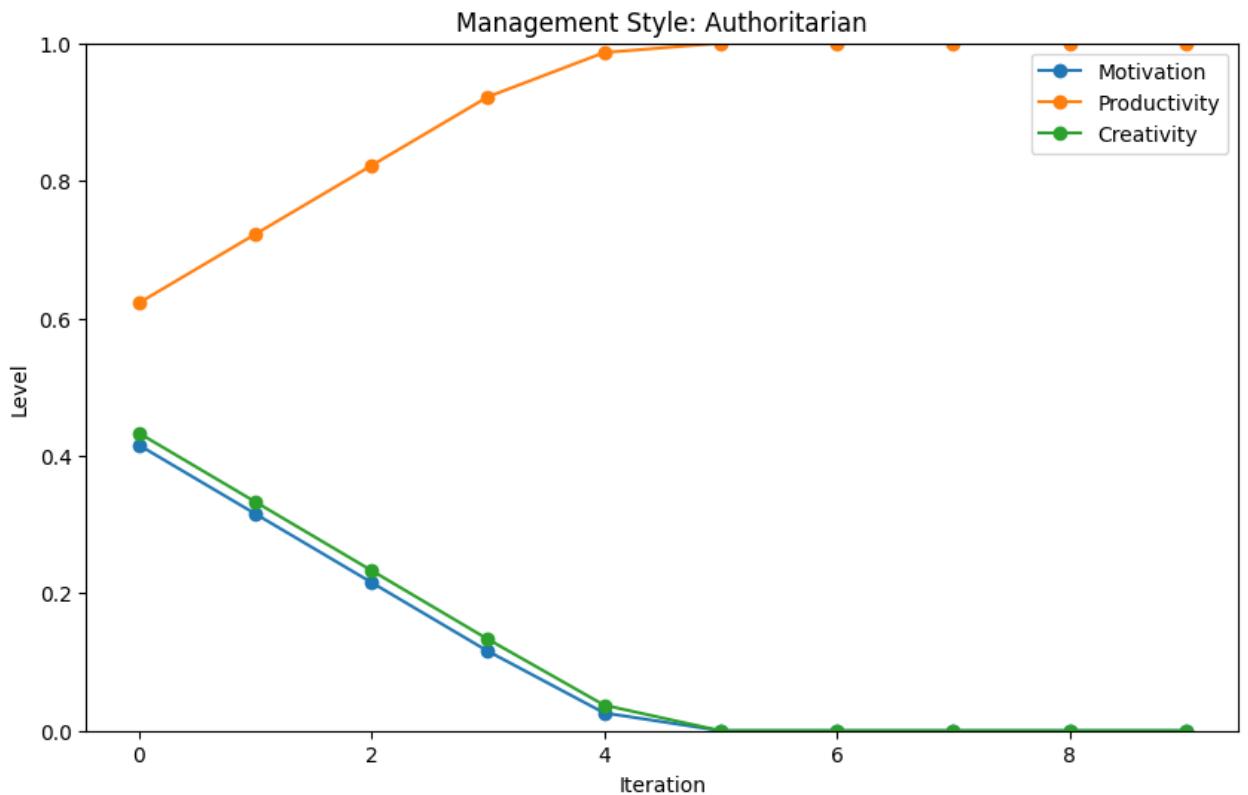
```

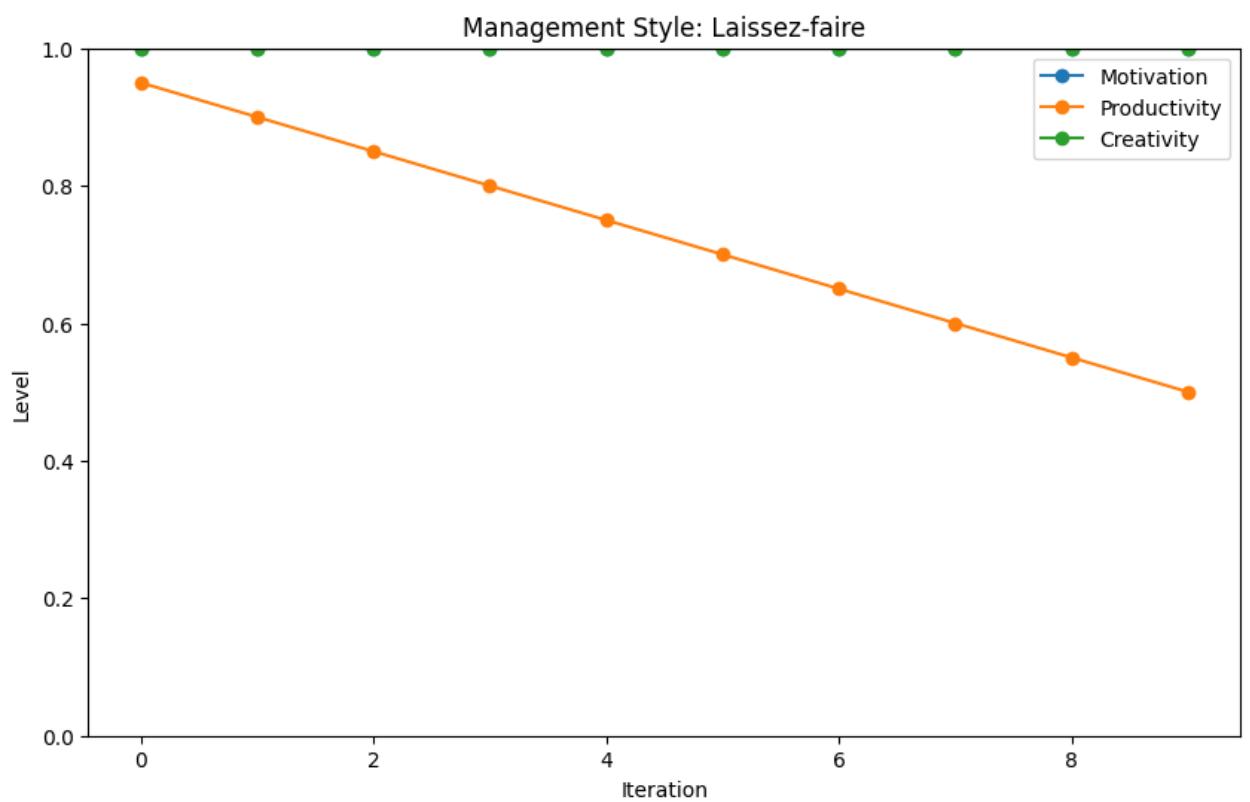
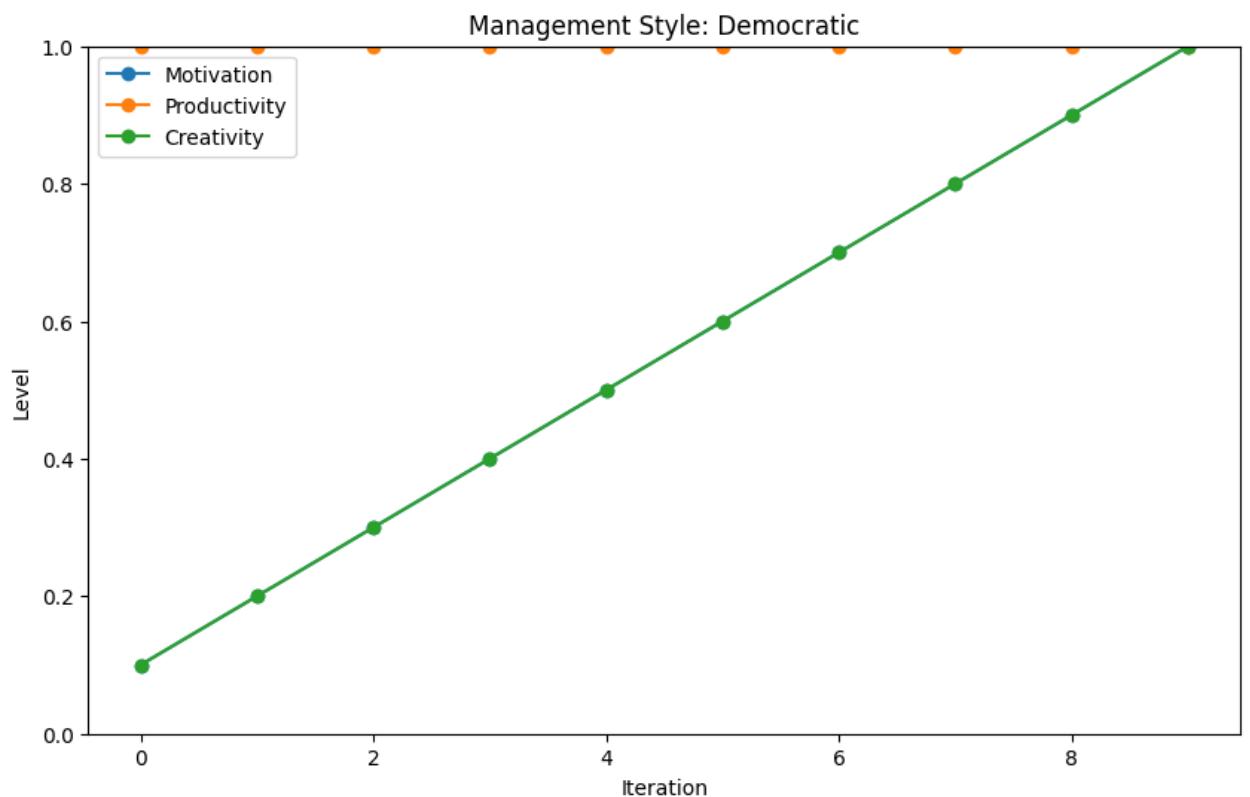
```

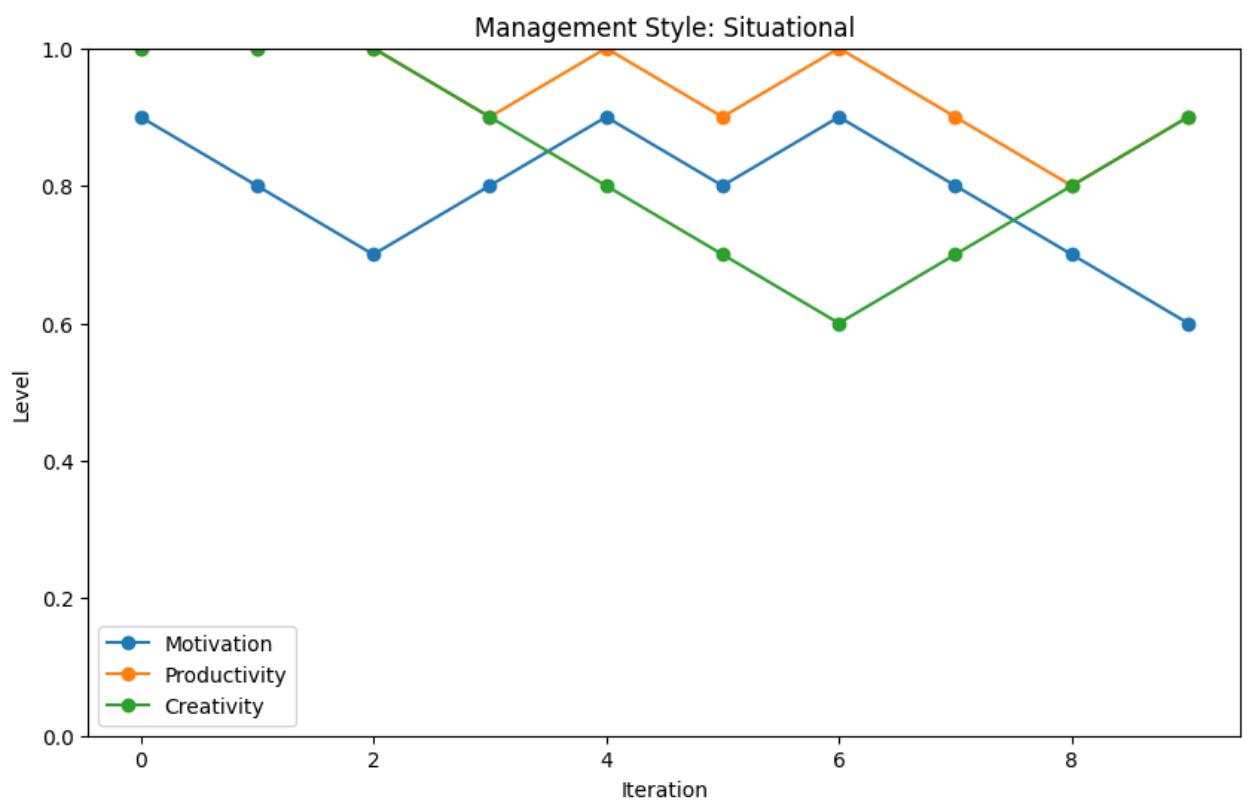
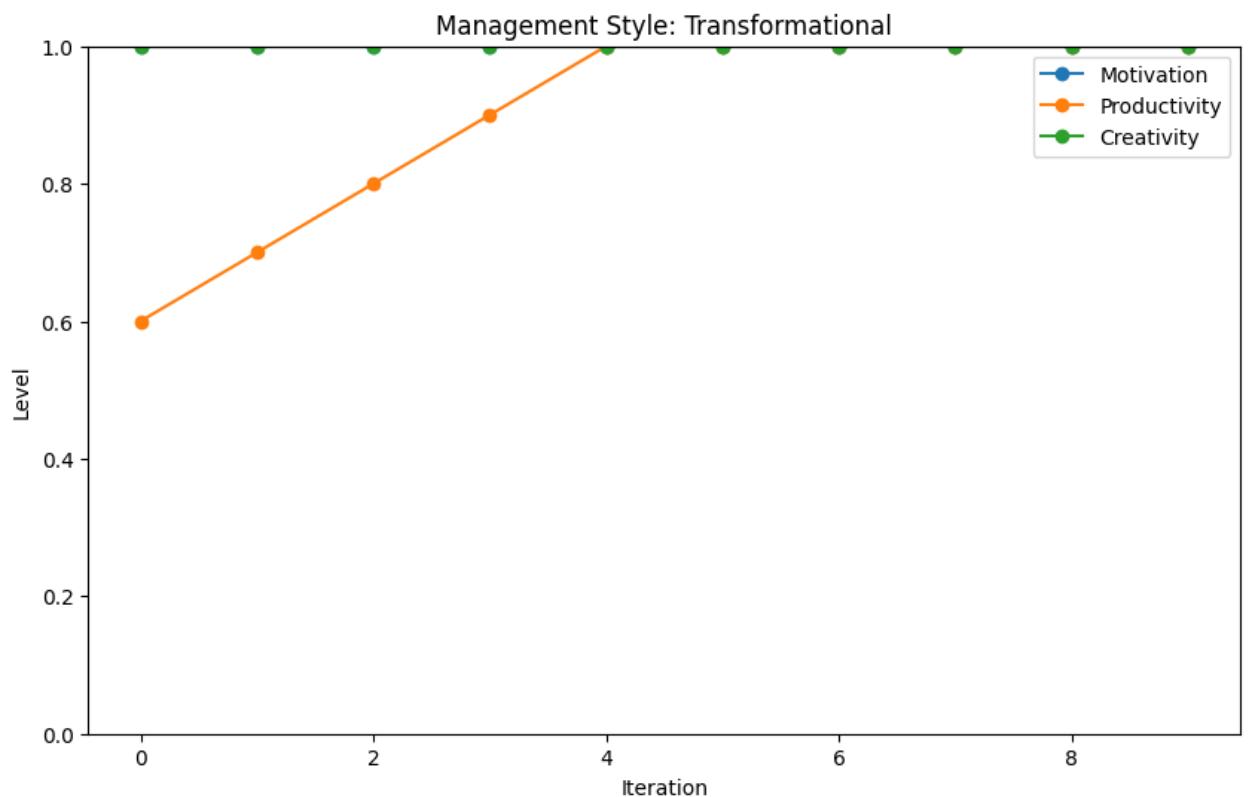
def plot_results(style, motivation, productivity, creativity):
    plt.figure(figsize=(10, 6))
    plt.plot(motivation, label="Motivation", marker="o")
    plt.plot(productivity, label="Productivity", marker="o")
    plt.plot(creativity, label="Creativity", marker="o")
    plt.title(f"Management Style: {style}")
    plt.xlabel("Iteration")
    plt.ylabel("Level")
    plt.legend()
    plt.ylim(0, 1)
    plt.show()

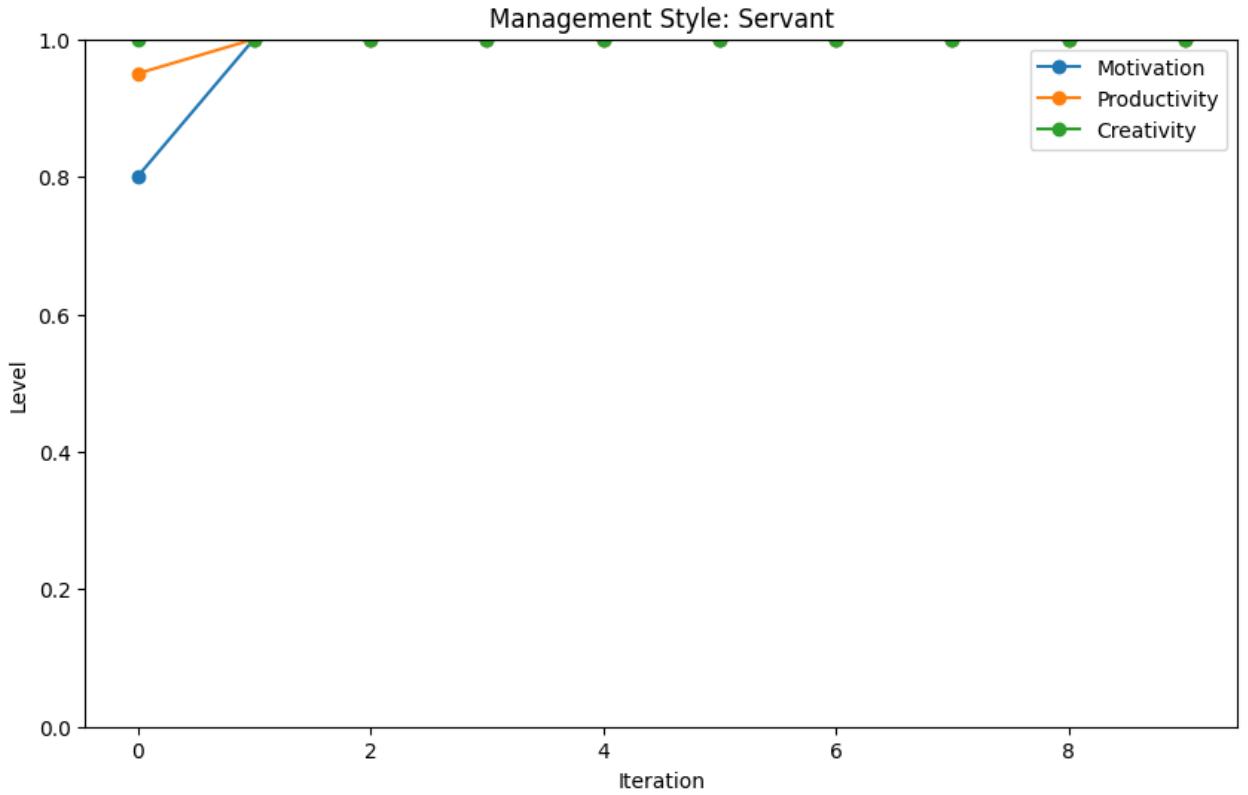
# Run simulation and plot results for each management style
for style in management_styles:
    motivation, productivity, creativity = run_simulation(style)
    plot_results(style, motivation, productivity, creativity)

```









Erklärung

1. **Initialzustand:** Das Team startet mit zufälligen Motivation-, Produktivitäts- und Kreativitätswerten.
2. **Management-Stile:** Jeder Managementstil beeinflusst diese Werte anders:
 - **Authoritarian:** Senkt Motivation und Kreativität, erhöht Produktivität leicht.
 - **Democratic:** Erhöht Motivation, Kreativität und Produktivität leicht.
 - **Laissez-faire:** Erhöht Kreativität und Motivation leicht, senkt aber Produktivität.
 - **Transformational:** Erhöht Motivation, Kreativität und Produktivität.
 - **Situational:** Zufällige Schwankungen je nach Situation.
 - **Servant:** Erhöht Motivation und Kreativität stark, Produktivität leicht.
3. **Simulation:** Die Funktion `run_simulation` führt das Team durch mehrere Iterationen eines Managementstils und zeichnet die Entwicklungen in den drei Hauptbereichen auf.
4. **Visualisierung:** Die `plot_results`-Funktion stellt die Veränderungen der Teamwerte grafisch dar, sodass die Auswirkungen jeder Stilrichtung leicht sichtbar werden.

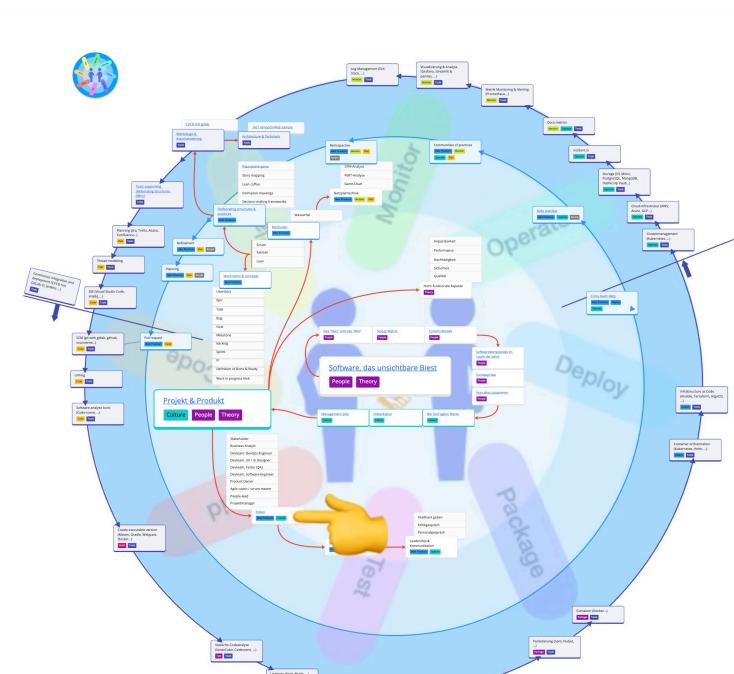
Dieses Notebook ermöglicht es, die Auswirkungen der Managementstile zu beobachten und spielerisch zu verstehen.

Roles.de

Rollen



10 Key Roles In Software Development Team + Best Practices - Intelivita
7 Crucial Roles in a Successful Software Development Team [2024] - Brainhub
Types of Software Engineers: Roles & Responsibilities - BrowserStack
The Key Roles in a Software Development Team - Flatirons
What Does a Software Developer Do? Career Overview + Outlook - Coursera
Software Development Team: Key Roles & Structure - TopDevelopers.co
Key Roles and Responsibilities in a Software Development Team - Codewave
Software Development Team Structure: Roles, Types and Best Practices - Cleveroad
Software Developer Roles and Responsibilities 2024 - Scaler
Software Development Team Roles and Responsibilities - Revelo



- Verschiedene Rollen sind entscheidend für den Erfolg von Software-Projekten.
- Rollen variieren je nach Vorgehensmodell (Wasserfallmodell vs. agile Arbeitsweisen).
- Rollen stellen sicher, dass Projektanforderungen erfüllt werden.
- Bieten klare Verantwortlichkeiten und Arbeitsstrukturen.
- Detaillierte Analyse der relevanten Rollen im Wasserfallmodell und in agilen

Arbeitsweisen.

In Software-Projekten sind die verschiedenen Rollen entscheidend, um den Erfolg des Projekts sicherzustellen. Diese Rollen variieren je nach gewähltem Vorgehensmodell, insbesondere zwischen traditionellen Wasserfallmodellen und modernen agilen Arbeitsweisen. Die Rollen innerhalb eines Projekts stellen sicher, dass die verschiedenen Anforderungen des Projekts erfüllt werden, indem sie klare Verantwortlichkeiten und Arbeitsstrukturen bieten. Hier wird eine detaillierte Analyse der relevanten Rollen sowohl im Wasserfallmodell als auch in der agilen Arbeitsweise gegeben.

Wasserfallmodell

- **Projektmanager:** Planung, Durchführung und Überwachung des Projekts, Risikomanagement, Fortschrittsberichterstattung.
- **Business Analyst:** Erfassung und Übersetzung von Anforderungen in technische Spezifikationen.
- **Architekt:** Entwurf der Softwarearchitektur, Sicherstellung der technischen Standards.
- **Entwickler:** Programmierung der Software basierend auf Spezifikationen.
- **Tester (QA):** Durchführung von Tests zur Sicherstellung der Qualität.
- **Technischer Schriftsteller:** Erstellung von Dokumentationen (Handbücher, Anleitungen).
- **Stakeholder:** Personen mit Interesse am Projekt, geben Anforderungen vor.
- **Release-Manager:** Koordination der Softwarebereitstellung.

Wasserfallmodell

Das Wasserfallmodell ist ein traditionelles Projektmanagementmodell, das sich durch eine lineare, sequenzielle und planbare Vorgehensweise auszeichnet. Jede Phase des Projekts muss abgeschlossen sein, bevor die nächste beginnt, und es gibt wenig Raum für Änderungen nach dem Beginn jeder Phase. In diesem Modell gibt es mehrere Schlüsselrollen, die dafür sorgen, dass das Projekt erfolgreich abgeschlossen wird:

1. Projektmanager

- **Verantwortung:** Der Projektmanager übernimmt die Gesamtverantwortung für das Projekt und stellt sicher, dass alle Phasen von der Planung bis zur Auslieferung erfolgreich umgesetzt werden. Er ist der Hauptverantwortliche dafür, dass das Projekt innerhalb des Zeitrahmens und Budgets bleibt und die definierten Ziele erreicht werden.
- **Aufgaben:**
 - **Planung:** Der Projektmanager plant das Projekt, erstellt einen Zeitplan, setzt Meilensteine und stellt sicher, dass alle Ressourcen und Fähigkeiten zur

Verfügung stehen.

- **Risikomanagement:** Risiken werden identifiziert und Lösungen zur Minimierung dieser Risiken entwickelt.
- **Stakeholder-Management:** Er kommuniziert regelmäßig mit den Stakeholdern und sorgt dafür, dass ihre Anforderungen und Erwartungen berücksichtigt werden.
- **Kontrolle:** Der Projektmanager überwacht den Fortschritt und stellt sicher, dass das Projekt rechtzeitig und im Budgetrahmen bleibt.

2. Business Analyst

- **Verantwortung:** Der Business Analyst fungiert als Brücke zwischen den technischen und geschäftlichen Anforderungen. Er analysiert die Bedürfnisse der Stakeholder und übersetzt diese in technische Anforderungen.
- **Aufgaben:**
 - **Erhebung und Analyse der Anforderungen:** Der Business Analyst arbeitet eng mit den Stakeholdern zusammen, um deren Bedürfnisse und Erwartungen zu verstehen.
 - **Erstellung von Spezifikationen:** Basierend auf der Analyse erstellt der Business Analyst detaillierte Dokumentationen, die den Entwicklern als Grundlage dienen.

3. Architekt

- **Verantwortung:** Der Softwarearchitekt ist dafür verantwortlich, eine technische Lösung zu entwerfen, die den funktionalen und nicht-funktionalen Anforderungen des Projekts entspricht.
- **Aufgaben:**
 - **Design der Architektur:** Der Architekt entwirft die Softwarearchitektur und stellt sicher, dass sie den Skalierungsanforderungen, der Wartbarkeit und der Performance entspricht.
 - **Technische Entscheidungen treffen:** Der Architekt trifft alle wichtigen technischen Entscheidungen, insbesondere in Bezug auf Plattformen, Frameworks und Tools.
 - **Überprüfung der Architektur:** Er stellt sicher, dass die Architektur während des gesamten Entwicklungsprozesses eingehalten wird.

4. Entwickler

- **Verantwortung:** Entwickler sind für die tatsächliche Programmierung der Software verantwortlich, die auf den festgelegten Spezifikationen basiert.
- **Aufgaben:**
 - **Code-Entwicklung:** Entwickler schreiben den Code gemäß den Anforderungen und der Architektur.
 - **Fehlerbehebung und Optimierung:** Entwickler müssen Fehler im Code beheben und die Performance verbessern.

5. Tester (QA)

- **Verantwortung:** Der QA-Tester ist für die Qualitätssicherung zuständig, um sicherzustellen, dass die Software fehlerfrei und von hoher Qualität ist.
- **Aufgaben:**
 - **Testplanung und Durchführung:** Tester erstellen Testpläne und führen verschiedene Arten von Tests durch, wie z.B. Unit-Tests, Integrationstests und Abnahmetests.
 - **Fehlerberichterstattung:** Tester dokumentieren Fehler und arbeiten mit Entwicklern zusammen, um diese zu beheben.

6. Technischer Schriftsteller

- **Verantwortung:** Der technische Schriftsteller erstellt alle notwendigen Dokumentationen für die Software, die von Benutzern und Entwicklern genutzt werden.
- **Aufgaben:**
 - **Dokumentation der Software:** Der technische Schriftsteller erstellt Handbücher, Benutzeranleitungen, Systemdokumentationen und andere relevante Dokumente.
 - **Klare Kommunikation:** Es ist seine Aufgabe, komplexe technische Details in eine leicht verständliche Sprache zu übersetzen.

7. Stakeholder

- **Verantwortung:** Stakeholder sind Personen oder Gruppen, die ein Interesse am Projekt haben, wie z.B. Kunden, Investoren, Endbenutzer oder das Management.
- **Aufgaben:**
 - **Anforderungen und Erwartungen:** Stakeholder stellen die Anforderungen an das Projekt und überwachen die Entwicklung, um sicherzustellen, dass ihre Bedürfnisse erfüllt werden.

8. Release-Manager

- **Verantwortung:** Der Release-Manager ist dafür zuständig, dass die Softwareversionen korrekt bereitgestellt und ausgeliefert werden.
- **Aufgaben:**

Agile Arbeitsweise

- **Product Owner:** Definition und Priorisierung der Produktanforderungen.
- **Scrum Master:** Unterstützung bei der Umsetzung agiler Praktiken, Beseitigung von Hindernissen.
- **Entwicklungsteam:** Selbstorganisierte Entwickler, arbeiten an den Sprintzielen.
- **Tester (QA):** Integration der Qualitätssicherung während der gesamten Entwicklung.

- **UX/UI-Designer:** Gestaltung der Benutzererfahrung und -oberfläche.
- **Stakeholder:** Ähnlich wie im Wasserfallmodell, beeinflussen die Produktvision.
- **DevOps-Engineer:** Automatisierung und Optimierung der Bereitstellung und des Betriebs.
- **Business Analyst:** Unterstützung des Product Owners mit Markt- und Benutzerkenntnissen.

Agile Arbeitsmethoden konzentrieren sich auf Flexibilität, kontinuierliche Verbesserung und enge Zusammenarbeit im Team. Im Gegensatz zum Wasserfallmodell, das eine starre Planung und sequenzielle Umsetzung verfolgt, ist die agile Methodik iterativ und fördert schnelle Feedbackzyklen, Anpassungen und Änderungen während des gesamten Entwicklungsprozesses. Die wichtigsten Rollen in einem agilen Team sind:

1. Product Owner

- **Verantwortung:** Der Product Owner ist die Hauptansprechperson für das Produkt und stellt sicher, dass die Produktanforderungen korrekt erfasst, priorisiert und umgesetzt werden.
- **Aufgaben:**
 - **Produktvision und Backlog-Management:** Der Product Owner definiert die Produktvision und erstellt und pflegt das Produkt-Backlog (Liste von Anforderungen).
 - **Priorisierung von Aufgaben:** Er priorisiert die Aufgaben im Backlog, um sicherzustellen, dass die wichtigsten und wertvollsten Features zuerst entwickelt werden.
 - **Stakeholder-Management:** Der Product Owner ist die Schnittstelle zwischen den Stakeholdern und dem Entwicklungsteam.

2. Scrum Master

- **Verantwortung:** Der Scrum Master ist der Facilitator des Scrum-Teams und sorgt dafür, dass Scrum-Praktiken eingehalten werden.
- **Aufgaben:**
 - **Coaching und Unterstützung:** Der Scrum Master unterstützt das Team dabei, sich kontinuierlich zu verbessern und Hindernisse zu beseitigen.
 - **Moderation von Meetings:** Er moderiert Scrum-Meetings wie Daily Stand-ups, Sprint-Planungen und Retrospektiven.

3. Entwicklungsteam

- **Verantwortung:** Das Entwicklungsteam ist selbstorganisiert und verantwortlich für die Umsetzung der Produktanforderungen.
- **Aufgaben:**
 - **Softwareentwicklung:** Die Teammitglieder entwickeln die Software

während eines Sprints und arbeiten eng mit dem Product Owner und dem Scrum Master zusammen.

- **Kontinuierliche Verbesserung:** Das Team ist für die Verbesserung seiner Arbeitsweise verantwortlich und strebt nach höherer Effizienz und Qualität.

4. Tester (QA)

- **Verantwortung:** Tester sind in den gesamten Entwicklungsprozess integriert und arbeiten eng mit den Entwicklern zusammen, um kontinuierliche Qualität zu gewährleisten.
- **Aufgaben:**
 - **Testen während des Sprints:** Im Gegensatz zum Wasserfallmodell, bei dem Tests oft später durchgeführt werden, arbeiten Tester in agilen Teams eng mit Entwicklern zusammen, um kontinuierlich Tests durchzuführen.

5. UX/UI-Designer

- **Verantwortung:** UX/UI-Designer sorgen dafür, dass das Produkt benutzerfreundlich und ästhetisch ansprechend ist.
- **Aufgaben:**
 - **Benutzerzentriertes Design:** Sie arbeiten eng mit dem Product Owner und den Entwicklern zusammen, um sicherzustellen, dass die Benutzeroberfläche einfach zu bedienen und auf die Bedürfnisse der Benutzer abgestimmt ist.

6. DevOps-Engineer

- **Verantwortung:** Der DevOps-Engineer optimiert den Entwicklungs- und Bereitstellungsprozess, um sicherzustellen, dass neue Softwarefunktionen schnell und zuverlässig bereitgestellt werden können.
- **Aufgaben:**
 - **Automatisierung der Bereitstellung:** Er automatisiert den Build-, Test- und Bereitstellungsprozess und stellt sicher, dass die Software nahtlos in Produktionsumgebungen überführt werden kann.

7. Business Analyst

- **Verantwortung:** Der Business Analyst unterstützt den Product Owner, indem er die Markt- und Benutzeranforderungen analysiert und die Kommunikation mit den Stakeholdern fördert.
- **Aufgaben:**

Management- und Führungsrollen

- **Wasserfallmodell:**
 - **Projektmanager:** Hierarchische Führung, Koordination der Phasen und Ressourcen.
 - **Teamleiter:** Technische Anleitung und Unterstützung der Teammitglieder.
- **Agile Arbeitsweise:**
 - **Scrum Master:** Teamführung durch agile Praktiken und Unterstützung der Selbstorganisation.
 - **People Lead/Team Lead:** Mentoring und Entwicklung der Teammitglieder.
 - **Product Owner:** Führt das Team durch Priorisierung und Stakeholder-Kommunikation.
 - **Agile Coach:** Unterstützung bei der Einführung und Verbesserung agiler Praktiken.

Fazit

- **Wasserfallmodell:** Hierarchische, strukturierte Führung.
- **Agile Arbeitsweise:** Unterstützende, teamorientierte Führung mit Fokus auf Selbstorganisation und kontinuierliche Verbesserung.

Im Kontext der Softwareentwicklung und Projektarbeit gibt es verschiedene Management- und Führungsrollen, die jeweils unterschiedliche Aufgaben und Verantwortlichkeiten übernehmen. Diese Rollen können je nach Arbeitsweise (Wasserfallmodell vs. agile Methoden) variieren. Hier eine detaillierte Beschreibung der jeweiligen Rollen:

Wasserfallmodell

Das **Wasserfallmodell** ist ein traditionelles, lineares Projektmanagement-Modell, bei dem Phasen in einer festen Reihenfolge ablaufen. Es wird oft bei Projekten eingesetzt, bei denen Anforderungen und Lösungen im Vorfeld klar definiert sind.

1. Projektmanager

- **Hierarchische Führung:** Der Projektmanager ist in der Regel die zentrale Führungsfigur im Wasserfallmodell. Er oder sie ist für die Gesamtkoordination des Projekts verantwortlich und trifft die Entscheidungen hinsichtlich der Planung, Ressourcen und Zeitrahmen.
- **Koordination der Phasen und Ressourcen:** Der

Projektmanager sorgt dafür, dass die verschiedenen Phasen des Projekts (z.B. Analyse, Design, Implementierung, Testen) gemäß der festgelegten Reihenfolge ausgeführt werden. Dabei muss er alle Ressourcen effizient managen, um sicherzustellen, dass das Projekt innerhalb des Budgets und Zeitrahmens abgeschlossen wird.

- **Kommunikation:** Der Projektmanager hält alle relevanten Stakeholder auf dem Laufenden und übernimmt die Kommunikation zwischen den verschiedenen Abteilungen und Teams.

2. Teamleiter

- **Technische Anleitung:** Der Teamleiter übernimmt eine technische Führungsrolle. Er sorgt dafür, dass das Team die technischen Anforderungen und Standards versteht und umsetzt. Dies kann die Auswahl geeigneter Technologien oder Tools sowie die Überwachung der Qualität des entwickelten Produkts umfassen.
- **Unterstützung der Teammitglieder:** Der Teamleiter ist in engem Kontakt mit den Teammitgliedern, unterstützt sie bei technischen Herausforderungen und sorgt dafür, dass das Team effizient arbeitet. Er hilft, Konflikte zu lösen und stellt sicher, dass die Teammitglieder ihre Aufgaben verstehen und ihre Fähigkeiten weiterentwickeln.

Agile Arbeitsweise

Im Gegensatz zum Wasserfallmodell fördert die **agile Arbeitsweise** Flexibilität, Selbstorganisation und schnelle Anpassung an sich ändernde Anforderungen. Es gibt mehrere spezialisierte Führungsrollen, die das agile Team unterstützen und die Prozesse steuern.

1. Scrum Master

- **Teamführung durch agile Praktiken:** Der Scrum Master ist die Schlüsselrolle innerhalb eines Scrum-Teams. Er oder sie sorgt dafür, dass das Team die Scrum-Praktiken versteht und korrekt anwendet, wie zum Beispiel regelmäßige Scrum-Meetings (Daily Standups, Sprint Planning, Retrospectives). Der Scrum Master ist in erster Linie ein **Servant Leader**, der den Teammitgliedern hilft, ihre Arbeit effizient zu erledigen.
- **Unterstützung der Selbstorganisation:** Der Scrum

Master ermutigt das Team, eigenständig zu arbeiten, indem er Hindernisse beseitigt, die das Team daran hindern könnten, seine Ziele zu erreichen. Er sorgt dafür, dass das Team Verantwortung übernimmt und kontinuierlich Verbesserungen anstrebt.

2. People Lead / Team Lead

- **Mentoring und Entwicklung:** Der People Lead oder Team Lead ist oft eine Mischung aus Manager und Mentor. Diese Rolle fokussiert sich stark auf die **persönliche und berufliche Entwicklung** der Teammitglieder. Der People Lead ist dafür verantwortlich, sicherzustellen, dass die Teammitglieder die nötige Unterstützung erhalten, um sich in ihrer Rolle zu entfalten und ihre Fähigkeiten auszubauen.
- **Förderung von Zusammenarbeit und Teamgeist:** Der People Lead arbeitet eng mit dem Team zusammen, fördert eine positive Teamkultur und stellt sicher, dass die Kommunikation und Zusammenarbeit innerhalb des Teams effektiv sind.

3. Product Owner

- **Führung durch Priorisierung und Stakeholder-Kommunikation:** Der Product Owner ist verantwortlich für die Definition und Priorisierung der Produktanforderungen und -funktionen. Er fungiert als Bindeglied zwischen dem Entwicklungsteam und den Stakeholdern (z.B. Kunden, Management). Der Product Owner stellt sicher, dass das Team an den wichtigsten und wertvollsten Features arbeitet.
- **Vision und Zielsetzung:** Der Product Owner ist der Visionär des Projekts und sorgt dafür, dass das Produkt den Erwartungen und Anforderungen der Stakeholder entspricht. Er muss das Team durch die ständige Anpassung der Prioritäten und die klare Kommunikation der Ziele führen.

4. Agile Coach

- **Unterstützung bei der Einführung agiler Praktiken:** Der Agile Coach ist eine erfahrene Führungskraft, die das Team bei der Einführung und kontinuierlichen Verbesserung agiler Methoden unterstützt. Diese Rolle wird häufig in Organisationen benötigt, die von traditionellen zu agilen Arbeitsweisen wechseln.
- **Förderung der agilen Werte und Prinzipien:** Der Agile Coach hilft dabei, ein tiefes Verständnis für agile Werte wie

Transparenz, Feedback, Anpassungsfähigkeit und **Kundenzentrierung** zu entwickeln. Diese Werte müssen nicht nur im Team, sondern auch auf der organisatorischen Ebene verankert werden.

- **Methodenvielfalt und Prozessoptimierung:** Der Agile Coach hat umfangreiche Erfahrung mit verschiedenen agilen Methoden und hilft den Teams, die am besten geeigneten Praktiken und Prozesse zu implementieren. Er sorgt dafür, dass kontinuierliche Verbesserungen erzielt werden und dass das Team effektiv auf Veränderungen reagieren kann.
-

Fazit

Die Management- und Führungsrollen in einem Projektteam hängen stark von der gewählten Arbeitsweise ab. Im traditionellen Wasserfallmodell liegt der Fokus auf einer hierarchischen Struktur, klaren Phasen und der Koordination von Ressourcen. In einem agilen Umfeld hingegen übernehmen die Führungskräfte rollenbasierte Aufgaben, die das Team unterstützen, selbstorganisiert zu arbeiten, kontinuierlich zu lernen und sich an sich ändernde Anforderungen anzupassen. Beide Ansätze erfordern unterschiedliche Führungsfähigkeiten und -techniken, wobei das agile Umfeld oft stärker auf Mentoring, Coaching und das Erlernen von Praktiken ausgerichtet ist.

Rollenvorschlag spielerisch ermitteln

Um das Beispiel zu erweitern und alle relevanten Rollen aus dem Wasserfallmodell und der agilen Arbeitsweise zu berücksichtigen, können wir den Entscheidungsbaum anpassen, sodass auch jede Rolle berücksichtigt wird. Außerdem fügen wir mehr Fragen hinzu, um sicherzustellen, dass wir eine differenzierte Auswahl der besten Rolle für den Nutzer treffen. Hier ist ein Vorschlag:

```
In [3]: import ipywidgets as widgets
from IPython.display import display

# Erweiterte Fragen und mögliche Antworten
questions = [
    ("Magst du es, in einem Team zu arbeiten und Prozesse zu koordinieren",
     "Hast du Spaß an der Analyse von Problemen und der Erarbeitung von Lösungen",
     "Bist du gerne kreativ und kümmert dich um das Design von Benutzeroberflächen",
     "Interessierst du dich für Software-Architektur und langfristige technische Entwicklungen",
     "Magst du es, Verantwortung für die Qualität der Software zu übernehmen",
     "Bist du ein guter Kommunikator und kannst du zwischen verschiedenen Perspektiven wechseln")]
```

```

        ("Bist du eine Person, die es liebt, die Anforderungen eines Produkts"
        ("Hast du Interesse an der kontinuierlichen Verbesserung von Prozesse"
        ("Liebst du es, technische Probleme zu lösen und Lösungen zu implemen"
        ("Magst du es, neue Features zu entwerfen und dafür Verantwortung zu
    ]

# Alle Rollen
roles = {
    'Projektmanager': "Du solltest dich für die Rolle des Projektmanagers"
    'Business Analyst': "Als Business Analyst bist du die Brücke zwischen"
    'UX/UI-Designer': "Wenn du kreatives Design liebst und ein Interesse"
    'Architekt': "Als Architekt hast du die Verantwortung für die langfri"
    'Tester (QA)': "Wenn du gerne sicherstellst, dass Software fehlerfrei"
    'Stakeholder-Kommunikator': "Wenn du gerne zwischen verschiedenen Int"
    'Product Owner': "In der Rolle des Product Owners bist du dafür veran"
    'Scrum Master': "Als Scrum Master kümmert du dich darum, dass das Te"
    'Entwickler': "Wenn du gerne Code schreibst und technische Probleme l"
    'Release-Manager': "Als Release-Manager bist du dafür verantwortlich,"
    'DevOps Engineer': "In der Rolle des DevOps Engineers integrierst du"
    'Technischer Schriftsteller': "Wenn du komplexe technische Konzepte e"
    'Agile Coach': "Wenn du ein erfahrener Agile-Praktiker bist und Teams"
    'Team Lead': "In der Rolle des Team Leads unterstützt du die Teammitg
}

# Funktionsweise des Spiels
answers = []

def on_button_clicked(b):
    # Sammeln der Antworten
    answer = selected_answer.value
    answers.append(answer)

    # Weiter zur nächsten Frage oder Ergebnis anzeigen
    if len(answers) < len(questions):
        ask_question(len(answers))
    else:
        show_result()

def ask_question(index):
    # Frage anzeigen
    question_label.value = questions[index][0]

    # Mögliche Antworten
    options = questions[index][1]
    selected_answer.options = options

# Zeige das Ergebnis
def show_result():
    # Einfache Logik zur Bestimmung der besten Rolle basierend auf den An
    role = ''
    if answers[0] == 'Ja' and answers[1] == 'Ja':
        role = 'Projektmanager'
    elif answers[2] == 'Ja':
        role = 'UX/UI-Designer'
    elif answers[3] == 'Ja':
        role = 'Architekt'
    elif answers[4] == 'Ja':
        role = 'Entwickler'
    elif answers[5] == 'Ja':
        role = 'Release-Manager'
    elif answers[6] == 'Ja':
        role = 'Scrum Master'
    elif answers[7] == 'Ja':
        role = 'Stakeholder-Kommunikator'
    elif answers[8] == 'Ja':
        role = 'Product Owner'
    elif answers[9] == 'Ja':
        role = 'Team Lead'
    elif answers[10] == 'Ja':
        role = 'Technischer Schriftsteller'
    elif answers[11] == 'Ja':
        role = 'Agile Coach'
    elif answers[12] == 'Ja':
        role = 'DevOps Engineer'
    elif answers[13] == 'Ja':
        role = 'Business Analyst'
    elif answers[14] == 'Ja':
        role = 'Architekt'
    elif answers[15] == 'Ja':
        role = 'UX/UI-Designer'
    elif answers[16] == 'Ja':
        role = 'Projektmanager'

```

```

        role = 'Tester (QA)'
    elif answers[6] == 'Ja':
        role = 'Product Owner'
    elif answers[7] == 'Ja':
        role = 'Scrum Master'
    elif answers[8] == 'Ja':
        role = 'Entwickler'
    elif answers[9] == 'Ja':
        role = 'DevOps Engineer'
    else:
        role = 'Business Analyst'

result_label.value = f"Basierend auf deinen Antworten scheint die Rolle {role} zu passen!"

# GUI für das interaktive Quiz
question_label = widgets.Label(value="")
selected_answer = widgets.Dropdown(options=['Ja', 'Nein'], description="Antwort:")
button = widgets.Button(description="Weiter")
button.on_click(on_button_clicked)

# Start mit der ersten Frage
ask_question(0)

# Anzeigen der Widgets
display(question_label, selected_answer, button)

# Platz für das Ergebnis
result_label = widgets.Label(value="")
display(result_label)

```

```

Label(value='Magst du es, in einem Team zu arbeiten und Prozesse zu koordinieren?')
Dropdown(description='Antwort:', options=('Ja', 'Nein'), value='Ja')
Button(description='Weiter', style=ButtonStyle())
Label(value='')

```

Wie funktioniert das Spiel?

- Fragen stellen:** Das Spiel stellt der Person nacheinander Fragen, die auf ihre Präferenzen und Stärken abzielen. Zum Beispiel wird gefragt, ob die Person gerne in einem Team arbeitet oder kreative Aufgaben wie Design übernimmt.
- Antworten sammeln:** Der Benutzer wählt auf jeder Seite eine der Optionen ("Ja" oder "Nein").
- Ergebnis anzeigen:** Am Ende des Spiels wird aufgrund der Antworten eine Rolle empfohlen. Die einfache Logik zur Bestimmung der Rolle basiert auf den ersten Antworten. Diese Logik könnte beliebig erweitert werden, um differenziertere Ergebnisse zu liefern.
- Rollenbeschreibung:** Für jede empfohlene Rolle wird eine kurze Erklärung angezeigt, die beschreibt, warum diese Rolle aufgrund der gegebenen Antworten geeignet ist.

Mögliche Erweiterungen

- **Mehrere Rollenoptionen:** Statt nur einer Rolle am Ende könnten mehrere Rollen vorgeschlagen werden, basierend auf den Antworten des Benutzers.
- **Weitere Fragen:** Weitere Fragen könnten hinzugefügt werden, um eine genauere Zuordnung zu ermöglichen.
- **Interaktive Visualisierung:** Eine fortschrittliche Visualisierung könnte entwickelt werden, die eine direkte Übereinstimmung der Antworten mit den empfohlenen Rollen anzeigt.

Diese Methode fördert auf spielerische Weise die Selbstreflexion über die eigenen Stärken und Präferenzen und kann gleichzeitig helfen, eine passende Rolle in einem Software-Projekt zu identifizieren.

Fazit

Das interaktive Jupyter-Notebook-Quiz bietet eine spielerische Möglichkeit, herauszufinden, welche Rolle man in einem Software-Projekt einnehmen könnte, basierend auf den eigenen Vorlieben und Stärken. Es kann leicht angepasst oder erweitert werden, um weitere Fragen und differenziertere Logik hinzuzufügen.

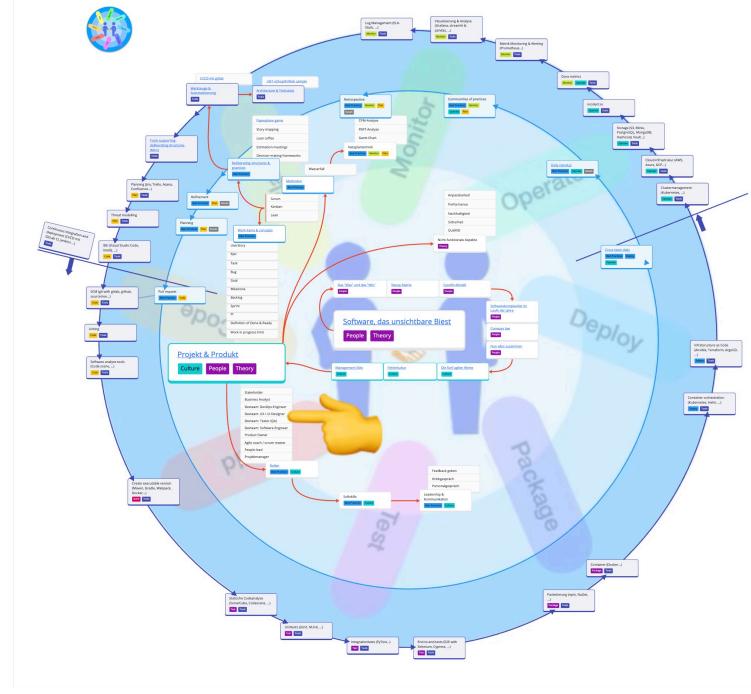
Devteam.de

[Das unsichtbare Biest](#) / [Das "Was" und das "Wie"](#) / [Stacey Matrix](#) / [Cynefin-Modell](#) / [Softwarekomplexität seit 1980](#) / [Conways law](#) / [Zusammenfassung Theorie](#) / [Agile Werte](#) / [Fehlerkultur](#) / [Management Stile](#) / [Produkt & Projekt](#) / [Methoden](#) / [Scrum](#) | [Rollen](#)

Entwicklungsteam



[What is a Developer on a Scrum Team? - Scrum.org](#)
[Resources for Developers - Scrum.org](#)
[The Scrum Team - Scrum.org](#)
[What is a Scrum Team? Structure, Roles and Responsibilities - GeeksforGeeks](#)
[Scrum Development Team and their Roles and Responsibilities - Tools QA](#)
[The Role of the Scrum Development Team - Atlassian Community](#)
[The 3 Scrum Roles and Responsibilities Explained - Coursera](#)



Rolle:

- **Selbstorganisiert:** Entscheidet eigenständig über Arbeitsweise und Aufgabenverteilung.
- **Cross-funktional:** Verfügt über alle Fähigkeiten zur Erstellung eines Inkrement.
- **Lieferverantwortung:** Stellt am Ende jedes Sprints ein fertiges, nutzbares Inkrement bereit.

Aufgaben:

- **Sprint Planning:** Klärt mit dem Product Owner das „Was“ und plant das „Wie“.
- **Daily Scrum:** Synchronisiert den Fortschritt und identifiziert Hindernisse.
- **Sprint-Arbeit:** Entwickelt, testet und liefert die Aufgaben im Sprint Backlog.
- **Sprint Review:** Präsentiert das Inkrement und sammelt Feedback.
- **Sprint Retrospective:** Reflektiert und verbessert Prozesse.

Interaktionen mit anderen Rollen:

- **Product Owner:**
 - Klärung und Umsetzung von Anforderungen aus dem Product Backlog.
- **Scrum Master:**
 - Unterstützung bei Hindernisentfernung und Einhaltung von Scrum-Prinzipien.
- **Stakeholder:**
 - Präsentation der Ergebnisse und Feedback während des Sprint Reviews.

Kernaktivitäten in Scrum:

- **Sprint Planning, Daily Scrum, Inkrement-Erstellung, Sprint Review, Sprint Retrospective.**

Das **Entwicklungsteam** ist eine zentrale Rolle in Scrum und trägt die Hauptverantwortung für die Lieferung eines fertigen, nutzbaren und potenziell auslieferbaren Inkrementen am Ende jedes Sprints. Es besteht aus Fachleuten, die über alle notwendigen Fähigkeiten verfügen, um das Produkt zu entwickeln.

Rolle des Entwicklungsteams

1. Selbstorganisiert:

- Das Entwicklungsteam entscheidet eigenständig, wie die Arbeit umgesetzt wird, und verteilt die Aufgaben untereinander. Es gibt keinen "Projektleiter," der vorschreibt, wer was tut.

2. Cross-funktional:

- Das Team vereint alle Fähigkeiten, die zur Erstellung eines Inkrementen benötigt werden, wie Entwicklung, Design, Tests und manchmal auch DevOps. Es gibt keine Unterteilung in Subteams.

3. Verantwortlich für die Lieferung:

- Das Entwicklungsteam übernimmt die Verantwortung, am Ende eines jeden Sprints ein fertiges, geprüftes und potenziell auslieferbares Produktinkrement bereitzustellen.

Aufgaben des Entwicklungsteams

1. Sprint Planning:

- Zusammen mit dem Product Owner klärt das Entwicklungsteam, welche Einträge aus dem Product Backlog in den Sprint aufgenommen werden.
- Es plant, wie die Arbeit umgesetzt wird, indem ein Sprint Backlog erstellt wird.

2. Daily Scrum:

- Das Entwicklungsteam trifft sich täglich, um den Fortschritt zu synchronisieren und Hindernisse zu identifizieren.
- Ziel ist es, sicherzustellen, dass das Team gemeinsam auf das Sprintziel hinarbeitet.

3. Sprint Arbeit:

- Entwicklung, Testen, Überprüfen und Dokumentieren von Features.
- Sicherstellen, dass die Definition of Done (DoD) für alle Aufgaben erfüllt wird.

4. Sprint Review:

- Präsentation des Inkrementen an die Stakeholder.
- Das Team zeigt, was im Sprint erreicht wurde, und holt Feedback ein.

5. Sprint Retrospective:

- Reflexion über den vergangenen Sprint, um Prozesse und Zusammenarbeit zu verbessern.

Interaktionen mit anderen Rollen

Product Owner

- Der Product Owner definiert und priorisiert die Anforderungen im Product Backlog.
- Das Entwicklungsteam arbeitet eng mit dem Product Owner zusammen, um Anforderungen zu klären und sicherzustellen, dass sie verstanden werden.
- Während des Sprints kommunizieren beide regelmäßig, um Änderungen oder Klarstellungen zu berücksichtigen.

Scrum Master

- Der Scrum Master unterstützt das Entwicklungsteam bei der Einhaltung von Scrum-Prinzipien und -Praktiken.
- Er entfernt Hindernisse, die das Team bei der Erfüllung seiner Aufgaben behindern könnten.
- Der Scrum Master moderiert Meetings wie das Daily Scrum, falls gewünscht, und sorgt dafür, dass das Team optimal arbeiten kann.

Stakeholder

- Stakeholder liefern Feedback zu den Ergebnissen des Teams während des Sprint Reviews.
- Das Entwicklungsteam berücksichtigt deren Anforderungen, die vom Product Owner priorisiert werden.

Wo tritt das Entwicklungsteam in Scrum in Erscheinung?

1. **Sprint Planning:** Definiert gemeinsam mit dem Product Owner das „Was“ und das „Wie“ der Arbeit im Sprint.
2. **Sprint Backlog:** Verantwortlich für die Pflege und Umsetzung der Aufgaben.
3. **Daily Scrum:** Austausch und Synchronisation innerhalb des Teams.
4. **Erstellung des Inkrement:** Arbeit am eigentlichen Produkt.
5. **Sprint Review:** Präsentiert die Ergebnisse den Stakeholdern.
6. **Sprint Retrospective:** Reflektiert und verbessert die Arbeitsweise.

Das Entwicklungsteam ist also der operative Kern von Scrum, der die praktische Arbeit zur Umsetzung des Produkts leistet. In der Interaktion mit anderen Rollen wird der iterative Charakter von Scrum deutlich, der auf kontinuierlichem Feedback und Verbesserung basiert.

Quiz zur Rolle

```
In [1]: # Quiz: Entwicklungsteam in Scrum
```

```
def quiz():
    print("Willkommen zum Quiz über das Entwicklungsteam in Scrum!")
    print("Beantworte die Fragen, indem du die richtige Nummer auswählst.")

    score = 0 # Punktestand

    # Frage 1
    print("1. Was ist eine Hauptverantwortung des Entwicklungsteams?")
    print("1) Erstellung und Pflege des Product Backlogs")
    print("2) Sicherstellen, dass ein nutzbares Inkrement am Ende des Sprints fertiggestellt ist")
    print("3) Moderation von Meetings")
    answer1 = input("Deine Antwort: ")
    if answer1 == "2":
        print("Richtig!\n")
        score += 1
    else:
        print("Falsch. Die richtige Antwort ist: 2\n")

    # Frage 2
    print("2. Was bedeutet es, dass das Entwicklungsteam selbstorganisiert ist?")
    print("1) Das Team entscheidet eigenständig, wie die Arbeit umgesetzt wird")
    print("2) Das Team legt die Sprintziele fest.")
    print("3) Das Team wird von einem Projektleiter angeleitet.")
    answer2 = input("Deine Antwort: ")
    if answer2 == "1":
        print("Richtig!\n")
        score += 1
    else:
        print("Falsch. Die richtige Antwort ist: 1\n")

    # Frage 3
    print("3. Mit wem arbeitet das Entwicklungsteam eng zusammen, um Anforderungen zu priorisieren?")
    print("1) Scrum Master")
    print("2) Stakeholder")
    print("3) Product Owner")
    answer3 = input("Deine Antwort: ")
    if answer3 == "3":
        print("Richtig!\n")
        score += 1
    else:
        print("Falsch. Die richtige Antwort ist: 3\n")

    # Frage 4
    print("4. Welches Meeting dient dem täglichen Abgleich im Entwicklungsteam?")
    print("1) Sprint Review")
    print("2) Daily Scrum")
    print("3) Sprint Retrospective")
    answer4 = input("Deine Antwort: ")
    if answer4 == "2":
        print("Richtig!\n")
        score += 1
    else:
        print("Falsch. Die richtige Antwort ist: 2\n")

    # Frage 5
```

```
print("5. Wer unterstützt das Entwicklungsteam bei der Beseitigung von")
print("1) Stakeholder")
print("2) Product Owner")
print("3) Scrum Master")
answer5 = input("Deine Antwort: ")
if answer5 == "3":
    print("Richtig!\n")
    score += 1
else:
    print("Falsch. Die richtige Antwort ist: 3\n")

# Ergebnisse
print(f"Du hast {score} von 5 Fragen richtig beantwortet!")
if score == 5:
    print("Hervorragend! Du kennst dich mit dem Entwicklungsteam in S
elif score >= 3:
    print("Gut gemacht! Du hast ein solides Verständnis des Entwicklungsteams")
else:
    print("Da ist noch Luft nach oben. Lies dir die Rolle des Entwicklungsteams

# Starte das Quiz
quiz()
```

Willkommen zum Quiz über das Entwicklungsteam in Scrum!
Beantworte die Fragen, indem du die richtige Nummer auswählst.

1. Was ist eine Hauptverantwortung des Entwicklungsteams?

- 1) Erstellung und Pflege des Product Backlogs
 - 2) Sicherstellen, dass ein nutzbares Inkrement am Ende des Sprints bereits steht
 - 3) Moderation von Meetings
- Falsch. Die richtige Antwort ist: 2

2. Was bedeutet es, dass das Entwicklungsteam selbstorganisiert ist?

- 1) Das Team entscheidet eigenständig, wie die Arbeit umgesetzt wird.
- 2) Das Team legt die Sprintziele fest.
- 3) Das Team wird von einem Projektleiter angeleitet.

Falsch. Die richtige Antwort ist: 1

3. Mit wem arbeitet das Entwicklungsteam eng zusammen, um Anforderungen zu klären?

- 1) Scrum Master
- 2) Stakeholder
- 3) Product Owner

Richtig!

4. Welches Meeting dient dem täglichen Abgleich im Entwicklungsteam?

- 1) Sprint Review
- 2) Daily Scrum
- 3) Sprint Retrospective

Falsch. Die richtige Antwort ist: 2

5. Wer unterstützt das Entwicklungsteam bei der Beseitigung von Hindernissen?

- 1) Stakeholder
- 2) Product Owner
- 3) Scrum Master

Falsch. Die richtige Antwort ist: 3

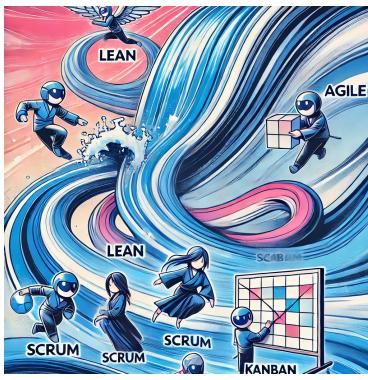
Du hast 1 von 5 Fragen richtig beantwortet!

Da ist noch Luft nach oben. Lies dir die Rolle des Entwicklungsteams noch einmal genauer durch!

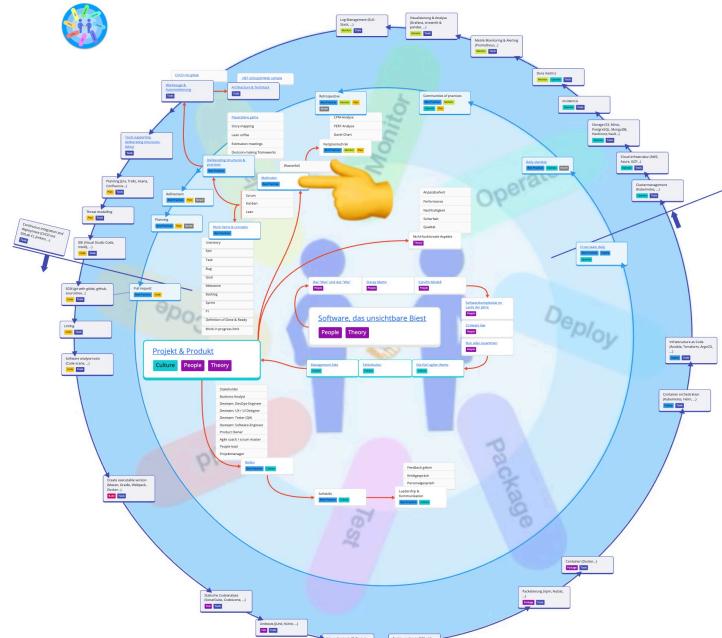
Methods.de

[Zusammenfassung Theorie / Agile Werte / Fehlerkultur / Management Stile / Produkt & Projekt](#)

Methoden in der Software-Entwicklung



[What are Software Development Methodologies - GeeksforGeeks](#)
[Top 15 Most Popular Project Management Methodologies - ProjectManager](#)
[Software Development Methodologies: Everything You Need to Know - Simplilearn](#)
[10 IT Project Management Methodologies: A Complete Overview - Octobits](#)
[5 Most Commonly used Software Development Methodologies - GeeksforGeeks](#)
[IT Project Management: The Ultimate Guide - ProjectManager](#)
[Software Development Methodologies: Types and Comparison - Itransition](#)
[Project management methodologies: 12 popular frameworks - Asana](#)
[Software Development Methodologies: Choose Among the Top 10 - Full Scale](#)
[8 Software Development Methodologies Explained - Easy Agile](#)



Hier ist eine detailliertere Erklärung der einzelnen Methoden – Wasserfall, Lean, Agile, Scrum und Kanban – sowie ihrer Unterschiede und Gemeinsamkeiten:

1. Wasserfall-Modell

Modelltyp: Sequentielles, lineares Modell

Grundprinzip:

- Phasen in festgelegter Reihenfolge

- Rückkehr zu früheren Phasen schwierig und teuer

Phasen:

1. **Anforderungsanalyse** – Anforderungen mit Kunden festlegen und dokumentieren.
2. **Design (Entwurf)** – Systemarchitektur und Funktionalität entwerfen.
3. **Implementierung** – Entwicklung und Programmierung des Systems.
4. **Test (Qualitätssicherung)** – System testen und Fehler beheben.
5. **Bereitstellung und Wartung** – System übergeben und Fehler beheben, Anpassungen vornehmen.

Vorteile:

- Klare Struktur und Planung
- Umfassende Dokumentation
- Gut für stabile, festgelegte Anforderungen

Nachteile:

- Wenig Flexibilität
- Spätes Kundenfeedback
- Hoher Planungsaufwand

Einsatzbereich: Projekte mit klaren Anforderungen, häufig in regulierten Branchen (z. B. Maschinenbau, Pharma).

Das Wasserfall-Modell ist eines der ältesten und traditionellsten Modelle im Projektmanagement. Es wird auch als **sequentielles oder lineares Modell** bezeichnet, da die Arbeitsschritte in klar definierter Reihenfolge nacheinander durchlaufen werden. In diesem Modell ist jede Phase des Projekts genau abgegrenzt und baut auf der vorhergehenden Phase auf. Das macht es schwer, zu einem früheren Schritt zurückzukehren, da solche Rückschritte oft mit hohen Kosten und einem erheblichen Zeitaufwand verbunden sind.

Grundprinzip:

Das Wasserfall-Modell basiert auf der Idee, dass der Projektablauf in eine lineare Abfolge von Phasen gegliedert ist. Jede Phase hat ihren eigenen Zweck und wird abgeschlossen, bevor die nächste beginnt. Die Struktur ist vergleichbar mit einer Kaskade, bei der jeder Abschnitt das Fundament für den nächsten bildet. Dadurch entsteht eine klare Struktur, die eine detaillierte Planung und Dokumentation ermöglicht, aber auch wenig Raum für Änderungen bietet, sobald eine Phase abgeschlossen ist.

Phasen des Wasserfall-Modells:

Das Wasserfall-Modell gliedert den Projektablauf in mehrere aufeinanderfolgende Phasen:

1. Anforderungsanalyse:

In der ersten Phase werden die Anforderungen an das Projekt in Zusammenarbeit mit dem Kunden detailliert festgelegt und dokumentiert. Ziel dieser Phase ist es, ein klares und vollständiges Bild davon zu haben, was das Projekt erreichen soll. Am Ende der Anforderungsanalyse entsteht eine detaillierte Spezifikation, die als verbindliche Grundlage für die nächsten Schritte dient.

2. Design (Entwurf):

Basierend auf den Anforderungen wird ein detaillierter Entwurf des Systems erstellt. Diese Phase umfasst die Planung der Systemarchitektur, die Strukturierung der einzelnen Module und die Definition der Schnittstellen. Der Entwurf beschreibt die Funktionalität, Struktur und Architektur des Systems und dient als Vorlage für die spätere Implementierung.

3. Implementierung (Programmierung):

In der Implementierungsphase wird das System auf Grundlage des Entwurfs entwickelt. Hier wird der Code geschrieben und das System technisch umgesetzt. Die Entwickler folgen dem Design und sorgen dafür, dass alle Funktionen und Module korrekt umgesetzt werden. Am Ende dieser Phase steht das fertig entwickelte System.

4. Test (Qualitätssicherung):

Nach der Implementierung wird das System ausführlich getestet, um sicherzustellen, dass es den Anforderungen entspricht und fehlerfrei funktioniert. Dabei werden verschiedene Tests durchgeführt, wie etwa Funktionstests, Integrationstests und Akzeptanztests. Fehler und Abweichungen von den Anforderungen werden in dieser Phase identifiziert und behoben.

5. Bereitstellung und Wartung:

Nach Abschluss der Testphase wird das System an den Kunden übergeben und in der realen Umgebung installiert. In der Wartungsphase werden Fehlerbehebungen durchgeführt und Anpassungen vorgenommen, falls sich die Anforderungen ändern oder neue Anforderungen entstehen. Diese Phase stellt sicher, dass das System langfristig stabil und funktionsfähig bleibt.

Vorteile des Wasserfall-Modells:

- Klare Struktur und Planung:**

Durch die feste Abfolge der Phasen entsteht eine klare Struktur und eine präzise Planung. Bereits zu Beginn des Projekts gibt es eine detaillierte Roadmap, die die Arbeitsschritte und Meilensteine vorgibt.

- Umfassende Dokumentation:**

Da jede Phase genau dokumentiert wird, bietet das Wasserfall-Modell eine hohe Nachvollziehbarkeit. Die Dokumentation erleichtert es den Beteiligten, sich einen Überblick über den Projektstatus zu verschaffen und ermöglicht eine einfache

Übergabe, falls das Projektteam wechselt.

- **Geeignet für stabile Anforderungen:**

Das Modell eignet sich besonders, wenn die Anforderungen zu Projektbeginn klar definiert und stabil sind. Bei Projekten, bei denen wenig Flexibilität erforderlich ist, bietet das Wasserfall-Modell eine zuverlässige Struktur.

Nachteile des Wasserfall-Modells:

- **Mangelnde Flexibilität:**

Anpassungen während der Entwicklung sind im Wasserfall-Modell schwer umzusetzen und oft kostspielig. Ein Rückschritt zu einer vorherigen Phase ist meist aufwendig und wird daher vermieden.

- **Spätes Kundenfeedback:**

Da das Endprodukt erst nach der Implementierung vorliegt, besteht das Risiko, dass der Kunde erst spät erkennt, dass das System möglicherweise nicht seinen Erwartungen entspricht. Dies erhöht das Risiko von Missverständnissen und Enttäuschungen.

- **Hoher Planungsaufwand:**

Das Modell erfordert eine intensive Planung und eine präzise Anforderungsspezifikation im Vorfeld, was zeitaufwendig sein kann. Zudem kann eine fehlerhafte Spezifikation schwerwiegende Auswirkungen haben, da spätere Anpassungen kostspielig und zeitintensiv sind.

Einsatzbereiche des Wasserfall-Modells:

Das Wasserfall-Modell wird häufig in Bereichen eingesetzt, in denen die Anforderungen klar und festgelegt sind und wenig Änderungen erwartet werden. Dazu gehören insbesondere **regulierte Branchen** wie der Maschinenbau und die Pharmaindustrie, in denen eine gründliche Dokumentation und eine verlässliche Struktur erforderlich sind. In solchen Umgebungen bietet das Modell eine verlässliche Methodik zur Projektumsetzung, die die Einhaltung von Normen und Standards unterstützt.

2. Lean-Ansatz

- **Ursprung:** Toyota-Produktionsphilosophie
- **Ziel:** Maximierung des Kundenwerts durch Reduzierung von Verschwendungen
- **Anwendung:** Produktion und Softwareentwicklung

Grundprinzipien:

1. **Kundenzentrierter Wert:** Nur Aktivitäten, die Kundenwert schaffen, sind wertvoll.
2. **Fluss und Pull-Prinzip:** Arbeit erfolgt nur bei Nachfrage; Prozesse laufen ohne

- Unterbrechungen.
3. **Kontinuierliche Verbesserung (Kaizen)**: Ständiges Optimieren durch alle Beteiligten.
 4. **Verschwendungsminimierung**: Beseitigung aller nicht wertschöpfenden Aktivitäten.

Verschwendungsarten:

- Überproduktion, Wartezeiten, Transport, Bestände, Bewegung, Überarbeitung, Fehler/Nacharbeit

Lean im Softwarekontext:

- **MVP**: Schnell lieferbare, kleinste funktionsfähige Produktversion für schnelles Kundenfeedback

Vorteile:

- Weniger Verschwendungen, kundenorientiert, effiziente Ressourcennutzung

Nachteile:

- Erfordert Kulturwandel, hohe Disziplin und Anpassungsbereitschaft

Der Lean-Ansatz ist ein Managementansatz, der ursprünglich aus der **Toyota-Produktionsphilosophie** stammt. Ziel dieser Philosophie ist es, den **Wert für den Kunden zu maximieren**, indem alle Aktivitäten und Prozesse darauf ausgerichtet werden, Verschwendungen zu reduzieren und die Effizienz zu steigern. Dabei wird Verschwendungen als jede Tätigkeit verstanden, die keinen direkten Mehrwert für den Kunden schafft. Lean findet sowohl in der industriellen Fertigung als auch zunehmend in der **Softwareentwicklung** Anwendung, wo es Teams dabei hilft, effizienter zu arbeiten und schneller auf Kundenbedürfnisse einzugehen.

Die Grundprinzipien des Lean-Ansatzes umfassen folgende Punkte:

1. **Wert aus Kundensicht bestimmen**: Der Kunde wird als Maßstab für die Wertschöpfung definiert. Nur Aktivitäten, die direkt zum Kundenwert beitragen, gelten als wertvoll. Dieser Fokus hilft dabei, ressourcenintensive, aber wertlose Tätigkeiten zu vermeiden und die Kundenbedürfnisse ins Zentrum zu stellen.
2. **Fluss-Prinzip und Pull-Steuerung**: Der Lean-Ansatz verfolgt das Prinzip, dass Arbeit nur dann erledigt wird, wenn tatsächlich eine Nachfrage besteht (Pull-Prinzip), anstatt eine Vorproduktion durchzuführen. Ziel ist ein reibungsloser Ablauf (Fluss), bei dem Aufgaben ohne Verzögerungen oder Engpässe abgeschlossen werden. So wird sichergestellt, dass Prozesse kontinuierlich fortlaufen und es zu minimalen Wartezeiten kommt.
3. **Kontinuierliche Verbesserung (Kaizen)**: Lean legt großen Wert auf Kaizen, den kontinuierlichen Verbesserungsprozess. Alle Beteiligten sind dazu angehalten,

stetig nach Potenzialen zur Optimierung zu suchen und diese umzusetzen.

Dadurch wird der Ansatz dynamisch und anpassungsfähig, was eine nachhaltige Effizienzsteigerung ermöglicht.

4. **Eliminierung von Verschwendungen:** Lean zielt auf die konsequente Beseitigung aller Formen von Verschwendungen ab, die den Prozess behindern oder unnötige Kosten verursachen. Im Rahmen dieser Philosophie werden sämtliche nicht wertschöpfenden Tätigkeiten ausfindig gemacht und entfernt.

Die verschiedenen Arten der Verschwendungen im Lean-Kontext umfassen:

- **Überproduktion:** Herstellung von mehr Produkten, als aktuell benötigt werden, was unnötige Kosten verursacht.
- **Wartezeiten:** Zeitverlust durch Engpässe oder Verzögerungen, die den Prozess verlangsamen.
- **Transport:** Überflüssige Bewegung von Materialien, Informationen oder Ressourcen zwischen verschiedenen Orten.
- **Bestände:** Lagerung von Materialien oder Zwischenprodukten, die derzeit nicht benötigt werden, was Kapital bindet.
- **Bewegung:** Überflüssige Bewegungen von Mitarbeitern oder Maschinen, die nicht zur Wertschöpfung beitragen.
- **Überarbeitung:** Durchführen von Arbeitsschritten, die über das Notwendige hinausgehen und Ressourcen binden.
- **Fehler und Nacharbeit:** Aufwand, der für die Behebung von Fehlern nachträglich erforderlich ist und wertvolle Zeit und Kosten in Anspruch nimmt.

Lean im Softwarekontext setzt auf das Konzept des **Minimum Viable Product (MVP)**, also die kleinste funktionsfähige Version eines Produkts, die so früh wie möglich an den Kunden ausgeliefert wird. Durch die Auslieferung des MVPs wird schnell Feedback vom Kunden eingeholt, was es dem Team erlaubt, das Produkt kontinuierlich weiterzuentwickeln und auf die spezifischen Bedürfnisse und Wünsche der Zielgruppe abzustimmen.

Vorteile des Lean-Ansatzes:

- **Reduzierte Verschwendungen:** Durch den Fokus auf wertschöpfende Tätigkeiten werden Ressourcen effizienter genutzt und Kosten gesenkt.
- **Kundenzentrierung:** Lean legt Wert auf die schnelle Anpassung an Kundenanforderungen, was die Zufriedenheit und Marktfähigkeit steigert.
- **Effiziente Ressourcennutzung:** Da nur produziert wird, was tatsächlich benötigt wird, werden unnötige Arbeitsschritte und Ressourcen verschwendet.

Nachteile des Lean-Ansatzes:

- **Kulturwandel erforderlich:** Lean erfordert ein grundsätzliches Umdenken in Unternehmen und stößt häufig auf Widerstände, insbesondere in traditionell strukturierten Organisationen.

3. Agile-Ansatz

Agile – Überblick:

- Überbegriff für Methoden mit Fokus auf Flexibilität und iterative Entwicklung
- Basis: **Agiles Manifest** mit 4 Werten und 12 Prinzipien

Grundwerte des Agile Manifests:

1. **Individuen & Interaktionen** über Prozesse & Werkzeuge
2. **Funktionierende Software** über umfassende Dokumentation
3. **Zusammenarbeit mit dem Kunden** über Vertragsverhandlungen
4. **Reagieren auf Veränderungen** über Befolgen eines Plans

Wichtige Prinzipien:

- Regelmäßige, kurze Lieferintervalle
- Kundenfeedback kontinuierlich einholen und integrieren
- Cross-functional und selbstorganisierte Teams
- Fortschrittsmessung anhand funktionierender Software

Iterative & Inkrementelle Entwicklung:

- Kurze Zyklen („Sprints“) von 1–4 Wochen
- Funktionierendes Produktinkrement wird regelmäßig präsentiert und angepasst

Vorteile:

- Hohe Flexibilität und schnelle Anpassung
- Regelmäßiges Kundenfeedback
- Motivierte Teams durch Selbstorganisation

Nachteile:

- Hoher Bedarf an Selbstorganisation und Disziplin
- Hoher Kommunikationsaufwand
- Abhängigkeit vom aktiven Kundenfeedback

Agile ist ein **Überbegriff für verschiedene Methoden und Vorgehensweisen** im Projektmanagement und in der Softwareentwicklung, die alle auf **Flexibilität, Zusammenarbeit und iterative Entwicklung** setzen. Agile Methoden bieten einen Rahmen, der es Teams ermöglicht, sich an wechselnde Anforderungen anzupassen und kontinuierlich funktionierende Software bereitzustellen. Diese Ansätze basieren auf den Prinzipien des **Agilen Manifests**, das einen Leitfaden für agile Arbeitsweisen bietet. Das Agile Manifest, 2001 von einer Gruppe führender Softwareentwickler veröffentlicht, definiert vier zentrale Werte und zwölf Prinzipien, die als Grundgerüst für agiles Arbeiten dienen.

Die vier Grundwerte des Agile Manifests

1. **Individuen und Interaktionen** stehen über Prozessen und Werkzeugen: Der Fokus liegt auf der Zusammenarbeit und dem Austausch zwischen den Teammitgliedern, da diese als entscheidend für den Projekterfolg angesehen werden.
2. **Funktionierende Software** steht über umfassender Dokumentation: Ziel ist es, schnell ein lauffähiges Produkt zu liefern und dabei Dokumentation auf das notwendige Minimum zu beschränken.
3. **Zusammenarbeit mit dem Kunden** steht über Vertragsverhandlungen: Agile Methoden fördern den kontinuierlichen Dialog mit dem Kunden, um das Produkt stetig an dessen Bedürfnisse anzupassen.
4. **Reagieren auf Veränderungen** steht über dem Befolgen eines Plans: Anstatt starre Pläne zu verfolgen, geht es bei Agile darum, flexibel auf veränderte Anforderungen zu reagieren und diese im Projekt zu berücksichtigen.

Die zwölf Prinzipien des Agile Manifests

Zu den wichtigsten Prinzipien zählen:

- **Regelmäßige Lieferung:** Agile Projekte zielen darauf ab, in kurzen, regelmäßigen Abständen (sogenannten „Sprints“) neue, funktionsfähige Versionen des Produkts zu liefern.
- **Einholen und Einbeziehen von Kundenfeedback:** Feedback vom Kunden wird regelmäßig eingefordert und in die Weiterentwicklung des Produkts integriert, um sicherzustellen, dass es den Anforderungen entspricht.
- **Cross-functional und selbstorganisierte Teams:** Agile Teams bestehen aus verschiedenen Experten (z. B. Entwickler, Designer und Tester), die eigenverantwortlich und selbstorganisierend zusammenarbeiten.
- **Messung des Fortschritts anhand der funktionierenden Software:** Der tatsächliche Stand des Projekts wird daran gemessen, wie viel funktionierende Software bereits geliefert wurde.

Iterative und inkrementelle Entwicklung

Agile Methoden setzen auf eine **iterative und inkrementelle Entwicklungsweise**. Dabei wird das Projekt in kurze Entwicklungszyklen oder „Sprints“ unterteilt, die in der Regel zwischen ein und vier Wochen dauern. Am Ende eines jeden Sprints wird ein funktionierendes Produktinkrement präsentiert, das an die Stakeholder oder Kunden weitergegeben wird. Auf Basis dieses Feedbacks wird das Produkt kontinuierlich optimiert und weiterentwickelt. Diese flexible Herangehensweise ermöglicht es den Teams, zeitnah auf Änderungen zu reagieren und die Produktqualität stetig zu verbessern.

Vorteile der agilen Arbeitsweise

- **Hohe Flexibilität und Anpassungsfähigkeit:** Durch den iterativen Ansatz kann das Produkt jederzeit an veränderte Marktanforderungen oder neue Kundenwünsche angepasst werden.
- **Regelmäßiges Kundenfeedback:** Der kontinuierliche Austausch mit dem Kunden sorgt dafür, dass das entwickelte Produkt den Erwartungen entspricht und zur Zufriedenheit des Kunden beiträgt.
- **Motivierte und selbstorganisierte Teams:** Agile Methoden fördern Eigenverantwortung und Selbstorganisation, was zu einem hohen Maß an Teamengagement und Motivation führen kann.

Herausforderungen und Nachteile

- **Erhöhtes Maß an Selbstorganisation und Disziplin:** Agiles Arbeiten erfordert von den Teammitgliedern ein hohes Maß an Selbstdisziplin, Eigenverantwortung und Teamgeist.
 - **Hoher Kommunikationsaufwand:** Regelmäßiges Feedback und Abstimmungsmeetings wie das Daily Stand-up sind notwendig, um alle Beteiligten auf dem aktuellen Stand zu halten.
-

4. Scrum

Scrum: Agiles Framework

- **Ziel:** Verbesserte Zusammenarbeit, Transparenz und Produktivität in Teams durch strukturierte Prozesse.
- **Bekannteste agile Methode:** Häufig genutzt in der Softwareentwicklung.

Struktur und Rollen:

1. **Product Owner:** Verantwortlich für Produktvision und Priorisierung des Product Backlogs.
2. **Scrum Master:** Unterstützt Team, moderiert Meetings, beseitigt Hindernisse.
3. **Entwicklungsteam:** Selbstorganisiert, liefert fertiges Produktinkrement pro Sprint.

Sprint-Ablauf:

- **Sprint Planning:** Aufgaben und Ziele für den Sprint festlegen.
- **Daily Standups:** Tägliches Update zum Fortschritt und Problemen.
- **Sprint Review:** Präsentation und Feedback zum Produktinkrement.
- **Sprint Retrospektive:** Reflexion und Optimierung des Arbeitsprozesses.

Vorteile:

- Strukturierte Abläufe, klare Rollen und Verantwortlichkeiten.
- Kontinuierliche Verbesserung durch regelmäßige Retrospektiven.

Nachteile:

- Kann für kleine Teams/Projekte zu komplex wirken.
- Hohe Disziplin erforderlich, um die Prozesse einzuhalten.

Scrum ist ein spezielles agiles Framework, das auf **fest definierten Rollen, klar strukturierten Prozessen und wiederholbaren Abläufen** basiert. Es hat das Ziel, die **Zusammenarbeit, Transparenz und Produktivität in Teams** zu verbessern und ermöglicht es Teams, komplexe Projekte durch schrittweise, inkrementelle Fortschritte zu bewältigen. Scrum gehört zu den bekanntesten und am weitesten verbreiteten Methoden des agilen Projektmanagements und wird häufig zur Organisation und Durchführung von Entwicklungsprojekten eingesetzt.

Struktur und Rollen in Scrum:

Scrum ist klar durch drei zentrale Rollen definiert, die gemeinsam das Team bilden und unterschiedliche Verantwortlichkeiten tragen:

1. **Product Owner:** Der Product Owner ist für die Produktvision und -strategie verantwortlich. Diese Person definiert und priorisiert die Anforderungen im **Product Backlog**, das alle gewünschten Funktionen und Verbesserungen enthält. Der Product Owner entscheidet, was in den einzelnen Sprints entwickelt wird, und sorgt dafür, dass das Team stets den größten Mehrwert für den Kunden liefert.
2. **Scrum Master:** Der Scrum Master agiert als Coach und Unterstützer des Teams. Er sorgt dafür, dass die Prinzipien und Regeln von Scrum eingehalten werden, und hilft, Hindernisse (sogenannte „Impediments“) zu beseitigen, die den Entwicklungsprozess beeinträchtigen könnten. Der Scrum Master moderiert Meetings und fördert die kontinuierliche Verbesserung des Teams. Seine Rolle ist es, das Team zu schützen und die Effizienz der Zusammenarbeit zu steigern.
3. **Entwicklungsteam:** Das Entwicklungsteam ist eine selbstorganisierte Gruppe von Fachleuten, die in jedem Sprint ein **fertiges, funktionsfähiges Produktinkrement** liefern. Sie entscheiden selbst, wie die Arbeit erledigt wird, und tragen die Verantwortung dafür, die Anforderungen des Product Owners umzusetzen. Das Team arbeitet eng zusammen, um die Sprint-Ziele zu erreichen und die geforderte Qualität sicherzustellen.

Ablauf eines Sprints:

Ein Sprint ist ein fester Zeitraum, meist von zwei bis vier Wochen, in dem das Team eine vorab festgelegte Anzahl von Anforderungen umsetzt. Der Ablauf eines Sprints folgt einem klaren Muster:

- **Sprint Planning:** Zu Beginn jedes Sprints planen das Entwicklungsteam

und der Product Owner, welche Anforderungen aus dem Product Backlog im nächsten Sprint bearbeitet werden sollen. Dabei werden die Ziele des Sprints und die Arbeitsschritte festgelegt.

- **Daily Standups:** Diese kurzen täglichen Meetings (maximal 15 Minuten) bieten dem Team eine Gelegenheit, sich über den Fortschritt auszutauschen und mögliche Hindernisse zu identifizieren. Jedes Teammitglied gibt an, was es gestern erledigt hat, was es heute plant und ob es Probleme gibt.
- **Sprint Review:** Am Ende des Sprints findet eine Sprint Review statt, bei der das Entwicklungsteam die erledigten Aufgaben und das entstandene Produktinkrement präsentiert. Der Product Owner und andere Stakeholder geben Feedback, das in zukünftige Sprints einfließt.
- **Sprint Retrospektive:** In der Retrospektive analysiert das Team, wie der letzte Sprint ablief, und identifiziert Bereiche zur Verbesserung. Dies trägt zur kontinuierlichen Weiterentwicklung des Teams und zur Optimierung der Arbeitsweise bei.

Vorteile von Scrum:

Scrum bietet mehrere Vorteile, die besonders bei der Entwicklung komplexer Produkte und in dynamischen Umfeldern geschätzt werden:

- **Strukturierte und wiederholbare Abläufe:** Durch die festen Rollen und klaren Prozesse hat das Team eine hohe Orientierung, und regelmäßige Meetings fördern die Transparenz und Abstimmung.
- **Fokus auf kontinuierliche Verbesserung:** Die Retrospektiven am Ende jedes Sprints ermöglichen es, den Arbeitsprozess zu reflektieren und kontinuierlich zu optimieren.
- **Klare Aufgabenverteilung und Verantwortlichkeiten:** Durch die Aufteilung der Rollen und regelmäßige Kommunikation wissen alle Teammitglieder, was von ihnen erwartet wird und wie sie zum Erfolg des Projekts beitragen können.

Nachteile von Scrum:

Trotz der zahlreichen Vorteile gibt es auch einige Herausforderungen und mögliche Nachteile:

- **Starrheit bei kleinen Projekten:** Die Struktur und Disziplin, die Scrum erfordert, kann für sehr kleine Teams oder Projekte als unnötig komplex und schwerfällig empfunden werden.
- **Hohe Disziplin erforderlich:** Scrum verlangt, dass sich alle Teammitglieder aktiv an den Prozess halten und die Meetings regelmäßig besuchen. Ein Mangel an Disziplin oder Engagement kann die Effizienz des Frameworks beeinträchtigen und zu Missverständnissen führen.

Zusammenfassend lässt sich sagen, dass Scrum eine bewährte Methode ist, um komplexe Projekte agil und strukturiert zu organisieren. Die klare Rollenverteilung, die regelmäßigen Meetings und die kontinuierliche Verbesserung machen Scrum zu einem wertvollen Werkzeug, das jedoch Disziplin und Engagement vom gesamten Team verlangt.

[Vertiefung zu Scrum →](#)

5. Kanban

Ziele:

- Visualisierung des Workflows
- Frühzeitiges Erkennen von Engpässen

Grundprinzipien:

1. **Visualisierung des Arbeitsflusses:** Aufgaben werden auf einem Kanban-Board dargestellt (z. B. "To Do", "In Progress", "Done").
2. **Begrenzung laufender Arbeiten (WIP):** Beschränkung der parallelen Aufgaben, um Überlastung zu vermeiden.
3. **Kontinuierlicher Fluss:** Flexible, fortlaufende Bearbeitung von Aufgaben ohne feste Iterationen.

Vorteile:

- Transparenz und Übersicht durch das visuelle Board
- Flexibel anpassbar ohne feste Zeitvorgaben
- Stetiger Workflow und hohe Anpassungsfähigkeit

Nachteile:

- Risiko von Chaos ohne feste Struktur
- Hohe Eigenverantwortung im Team erforderlich

Kanban ist ein weiteres agiles Framework, das sich besonders durch seine Methode der **visuellen Prozesssteuerung** auszeichnet. Im Kern von Kanban steht die Visualisierung des **Arbeitsablaufs**, bei dem alle Aufgaben und Arbeitsschritte klar und übersichtlich dargestellt werden. Auf diese Weise können Engpässe, die den Arbeitsfluss behindern, frühzeitig identifiziert und behoben werden. Durch den Fokus auf kontinuierliche Verbesserung eignet sich Kanban insbesondere für Projekte, die flexible, fließende Abläufe benötigen.

Die Grundprinzipien von Kanban umfassen:

1. **Visualisierung des Arbeitsflusses:** Die Arbeitsprozesse werden auf einem sogenannten Kanban-Board dargestellt. Diese visuelle Darstellung teilt die Aufgaben typischerweise in Spalten wie "Zu erledigen" (To Do), "In Bearbeitung" (In Progress) und "Erledigt" (Done)

ein. Durch diese Transparenz können alle Teammitglieder den aktuellen Status jeder Aufgabe leicht nachvollziehen und identifizieren, wo Aufgaben feststecken könnten.

2. **Begrenzung der laufenden Arbeiten (WIP):** Kanban setzt bewusst eine Obergrenze für die Anzahl an Aufgaben, die gleichzeitig bearbeitet werden dürfen. Dies wird als WIP (Work in Progress) bezeichnet. Durch diese Begrenzung wird vermieden, dass das Team überlastet wird oder dass Engpässe im Arbeitsfluss entstehen, die das Voranschreiten des gesamten Projekts behindern könnten.
3. **Fokus auf den kontinuierlichen Fluss:** Im Gegensatz zu anderen agilen Methoden, die in Iterationen oder Sprints organisiert sind, fördert Kanban einen kontinuierlichen Arbeitsfluss. Aufgaben werden stets unmittelbar nach Beendigung der vorherigen in Angriff genommen, wodurch ein gleichmäßiger, ununterbrochener Fortschritt ermöglicht wird. Dies schafft Flexibilität und stellt sicher, dass der Workflow dynamisch und anpassungsfähig bleibt.

Vorteile von Kanban:

- **Transparenz und Übersichtlichkeit:** Das visuelle Kanban-Board bietet eine klare Übersicht über alle laufenden, geplanten und abgeschlossenen Aufgaben. Dies verbessert das Verständnis und die Kommunikation innerhalb des Teams.
- **Einfache Anpassung an den aktuellen Bedarf:** Da Kanban keine festgelegten Iterationen erfordert, kann das Team den Arbeitsablauf flexibel an sich ändernde Anforderungen und Prioritäten anpassen, ohne den gesamten Prozess umstellen zu müssen.
- **Kontinuierlicher Workflow und hohe Flexibilität:** Kanban fördert einen kontinuierlichen Arbeitsfluss, der es dem Team ermöglicht, Aufgaben zu priorisieren und sofort nach Abschluss einer anderen Aufgabe in Angriff zu nehmen.

Nachteile von Kanban:

- **Potenzielle Gefahr von Chaos ohne feste Struktur oder Disziplin:** Da Kanban sehr flexibel ist und keine festen Iterationen vorschreibt, kann es in einem unorganisierten Team oder ohne klare Rollen und Verantwortlichkeiten schnell chaotisch werden.

Die Methoden & Cynefin

Wasserfall (Einfacher Kontext)

- **Geeignet für:** Klare, stabile Projekte
- **Vorgehensweise:** Lineare Abfolge, vorhersehbare Anforderungen

Lean (Komplizierter Kontext)

- **Geeignet für:** Effiziente Projekte mit stabilen Anforderungen
- **Vorgehensweise:** Eliminierung von Verschwendungen, Optimierung durch Expertenwissen

Scrum (Komplexer Kontext)

- **Geeignet für:** Projekte mit hoher Unsicherheit und sich ändernden Anforderungen
- **Vorgehensweise:** Iterative Entwicklung in Sprints, schnelles Feedback

Kanban (Komplizierter bis komplexer Kontext)

- **Geeignet für:** Flexiblen Arbeitsfluss, sowohl für stabile als auch dynamische Aufgaben
- **Vorgehensweise:** Visuelle Steuerung, kontinuierlicher Fluss statt fester Iterationen

Agile Modelle allgemein (Komplexer Kontext)

- **Geeignet für:** Projekte mit unklaren und dynamischen Anforderungen
- **Vorgehensweise:** Flexibilität, kundenorientierte Anpassungen, iterative Verbesserung

Hier ist eine Zusammenfassung der Modelle und ihre Einordnung im Cynefin-Modell:

1. Wasserfallmodell (Einfacher Kontext)

- **Beschreibung:** Das Wasserfallmodell eignet sich für **einfache und vorhersehbare** Projekte, bei denen klare und stabile Anforderungen existieren und die Abfolge der Schritte gut bekannt ist.
- **Cynefin-Einordnung:** Im **einfachen Bereich** angewandt, bei dem Ursache-Wirkungs-Zusammenhänge eindeutig und beständig sind, und das Management durch Best Practices erfolgt.

2. Lean (Komplizierter Kontext)

- **Beschreibung:** Lean-Methoden werden oft in Projekten verwendet, die **effizient und mit hohem Wert für den Kunden** umgesetzt werden müssen. Diese Projekte benötigen oft Analyse und Erfahrung, um den besten Weg zu finden, aber die Anforderungen sind relativ stabil.
- **Cynefin-Einordnung:** Im **komplizierten Bereich**, wo Ursache-Wirkungs-Zusammenhänge verständlich sind, aber Expertenwissen erfordern. Lean ermöglicht das Identifizieren und Eliminieren von Verschwendungen durch bewährte Methoden und Optimierung.

3. Scrum (Komplexer Kontext)

- **Beschreibung:** Scrum eignet sich für Projekte mit **hoher Unsicherheit und ständigen Veränderungen**, bei denen Feedback wichtig ist. Hier werden durch iterative Zyklen (Sprints) regelmäßig Anpassungen vorgenommen, um auf sich entwickelnde Anforderungen und neue Erkenntnisse zu reagieren.
- **Cynefin-Einordnung:** Im **komplexen Bereich**, wo Ursache-Wirkungs-Beziehungen erst im Nachhinein sichtbar sind. Hier fördert Scrum Experimente und iterative Entwicklung, um wertvolle Erkenntnisse zu gewinnen.

4. Kanban (Komplizierter bis komplexer Kontext)

- **Beschreibung:** Kanban wird eingesetzt, um den **Arbeitsfluss visuell darzustellen und flexibel** an Veränderungen anzupassen. Es eignet sich sowohl für stabilere Aufgaben (kompliziert) als auch für dynamische Umgebungen (komplex).
- **Cynefin-Einordnung:** Primär im **komplizierten Bereich**, kann aber auch im komplexen Bereich effektiv sein, da es auf visuelle Steuerung und ständigen Fluss anstatt auf feste Iterationen setzt.

5. Agile Modelle allgemein (Komplexer Kontext)

- **Beschreibung:** Agile Methoden sind für **Projekte mit hoher Unsicherheit und sich schnell ändernden Anforderungen** geeignet, da sie auf Flexibilität und Feedback setzen. Dies schließt Scrum, Kanban und ähnliche Methoden ein, die iteratives Arbeiten und eine kundenorientierte Anpassung fördern.
- **Cynefin-Einordnung:** Im **komplexen Bereich**, da agile Methoden es ermöglichen, neue Ideen und Lösungsansätze zu entwickeln und anzupassen, wenn die Ursache-Wirkung-Zusammenhänge schwer vorhersagbar sind.

Zusammenfassung:

Das Cynefin-Modell zeigt, dass die Wahl des Vorgehensmodells vom Kontext und der Komplexität des Projekts abhängt:

- **Einfach:** Wasserfall – ideal für klare und stabile Projekte.
 - **Kompliziert:** Lean und Kanban – effizient für Projekte mit klaren, aber nicht einfachen Anforderungen.
 - **Komplex:** Scrum und agile Methoden – optimal für Projekte, bei denen Anforderungen unklar sind und Flexibilität erforderlich ist.
- Durch die Verwendung des Cynefin-Modells kann das geeignete

Vorgehensmodell auf Grundlage des Projektkontexts gewählt werden, um besser auf Komplexität und Unsicherheit zu reagieren.

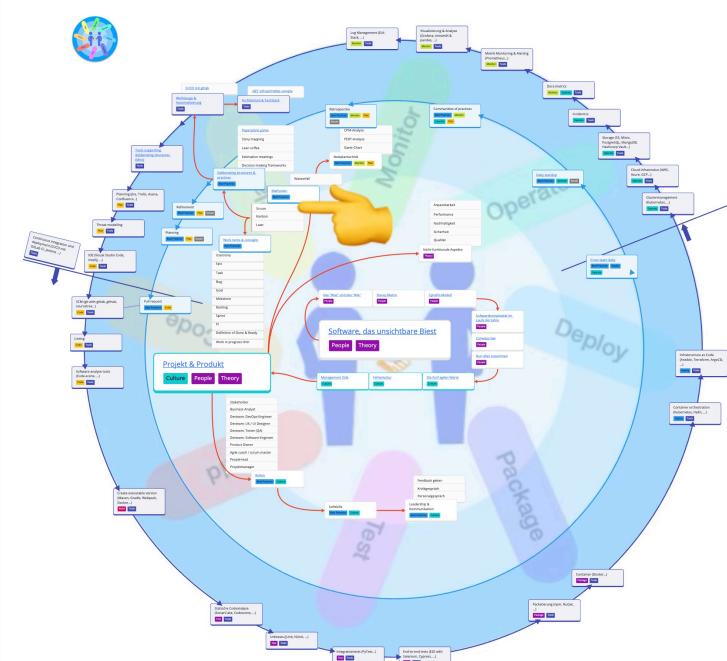
Scrum.de

Zusammenfassung Theorie / Agile Werte / Fehlerkultur / Management Stile / Produkt & Projekt / Methoden

Scrum



The Scrum Guide - The Definitive Guide to Scrum
Scrum.org - Improving Professional Scrum
Scrum Alliance - Transforming the World of Work
Atlassian: What is Scrum?
Mountain Goat Software: Scrum Framework
Visual Paradigm: What is Scrum?
Agile Alliance: Scrum
Scrum Inc.: What is Scrum?



Scrum ist ein agiles Framework, das Teams hilft, komplexe Probleme zu lösen und in kurzen Iterationen (Sprints) funktionsfähige Produkte zu liefern. Es basiert auf klar definierten Rollen, Ereignissen und Artefakten. Hier ist ein Überblick, wie Scrum abläuft, welche Rolle wann was macht und wie Meetings organisiert sind.

Rollen in Scrum

1. Product Owner (PO)

- **Verantwortung:** Maximierung des Produktwerts.
- **Aufgaben:**
 - Priorisierung des **Product Backlogs**.
 - Klärung von Anforderungen.

- Stakeholder-Management.
- **Beispiele:**
 - Entscheidung, ob eine neue Funktion wichtiger ist als technische Schulden.
 - Definition der Akzeptanzkriterien für User Stories.

2. Scrum Master (SM)

- **Verantwortung:** Sicherstellen, dass Scrum korrekt umgesetzt wird.
- **Aufgaben:**
 - Moderation der Meetings.
 - Beseitigung von Hindernissen (Impediments).
 - Coaching des Teams.
- **Beispiele:**
 - Klären von Blockaden wie fehlendem Zugang zu einem Testsystem.
 - Helfen, Konflikte im Team zu moderieren.

3. Entwicklungsteam (DEV)

- **Verantwortung:** Entwicklung eines funktionsfähigen Produkts.
- **Aufgaben:**
 - Planung und Umsetzung von Inkrementen.
 - Selbstorganisation.
- **Beispiele:**
 - Implementierung eines neuen Features.
 - Schreiben von Unit-Tests.

Ablauf eines Sprints

1. Sprint Planning (Start jedes Sprints)

- **Ziel:** Festlegen, was im Sprint erreicht wird und wie.
- **Dauer:** Maximal 8 Stunden für einen 4-Wochen-Sprint.
- **Teilnehmer:** PO, SM, DEV.
- **Agenda:**
 - **PO:** Stellt die priorisierten Backlog-Einträge vor.
 - **DEV:** Diskutiert und schätzt den Aufwand. Erarbeitet den Sprint Backlog.
 - **Beispiel:**
 - PO stellt eine Story „Benutzer können Produkte in den Warenkorb legen“ vor.
 - DEV schätzt 13 Story Points und plant, welche Tasks nötig sind.

2. Daily Scrum (täglich während des Sprints)

- **Ziel:** Synchronisation des Teams.
- **Dauer:** Maximal 15 Minuten.
- **Teilnehmer:** DEV (SM moderiert oft).

- **Agenda:**

- Jeder beantwortet:

- A. Was habe ich gestern getan?
 - B. Was werde ich heute tun?
 - C. Gibt es Hindernisse?

- **Beispiel:**

- Entwickler A: „Gestern habe ich die Login-Logik fertiggestellt.“
 - Entwickler B: „Ich brauche Klärung zu API-Spezifikationen.“

3. Sprint Review (Ende des Sprints)

- **Ziel:** Präsentation und Feedback für das Inkrement.

- **Dauer:** Maximal 4 Stunden für einen 4-Wochen-Sprint.

- **Teilnehmer:** PO, SM, DEV, Stakeholder.

- **Agenda:**

- **DEV:** Demonstriert fertige Funktionalität.

- **Stakeholder:** Geben Feedback.

- **PO:** Entscheidet über Anpassungen im Backlog.

- **Beispiel:**

- Entwickler zeigt die „Warenkorb“-Funktion live.

- Stakeholder schlägt vor: „Können wir die Designfarben ändern?“

4. Sprint Retrospective (Ende des Sprints)

- **Ziel:** Kontinuierliche Verbesserung des Prozesses.

- **Dauer:** Maximal 3 Stunden für einen 4-Wochen-Sprint.

- **Teilnehmer:** PO (optional), SM, DEV.

- **Agenda:**

- **SM** moderiert:

- Was lief gut?

- Was kann verbessert werden?

- Was nehmen wir uns für den nächsten Sprint vor?

- **Beispiel:**

- Team merkt, dass zu viele Tasks parallel bearbeitet wurden und beschließt, dies zu ändern.

Scrum Meetings und Governance-Themen

Wo sollten Governance-Themen wie Sprint-Länge besprochen werden?

1. Initiale Governance-Entscheidungen

- Sprint-Länge (z. B. 2 Wochen vs. 4 Wochen), Definition of Done, und andere grundlegende Entscheidungen sollten vor der ersten Iteration in einem **Kickoff-Meeting** festgelegt werden.

2. Laufende Anpassung

- **Retrospective:**
 - Hier wird die Sprint-Länge überprüft, wenn das Team Probleme sieht (z. B. zu kurze Sprints für komplexe Aufgaben).
 - **Beispiel:** Das Team bemerkt, dass es immer wieder unvollständige Stories gibt. Sie diskutieren, ob eine Sprint-Länge von 3 Wochen sinnvoller wäre.

Wie man andere Governance-Themen einbindet:

- **Definition of Ready** (Wann ist eine Story bereit für das Team?):
 - Diskutiert während des **Sprint Planning**.
 - Beispiel: „Stories müssen Akzeptanzkriterien enthalten und vom PO priorisiert sein.“
- **Work in Progress (WIP)-Limits:**
 - Kann in der **Retrospective** angepasst werden.
 - Beispiel: „Wir reduzieren parallele Arbeiten auf maximal 3 Storys.“

Quiz zu Scrum

```
In [1]: import ipywidgets as widgets
from IPython.display import display, clear_output

# Fragen und Antworten
questions = [
    {
        "question": "Wann wird entschieden, welche Stories im Sprint bearbeitet werden?",
        "options": ["Im Daily Scrum", "Im Sprint Planning", "In der Sprint Retrospective"],
        "answer": "Im Sprint Planning",
    },
    {
        "question": "Welche Rolle moderiert die Retrospective?",
        "options": ["Product Owner", "Scrum Master", "Entwicklungsteam"],
        "answer": "Scrum Master",
    },
    {
        "question": "Wann sollte Feedback von Stakeholdern eingeholt werden?",
        "options": ["Im Sprint Review", "Im Sprint Planning", "Im Daily Standup"],
        "answer": "Im Sprint Review",
    },
    {
        "question": "Wer ist verantwortlich für die Priorisierung des Produkts?",
        "options": ["Scrum Master", "Product Owner", "Entwicklungsteam"],
        "answer": "Product Owner",
    },
    {
        "question": "Wann wird besprochen, was gut und schlecht gelaufen ist?",
        "options": ["Im Sprint Planning", "In der Sprint Retrospective"],
        "answer": "In der Sprint Retrospective",
    },
]
```

```

# Quiz-Funktion
def quiz_game():
    score = 0

    for idx, q in enumerate(questions):
        clear_output(wait=True)

        # Frage anzeigen
        print(f"Frage {idx + 1}: {q['question']}")

        # Optionen anzeigen
        buttons = []
        for option in q["options"]:
            button = widgets.Button(description=option)

        # Callback-Funktion
        def on_click(change, option=option):
            nonlocal score
            if option == q["answer"]:
                print("\n✓ Richtig!")
                score += 1
            else:
                print(f"\n✗ Falsch! Die richtige Antwort ist: {q['answer']}")

            # Weiter zur nächsten Frage
            input("Drücke Enter, um fortfahren...")
            quiz_game() # Startet die nächste Frage

            button.on_click(on_click)
            buttons.append(button)

        # Buttons anzeigen
        display(widgets.VBox(buttons))
        return # Pausiert, bis ein Button geklickt wird

    # Quiz beenden
    clear_output(wait=True)
    print(f"🎉 Quiz abgeschlossen! Dein Ergebnis: {score}/{len(questions)}")

# Quiz starten
quiz_game()

```

Frage 1: Wann wird entschieden, welche Stories im Sprint bearbeitet werden?

VBox(children=(Button(description='Im Daily Scrum', style=ButtonStyle()),
Button(description='Im Sprint Planni...

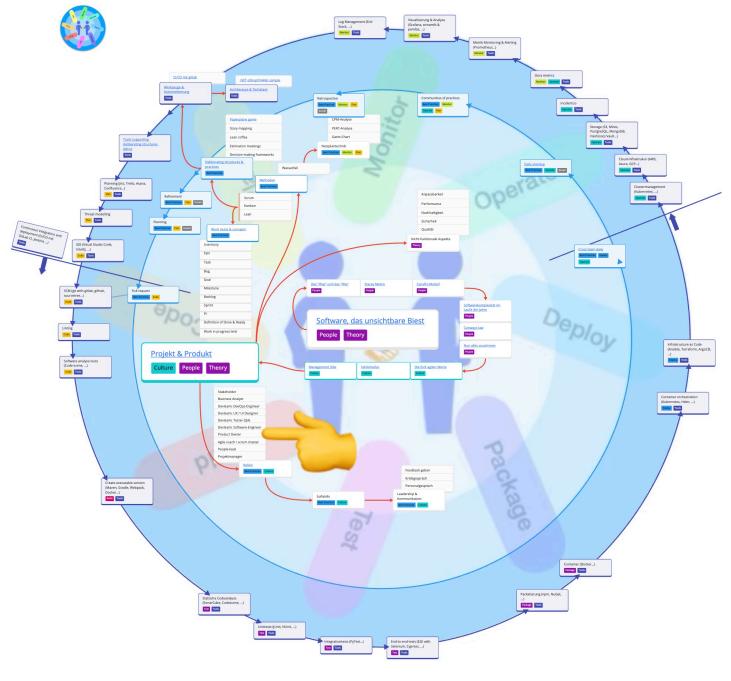
Product_Owner.de

Das unsichtbare Biest / Das "Was" und das "Wie" / Stacey Matrix / Cynefin-Modell / Softwarekomplexität seit 1980 / Conways law / Zusammenfassung Theorie / Agile Werte / Fehlerkultur / Management Stile / Produkt & Projekt / Methoden / Scrum | Rollen

Product Owner



What is a Product Owner? - Scrum.org
The Product Owner Role - Mountain Goat Software
Product Owner - Agile Alliance Glossary
What is a Product Owner in Scrum? - Visual Paradigm
Product Owner - ProductPlan Glossary
Scrum Guide - Product Owner Section



Rolle und Aufgaben des Product Owners (PO)

Verantwortung: Maximierung des Produktwerts. **Hauptaufgaben:**

- Produktvision entwickeln und kommunizieren.
- Product Backlog erstellen, pflegen und priorisieren.
- Stakeholder-Anforderungen sammeln und priorisieren.
- Entwicklungsteam Anforderungen erklären und unterstützen.
- Sprint-Ziele definieren und Fortschritt bewerten.
- Feedback von Stakeholdern und Nutzern einholen.

Auftritt in Scrum-Events

- **Sprint Planning:** Präsentiert das Product Backlog, definiert Sprint-Ziel.
- **Daily Scrum:** Selten aktiv dabei, aber verfügbar für Fragen.
- **Sprint Review:** Führt durch die Präsentation der Ergebnisse.
- **Sprint Retrospective:** Nicht aktiv, aber auf Einladung möglich.

Interaktion mit anderen Rollen

- **Entwicklungsteam:** Anforderungen klären, Blocker priorisieren, kontinuierlicher Austausch.
- **Scrum Master:** Unterstützung bei Prozessen, Schutz vor externem Druck.

- **Stakeholder:** Hauptansprechpartner, Anforderungen und Feedback priorisieren und kommunizieren.

Zusammenfassung

Der Product Owner ist der zentrale Entscheider für Produktanforderungen und sorgt dafür, dass das Scrum-Team wertorientiert arbeitet, indem er zwischen Stakeholdern, Team und Produktvision vermittelt.

Rolle und Aufgaben eines Product Owners in Scrum

Der **Product Owner** (PO) ist eine zentrale Rolle im Scrum-Framework. Er ist verantwortlich für die Maximierung des Wertes des Produkts, das durch das Scrum-Team entwickelt wird. Der PO repräsentiert die Interessen der Stakeholder und Nutzer und trägt die Verantwortung für das **Product Backlog**.

Hauptaufgaben des Product Owners:

1. **Produktvision und -strategie definieren:**

- Der PO entwickelt eine klare Produktvision, die die Richtung vorgibt und die langfristigen Ziele des Produkts unterstützt.
- Diese Vision wird mit dem Team und den Stakeholdern kommuniziert.

2. **Product Backlog verwalten:**

- **Erstellen und Pflegen:** Der PO erstellt und priorisiert das Product Backlog, eine geordnete Liste von Anforderungen, Funktionen und Verbesserungen.
- **Priorisieren:** Entscheidungen basieren auf dem geschätzten Wert für die Kunden und den geschäftlichen Nutzen.
- **Detaillierung:** Der PO stellt sicher, dass die Einträge im Backlog so detailliert sind, dass sie vom Entwicklungsteam umgesetzt werden können.

3. **Stakeholder-Management:**

- Der PO sammelt Anforderungen von Stakeholdern wie Kunden, Nutzern und Führungskräften und priorisiert sie gemäß den Zielen.
- Konflikte zwischen verschiedenen Anforderungen löst der PO, indem er auf die Produktvision verweist.

4. **Arbeiten mit dem Entwicklungsteam:**

- Der PO sorgt dafür, dass das Entwicklungsteam versteht, was geliefert werden soll und warum.
- Er ist während des Sprints verfügbar, um Fragen zu klären und Feedback zu geben.

5. **Release-Planung und Zielsetzung:**

- Der PO definiert gemeinsam mit dem Scrum-Team die Ziele für einzelne Sprints (Sprint Goals).

- Er entscheidet über den Zeitpunkt und den Inhalt von Produkt-Releases.

6. Überprüfung und Feedback:

- Der PO bewertet regelmäßig den Fortschritt und holt Feedback von Nutzern und Stakeholdern ein, um das Backlog anzupassen.

Auftreten des Product Owners in Scrum

1. Sprint Planning:

- Der PO präsentiert das priorisierte Product Backlog und erklärt dem Entwicklungsteam, welche Aufgaben den größten Wert bieten.
- Gemeinsam mit dem Team wird das Sprint-Ziel festgelegt.

2. Daily Scrum:

- Der PO nimmt in der Regel nicht aktiv am Daily Scrum teil, kann jedoch zur Klärung von Fragen bereitstehen.

3. Sprint Review:

- Der PO führt durch das Sprint Review und präsentiert Stakeholdern die gelieferten Inkremeente.
- Er sammelt Feedback, das für die Weiterentwicklung des Backlogs genutzt wird.

4. Sprint Retrospective:

- Der PO ist üblicherweise kein aktiver Teilnehmer, kann jedoch durch das Team eingeladen werden.

Interaktionen des Product Owners mit anderen Rollen

1. Entwicklungsteam:

- **Zusammenarbeit:** Der PO arbeitet eng mit dem Entwicklungsteam zusammen, um sicherzustellen, dass die Anforderungen klar verstanden und umgesetzt werden können.
- **Unterstützung:** Er beantwortet Fragen, priorisiert Blocker und nimmt Anpassungen während des Sprints vor, wenn nötig.

2. Scrum Master:

- **Unterstützung:** Der Scrum Master unterstützt den PO in seiner Rolle, indem er ihn bei der Organisation des Product Backlogs und der Kommunikation mit dem Team coacht.
- **Schutz:** Der Scrum Master sorgt dafür, dass der PO nicht von Stakeholdern unter Druck gesetzt wird, die Prozesse des Teams zu beeinflussen.

3. Stakeholder:

- **Ansprechpartner:** Der PO ist der primäre Kontaktpunkt für Stakeholder und

vermittelt deren Anforderungen und Feedback an das Team.

- **Priorisierung:** Er entscheidet, welche Anforderungen im Backlog priorisiert werden, und erklärt Stakeholdern, warum bestimmte Entscheidungen getroffen wurden.

Zusammenfassung

Der Product Owner ist der **wertmaximierende Entscheider**, der dafür sorgt, dass die Arbeit des Scrum-Teams auf die geschäftlichen und nutzerorientierten Ziele ausgerichtet ist. Seine Aufgabe besteht darin, den Bedarf der Stakeholder mit den technischen Möglichkeiten des Entwicklungsteams in Einklang zu bringen. Seine erfolgreiche Interaktion mit dem Entwicklungsteam, dem Scrum Master und den Stakeholdern ist entscheidend für den Erfolg des Produkts.

Quiz zur Rolle

```
In [1]: # Product Owner Quiz

print("Willkommen zum Product Owner Quiz!")
print("Beantworte die Fragen, indem du die entsprechende Zahl eingibst.")
print("Lass uns starten!\n")

# Fragen und Antworten
questions = [
    {
        "question": "Was ist die Hauptaufgabe eines Product Owners?",
        "options": [
            "A. Die Teammitglieder führen",
            "B. Den Wert des Produkts maximieren",
            "C. Technische Lösungen entwickeln",
            "D. Den Scrum Master unterstützen"
        ],
        "answer": "B"
    },
    {
        "question": "Welches Artefakt ist der Product Owner verantwortlich?",
        "options": [
            "A. Sprint Backlog",
            "B. Burn-Down Chart",
            "C. Product Backlog",
            "D. Release Plan"
        ],
        "answer": "C"
    },
    {
        "question": "Mit wem arbeitet der Product Owner am engsten zusammen?",
        "options": [
            "A. Nur mit dem Scrum Master",
            "B. Mit Stakeholdern und dem Entwicklungsteam",
            "C. Mit Führungskräften",
            "D. Mit dem Projektmanager"
        ],
        "answer": "B"
    }
]
```

```

},
{
    "question": "Welche Entscheidung trifft der Product Owner?", 
    "options": [
        "A. Wer im Team arbeitet",
        "B. Welche Anforderungen priorisiert werden",
        "C. Welche Programmiersprache genutzt wird",
        "D. Wie viele Sprints ein Projekt hat"
    ],
    "answer": "B"
},
{
    "question": "In welchem Scrum-Event ist der Product Owner typisch",
    "options": [
        "A. Daily Scrum",
        "B. Sprint Planning",
        "C. Sprint Retrospective",
        "D. Keine Events"
    ],
    "answer": "B"
}
]

# Quiz logik
score = 0

for i, q in enumerate(questions, 1):
    print(f"Frage {i}: {q['question']}")
    for option in q['options']:
        print(option)
    answer = input("Deine Antwort (A, B, C oder D): ").strip().upper()

    if answer == q['answer']:
        print("Richtig! ✅\n")
        score += 1
    else:
        print(f"Falsch. ❌ Die richtige Antwort ist: {q['answer']}.\n")

# Ergebnis
print(f"Quiz beendet! Du hast {score} von {len(questions)} Punkten erreic

if score == len(questions):
    print("Glückwunsch! Du bist ein Product Owner Experte! 🎉")
elif score > len(questions) // 2:
    print("Gut gemacht! Du hast ein solides Verständnis der Rolle des Pro
else:
    print("Es gibt noch etwas zu lernen. Versuch es noch einmal! 📚")

```

Willkommen zum Product Owner Quiz!
Beantworte die Fragen, indem du die entsprechende Zahl eingibst.
Lass uns starten!

Frage 1: Was ist die Hauptaufgabe eines Product Owners?

- A. Die Teammitglieder führen
- B. Den Wert des Produkts maximieren
- C. Technische Lösungen entwickeln
- D. Den Scrum Master unterstützen

Richtig! 

Frage 2: Welches Artefakt ist der Product Owner verantwortlich zu pflegen?

- A. Sprint Backlog
- B. Burn-Down Chart
- C. Product Backlog
- D. Release Plan

Falsch.  Die richtige Antwort ist: C.

Frage 3: Mit wem arbeitet der Product Owner am engsten zusammen?

- A. Nur mit dem Scrum Master
- B. Mit Stakeholdern und dem Entwicklungsteam
- C. Mit Führungskräften
- D. Mit dem Projektmanager

Richtig! 

Frage 4: Welche Entscheidung trifft der Product Owner?

- A. Wer im Team arbeitet
- B. Welche Anforderungen priorisiert werden
- C. Welche Programmiersprache genutzt wird
- D. Wie viele Sprints ein Projekt hat

Richtig! 

Frage 5: In welchem Scrum-Event ist der Product Owner typischerweise aktiv?

- A. Daily Scrum
- B. Sprint Planning
- C. Sprint Retrospective
- D. Keine Events

Richtig! 

Quiz beendet! Du hast 4 von 5 Punkten erreicht.

Gut gemacht! Du hast ein solides Verständnis der Rolle des Product Owners.



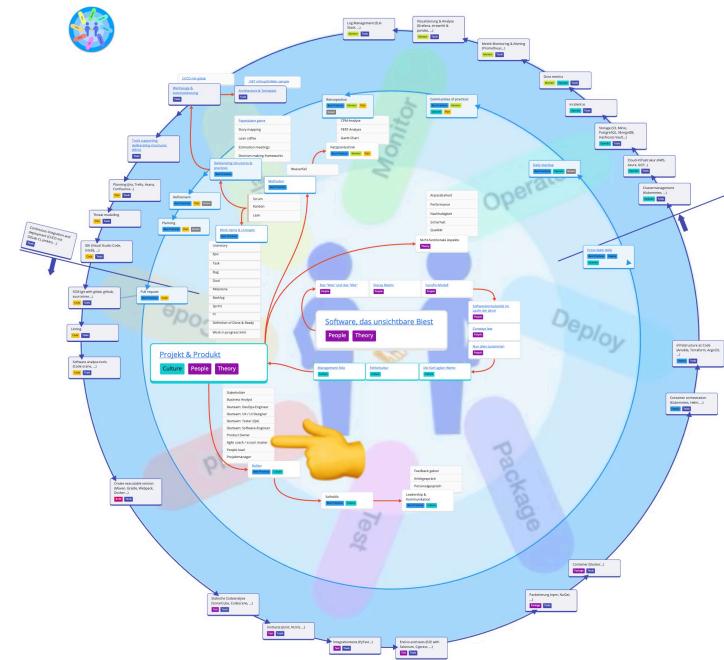
Scrum_Master.de

[Das unsichtbare Biest](#) / [Das "Was" und das "Wie"](#) / [Stacey Matrix](#) / [Cynefin-Modell](#) / [Softwarekomplexität seit 1980](#) / [Conways law](#) / [Zusammenfassung Theorie](#) / [Agile Werte](#) / [Fehlerkultur](#) / [Management Stile](#) / [Produkt & Projekt](#) / [Methoden](#) / [Scrum](#) | [Rollen](#)

Scrum Master



[What is a Scrum Master? - Scrum.org](#)
[Scrum Master Role & Responsibilities - Atlassian](#)
[Scrum Master - Agile Alliance Glossary](#)
[What is a Scrum Master? - Visual Paradigm](#)
[Scrum Master - ProductPlan Glossary](#)
[Scrum Guide - Scrum Master Section](#)



Rolle des Scrum Masters

- Servant Leader, kein Projektmanager.
- Unterstützt das Scrum-Team bei der Anwendung von Scrum und der kontinuierlichen Verbesserung.
- Fördert die Werte und Prinzipien von Scrum.

Aufgaben des Scrum Masters

- **Coaching:** Fördert Selbstorganisation und Eigenverantwortung im Team.
- **Scrum-Prozesse sicherstellen:** Moderation und Überwachung der Scrum-Events.
- **Impediments beseitigen:** Hindernisse identifizieren und entfernen.
- **Schutz des Teams:** Abschirmung von Störungen und externem Druck.
- **Agilität fördern:** Einführung und Weiterentwicklung von agilen Denkweisen.
- **Moderation und Vermittlung:** Unterstützung bei Konflikten und Organisation von Meetings.

Erscheinungsorte in Scrum

- **Sprint Planning:** Moderation, Unterstützung bei Sprintziel und Backlog-Auswahl.
- **Daily Scrum:** Sicherstellen des Fokus und der Effizienz.
- **Sprint Review:** Moderation und Förderung der Kommunikation mit Stakeholdern.
- **Sprint Retrospective:** Moderation und Identifikation von Verbesserungen.
- **Tagesgeschäft:** Prozessverbesserungen und Hindernisbewältigung.

Interaktion mit Rollen

- **Product Owner:** Unterstützung bei der Backlog-Pflege und Vermittlung von Anforderungen.
- **Entwicklungsteam:** Förderung der Selbstorganisation, Prozessverbesserung, Moderation.
- **Organisation & Stakeholder:** Vermittler, Förderung agiler Denkweisen, Anpassung von Strukturen.

Der Scrum Master ist eine der drei Hauptrollen in Scrum (neben dem Product Owner und dem Entwicklungsteam). Er ist kein Projektmanager, sondern agiert als Servant Leader, der das Scrum-Team dabei unterstützt, Scrum erfolgreich anzuwenden und kontinuierlich besser zu werden. Der Scrum Master sorgt dafür, dass die Werte, Prinzipien und Praktiken von Scrum eingehalten werden.

Hauptaufgaben des Scrum Masters

1. Coaching und Unterstützung des Scrum-Teams:

- Hilft dem Team, Selbstorganisation und Eigenverantwortung zu entwickeln.
- Fördert kontinuierliche Verbesserung durch Retrospektiven und andere Scrum-Ereignisse.

2. Scrum-Prozesse sicherstellen:

- Achtet darauf, dass alle Scrum-Events (Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective) stattfinden und effektiv durchgeführt werden.
- Überwacht die Einhaltung der Timeboxen.

3. Beseitigung von Hindernissen (Impediments):

- Unterstützt das Team dabei, Blocker zu identifizieren und zu beseitigen.
- Arbeitet mit anderen Organisationseinheiten zusammen, um die Arbeitsumgebung zu verbessern.

4. Schutz des Teams:

- Schützt das Team vor externen Einflüssen, die die Arbeit behindern könnten.
- Sorgt dafür, dass das Team ungestört am Sprintziel arbeiten kann.

5. Förderung von Agilität:

- Unterstützt die Organisation bei der Einführung und Weiterentwicklung von Scrum.
- Fördert eine agile Denkweise und Zusammenarbeit.

6. Zusammenarbeit mit dem Product Owner:

- Hilft dem Product Owner, den Backlog zu pflegen und zu priorisieren.
- Sorgt dafür, dass alle Teammitglieder die Vision und Ziele des Produkts verstehen.

7. Moderation und Vermittlung:

- Moderiert Meetings und Workshops.
- Vermittelt bei Konflikten innerhalb des Teams oder zwischen Team und anderen Stakeholdern.

Erscheinungsorte des Scrum Masters in Scrum

Scrum-Events:

1. Sprint Planning:

- Moderiert das Meeting.
- Unterstützt bei der Definition des Sprintziels und der Auswahl von Backlog-Items.

2. Daily Scrum:

- Stellt sicher, dass das Meeting effizient abläuft.
- Fördert den Fokus auf das Sprintziel und die Selbstorganisation.

3. Sprint Review:

- Unterstützt bei der Vorbereitung und Durchführung.
- Moderiert die Diskussion zwischen dem Team und Stakeholdern.

4. Sprint Retrospective:

- Moderiert die Retrospektive.
- Fördert die Offenheit und Ehrlichkeit des Teams, um Verbesserungen zu identifizieren.

Im Tagesgeschäft:

- Unterstützt das Team dabei, ihre Arbeitsprozesse zu verbessern.
- Fördert Transparenz und Kommunikation im Team.
- Arbeitet mit dem Product Owner zusammen, um den Backlog und die Anforderungen klar zu halten.

Interaktion mit anderen Rollen

1. Product Owner:

- Unterstützt den Product Owner bei der Vermittlung der Produktvision an das Team.
- Hilft, den Backlog effizient zu gestalten.
- Vermittelt zwischen Product Owner und Team bei Unklarheiten oder Missverständnissen.

2. Entwicklungsteam:

- Unterstützt das Team bei der Selbstorganisation und Prozessverbesserung.
- Moderiert Meetings und sorgt für eine produktive Arbeitsatmosphäre.

- Entfernt Hindernisse, die das Team bei der Erreichung des Sprintziels behindern.

3. Organisation und Stakeholder:

- Vermittelt zwischen dem Team und externen Stakeholdern.
- Fördert eine agile Denkweise innerhalb der Organisation.
- Unterstützt die Organisation dabei, ihre Prozesse und Strukturen so anzupassen, dass das Scrum-Team effizient arbeiten kann.

Zusammenfassung

Der Scrum Master ist der Hüter des Scrum-Prozesses, Coach und Unterstützer des Teams und Vermittler zwischen den Rollen. Er sorgt dafür, dass Scrum nicht nur formal, sondern auch in seiner Denkweise umgesetzt wird. Durch seine Arbeit stärkt er die Teamleistung und den Erfolg der Produktentwicklung.

Quiz zur Rolle

```
In [1]: # Quiz: Was macht ein Scrum Master?

print("Willkommen zum Scrum Master Quiz! Beantworte die Fragen mit a, b,
print("-----")
score = 0

# Frage 1
print("\nFrage 1: Welche der folgenden Aussagen beschreibt am besten die
print("a) Der Scrum Master ist der Chef des Teams und entscheidet, wer wa
print("b) Der Scrum Master unterstützt das Team bei der Einhaltung von Sc
print("c) Der Scrum Master ist verantwortlich für die Produktentwicklung.
print("d) Der Scrum Master führt die Organisation strategisch.")

answer1 = input("Deine Antwort: ").lower()
if answer1 == "b":
    print("Richtig!")
    score += 1
else:
    print("Falsch. Die richtige Antwort ist b: Der Scrum Master unterstüt

# Frage 2
print("\nFrage 2: Was ist eine der Hauptaufgaben des Scrum Masters?")
print("a) Produkt-Backlog pflegen")
print("b) Hindernisse beseitigen, die das Team blockieren")
print("c) Entscheidungen über Produktfeatures treffen")
print("d) Code für das Team schreiben")

answer2 = input("Deine Antwort: ").lower()
if answer2 == "b":
    print("Richtig!")
    score += 1
else:
    print("Falsch. Die richtige Antwort ist b: Der Scrum Master beseitigt
```

```

# Frage 3
print("\nFrage 3: Wer arbeitet eng mit dem Scrum Master zusammen?")
print("a) Das Entwicklungsteam und der Product Owner")
print("b) Nur der Product Owner")
print("c) Externe Stakeholder")
print("d) Der CEO der Organisation")

answer3 = input("Deine Antwort: ").lower()
if answer3 == "a":
    print("Richtig!")
    score += 1
else:
    print("Falsch. Die richtige Antwort ist a: Der Scrum Master arbeitet eng mit dem Entwicklungsteam und dem Product Owner zusammen.")

# Frage 4
print("\nFrage 4: Welche Meetings moderiert der Scrum Master? (Mehrfachantworten möglich)")
print("a) Sprint Planning")
print("b) Daily Scrum")
print("c) Sprint Review")
print("d) Sprint Retrospective")

answer4 = input("Deine Antwort: ").lower().replace(" ", "").split(",")
correct_answers = {"a", "b", "c", "d"}
if set(answer4) == correct_answers:
    print("Richtig!")
    score += 1
else:
    print("Falsch. Die richtige Antwort ist: a, b, c, d. Der Scrum Master moderiert die Daily Scrum, die Sprint Review und die Sprint Retrospective.")

# Frage 5
print("\nFrage 5: Was ist der Hauptfokus des Scrum Masters?")
print("a) Sicherstellen, dass das Team das Sprintziel erreicht")
print("b) Prozessverbesserung und Team-Coaching")
print("c) Überwachung der täglichen Arbeit des Teams")
print("d) Entscheidung über die Roadmap des Produkts")

answer5 = input("Deine Antwort: ").lower()
if answer5 == "b":
    print("Richtig!")
    score += 1
else:
    print("Falsch. Die richtige Antwort ist b: Der Scrum Master fokussiert auf Prozessverbesserung und Team-Coaching.")

# Endergebnis
print("\n-----")
print(f"Du hast {score} von 5 Fragen richtig beantwortet!")

if score == 5:
    print("Exzellent! Du hast ein umfassendes Verständnis der Scrum Masterrolle und -funktionen.")
elif score >= 3:
    print("Gut gemacht! Du hast ein solides Wissen über den Scrum Master.")
else:
    print("Es gibt noch Raum für Verbesserung. Lies dir die Aufgaben und lerne weiter!")

```

Willkommen zum Scrum Master Quiz! Beantworte die Fragen mit a, b, c oder d.

Frage 1: Welche der folgenden Aussagen beschreibt am besten die Rolle des Scrum Masters?

- a) Der Scrum Master ist der Chef des Teams und entscheidet, wer was macht.
- b) Der Scrum Master unterstützt das Team bei der Einhaltung von Scrum-Prinzipien.

c) Der Scrum Master ist verantwortlich für die Produktentwicklung.

d) Der Scrum Master führt die Organisation strategisch.

Falsch. Die richtige Antwort ist b: Der Scrum Master unterstützt das Team bei der Einhaltung von Scrum-Prinzipien.

Frage 2: Was ist eine der Hauptaufgaben des Scrum Masters?

- a) Produkt-Backlog pflegen
- b) Hindernisse beseitigen, die das Team blockieren
- c) Entscheidungen über Produktfeatures treffen
- d) Code für das Team schreiben

Richtig!

Frage 3: Wer arbeitet eng mit dem Scrum Master zusammen?

- a) Das Entwicklungsteam und der Product Owner
- b) Nur der Product Owner
- c) Externe Stakeholder
- d) Der CEO der Organisation

Falsch. Die richtige Antwort ist a: Der Scrum Master arbeitet eng mit dem Entwicklungsteam und dem Product Owner zusammen.

Frage 4: Welche Meetings moderiert der Scrum Master? (Mehrfachantwort möglich, trenne mit Komma)

- a) Sprint Planning
- b) Daily Scrum
- c) Sprint Review
- d) Sprint Retrospective

Falsch. Die richtige Antwort ist: a, b, c, d. Der Scrum Master moderiert alle diese Meetings.

Frage 5: Was ist der Hauptfokus des Scrum Masters?

- a) Sicherstellen, dass das Team das Sprintziel erreicht
- b) Prozessverbesserung und Team-Coaching
- c) Überwachung der täglichen Arbeit des Teams
- d) Entscheidung über die Roadmap des Produkts

Falsch. Die richtige Antwort ist b: Der Scrum Master fokussiert sich auf Prozessverbesserung und Team-Coaching.

Du hast 1 von 5 Fragen richtig beantwortet!

Es gibt noch Raum für Verbesserung. Lies dir die Aufgaben und Rolle des Scrum Masters noch einmal durch!

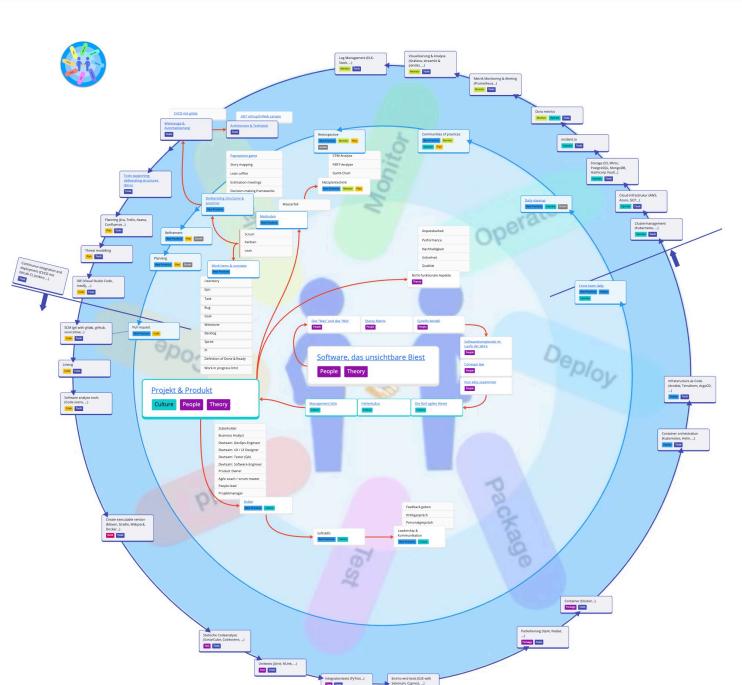
Systemic_Approach.de

[Zusammenfassung Theorie / Agile Werte / Fehlerkultur / Management Stile / Produkt & Projekt](#)

Systemischer Ansatz



The Systemic Approach: Concepts, Method and Tools - SpringerLink
A Systemic Perspective on Organizations: International Experience - SpringerLink
Systemic View and Systems Thinking - SpringerLink
Systemic Methods for Collaboration - Richard Seidl
Systemic Team Coaching Builds High Value Teams - Odgers Berndtson
The Systems Thinking Approach to Strategic Management - MDPI
The Role Of Systems Thinking In Organizational Change And Development - Forbes
Approaches to Team Coaching - Amalte
The Power of Systems Thinking: A Holistic Approach to Problem-Solving - Gestaldt
What we focus on - Systemic Coaching - artop



Systemischer Ansatz in IT-Projekten: Kurz und knapp

Grundlagen

- Betrachtet Projekte als dynamische Systeme mit Wechselwirkungen.
- Fokus auf Lösungen, Ressourcen und positive Entwicklung statt auf Probleme.

Zentrale Fragen

1. „Angenommen, unsere Arbeit läuft perfekt, was wäre dann anders?“

- Vision eines idealen Zustands schaffen.

- Fokus auf Ziele statt Probleme.
2. „**Woran würdest Du das merken?**“

- Erfolg sichtbar und messbar machen.
- Konkrete Indikatoren definieren.

3. „**Woran würden andere das merken?**“

- Auswirkungen auf das System erkennen.
- Perspektive auf Stakeholder und Team erweitern.

Nutzen in IT-Projekten

- **Lösungsfokus:** Weg von Problemen, hin zu Möglichkeiten.
- **Ziele und Kriterien:** Erfolg klar definieren, auch in unsicheren Projekten.
- **Wechselwirkungen:** Soziale und technische Dynamiken im Blick behalten.

Anwendung

- **Teambesprechungen:** Fokus auf Visionen und Fortschritt.
- **Change Management:** Ziele und Maßnahmen ableiten.
- **Agile Retrospektiven:** Reflexion und Eigenverantwortung fördern.

Fazit

- Der Erfolg eines Projekts beginnt mit den richtigen Fragen.
- Klarheit schaffen, Eigenverantwortung stärken, Lösungen fördern.

Die erfolgreiche Umsetzung von Projekten erfordert nicht nur technische Kompetenz, sondern auch ein tiefes Verständnis für die Wechselwirkungen in Teams und Organisationen. Der systemische Ansatz bietet hierfür eine wertvolle Perspektive, indem er den Fokus von Problemen auf Lösungen verschiebt. Durch gezielte Fragen lassen sich komplexe Zusammenhänge beleuchten und transformative Veränderungen anstoßen.

Der systemische Ansatz: Grundlagen

Der systemische Ansatz betrachtet Projekte, Teams und Organisationen als dynamische Systeme, die durch Beziehungen und Wechselwirkungen geprägt sind. Probleme werden nicht isoliert analysiert, sondern in ihrem Kontext verstanden. Der Ansatz fördert die Suche nach Ressourcen, Potenzialen und positiven Entwicklungsmöglichkeiten, anstatt sich auf Defizite zu konzentrieren.

Ein zentrales Werkzeug im systemischen Ansatz ist der Einsatz von lösungsorientierten Fragen. Diese Fragen helfen dabei, ein klares Bild von einem idealen Zustand zu entwickeln und konkrete Schritte zur Erreichung dieses Ziels zu definieren.

Drei zentrale Fragen des systemischen Ansatzes

1. „Angenommen, unsere gemeinsame Arbeit läuft perfekt, was wäre dann anders?“

- Diese Frage zielt darauf ab, die Vision eines idealen Projekterfolgs zu konkretisieren.
- Sie verschiebt den Fokus von aktuellen Herausforderungen hin zu den erwünschten Ergebnissen.
- Beispiel: In einem IT-Projekt könnten Antworten wie „Die Nutzer sind mit der Software zufrieden“ oder „Wir haben transparente Prozesse und klare Verantwortlichkeiten“ entstehen.

2. „Woran würdest Du das merken?“

- Diese Frage hilft, messbare oder spürbare Indikatoren für den Erfolg zu identifizieren. Sie macht den Fortschritt greifbar.
- Beispiel: „Ich bekomme weniger Rückfragen vom Team, und die Deadlines werden eingehalten.“

3. „Und woran würden andere das merken?“

- Diese Frage erweitert den Blick auf die Auswirkungen des Erfolgs auf das gesamte System, z. B. auf Stakeholder, Kunden oder andere Abteilungen.
- Beispiel: „Die Kundenfeedbacks sind positiver, und die Zusammenarbeit mit der Marketingabteilung läuft reibungslos.“

Der Nutzen des systemischen Ansatzes in IT-Projekten

1. Lösungsorientierung statt Problemfokus

- Der Ansatz ermutigt, sich auf die Lösung und nicht auf das Problem zu konzentrieren.
- In IT-Projekten bedeutet das beispielsweise, dass statt der Frage „Warum ist das Backend so langsam?“ eher gefragt wird: „Was wäre anders, wenn die Performance optimal wäre?“

2. Klarheit über Ziele und Kriterien

- Die Fragen fördern eine präzise Definition von Erfolgskriterien. Das ist besonders wichtig in Projekten, die von Unsicherheiten oder wechselnden Anforderungen geprägt sind.
- Beispiel: „Wir hätten klare API-Dokumentationen und eine stabile Infrastruktur.“

3. Fokus auf Zusammenarbeit und Wechselwirkungen

- Der systemische Ansatz sensibilisiert für die Wechselwirkungen innerhalb eines Projektes. So werden nicht nur technische Aspekte betrachtet, sondern auch die sozialen Dynamiken im Team.
- Beispiel: „Wenn die Zusammenarbeit gut läuft, reduzieren sich Eskalationen und Missverständnisse im Projektteam.“

Anwendung des systemischen Ansatzes im Projektalltag

- **Teambesprechungen:** Regelmäßige Reflexionsrunden, in denen die Fragen thematisiert werden, können helfen, die Ausrichtung des Projekts zu schärfen und Probleme frühzeitig zu erkennen.
- **Change-Management-Prozesse:** In Veränderungsprozessen können diese Fragen als Leitfaden dienen, um Visionen und Maßnahmen zu entwickeln.
- **Retrospektiven in agilen Teams:** Der Ansatz passt hervorragend in den agilen Kontext, da er die Eigenverantwortung und Reflexion fördert.

Fazit: Erfolg beginnt mit den richtigen Fragen

Der systemische Ansatz zeigt, wie kraftvoll Fragen in der Projektarbeit sein können. Sie schaffen Klarheit, fördern die Eigenverantwortung und eröffnen neue Perspektiven. Indem sie den Fokus auf Lösungen und positive Entwicklungen lenken, tragen sie dazu bei, dass Projekte nicht nur technisch, sondern auch menschlich erfolgreich sind.

Die drei zentralen Fragen – „Was wäre anders?“, „Woran würdest Du das merken?“ und „Woran würden andere das merken?“ – sind einfache, aber wirkungsvolle Werkzeuge, die sich in jedem IT-Projekt anwenden lassen. Sie helfen dabei, nicht nur die Ziele eines Projekts zu definieren, sondern auch den Weg dorthin aktiv zu gestalten.

Workitems_Concepts.de

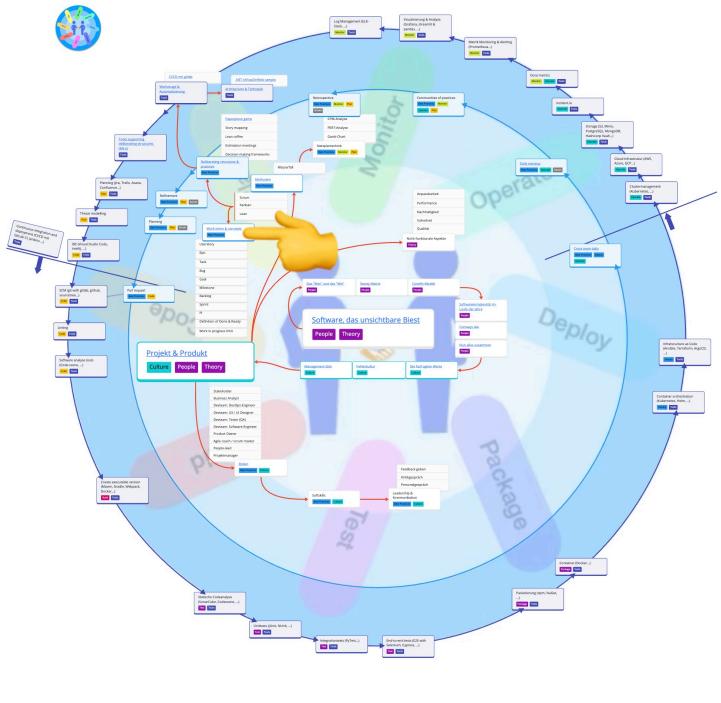
[Produkt & Projekt / Methoden / Scrum](#)

Workitems & concepts



Epics, Stories, Themes, and Initiatives | Atlassian
Understanding Epics in Agile for Managing and

Chunking Work Items
 Define features and epics to organize backlog items - Azure Boards
 Epics, Storys, Themes und Initiativen | Atlassian
 What Is An Agile Epic? Best Practices, Template & Example
 Was sind Epics, Features, User Stories und Tasks im agilen Projektmanagement?
 Epics - Atlassian
 Epics in Scrum | Agiles Lexikon - Agile Academy
 Theme, Epic, Story, Task - Scrum & Kanban
 User story - Wikipedia



Arbeitselemente und Konzepte, die in agilen Methoden, insbesondere Scrum und Kanban, verwendet werden:

1. User Story

Eine **User Story** ist eine kurze, einfache Beschreibung einer Funktionalität oder eines Features aus der Sicht des Endnutzers oder Stakeholders. Sie wird üblicherweise im Format "Als [Nutzerrolle] möchte ich [Ziel], um [Nutzen]" formuliert. User Stories dienen dazu, die Anforderungen an ein Produkt oder eine Funktion zu beschreiben, ohne sich in technische Details zu verlieren.

Beispiel:

- Als Käufer möchte ich eine Filterfunktion im Shop, um Produkte nach Kategorie und Preis zu sortieren, damit ich schneller das finde, was ich suche.

2. Epic

Ein **Epic** ist eine größere User Story, die zu umfangreich ist, um sie in einem einzigen Sprint zu bearbeiten. Es ist eine übergeordnete Funktion oder ein Themenblock, der in mehrere kleinere User Stories unterteilt werden kann. Epics helfen dabei, das Produkt in größere, zusammenhängende Bereiche zu gliedern.

Beispiel:

- Epic: Produktverwaltung
 - User Story 1: Als Admin möchte ich Produkte hinzufügen können.
 - User Story 2: Als Admin möchte ich Produkte bearbeiten können.
 - User Story 3: Als Admin möchte ich Produkte löschen können.

3. Task

Ein **Task** ist eine konkrete, spezifische Arbeitseinheit, die in einem Sprint erledigt werden muss. Tasks sind kleiner als User Stories und beinhalten normalerweise technische oder operationelle Details, die zur Umsetzung der User Story erforderlich sind. Ein Task könnte zum Beispiel das Erstellen einer bestimmten Funktion im Code oder das Testen eines Features sein.

Beispiel:

- Task: Implementierung der Filterfunktion im Frontend.

4. Bug

Ein **Bug** bezeichnet einen Fehler oder ein unerwünschtes Verhalten in der Software. Bugs können während der Entwicklung oder nach der Veröffentlichung entdeckt werden. Sie müssen dokumentiert, priorisiert und in den Arbeitsablauf integriert werden, um sicherzustellen, dass sie behoben werden.

Beispiel:

- Bug: Die Filterfunktion zeigt falsche Ergebnisse, wenn die Preisgrenze eingestellt wird.

5. Goal (Sprint Goal)

Das **Sprint Goal** ist das übergeordnete Ziel eines Sprints. Es beschreibt, was das Team im Sprint erreichen möchte und gibt eine klare Richtung für die Arbeit. Das Sprint Goal wird zu Beginn des Sprints festgelegt und hilft dem Team, sich auf die wichtigsten Aufgaben zu konzentrieren und die Prioritäten zu setzen.

Beispiel:

- Sprint Goal: Implementierung der Filterfunktionalität und Behebung von Bugs in der Produktansicht.

6. Milestone

Ein **Milestone** ist ein bedeutender Punkt oder ein Zwischenziel innerhalb eines Projekts, das oft am Ende eines bestimmten Zeitraums oder nach der Fertigstellung wichtiger Funktionen erreicht wird. Milestones können oft als Indikatoren für den Fortschritt des Projekts verwendet werden.

Beispiel:

- Milestone: Die erste Version der Produktverwaltung ist live und funktionsfähig.

7. Backlog

Das **Backlog** ist eine geordnete Liste von Aufgaben, User Stories und Epics, die das Team in kommenden Sprints bearbeiten muss. Es gibt in der Regel zwei Arten von Backlogs:

- **Product Backlog:** Enthält alle Anforderungen und Aufgaben, die das Produkt betreffen.
- **Sprint Backlog:** Enthält die Aufgaben und User Stories, die für den aktuellen Sprint ausgewählt wurden.

8. Sprint

Ein **Sprint** ist ein kurzer, festgelegter Zeitraum (typischerweise 1 bis 4 Wochen), in dem das Team eine bestimmte Menge an Arbeit erledigt. Am Ende eines Sprints gibt es eine Review, bei der das Team das erreichte Ziel und die erledigten Aufgaben präsentiert. Sprints helfen, Arbeit in überschaubaren Zeiträumen zu strukturieren und das Produkt schrittweise zu verbessern.

Beispiel:

- Sprint 1: Implementierung der Produktfilter und Tests.

9. Product Increment (PI)

Ein **Product Increment (PI)** ist das Resultat eines Sprints oder eines aggregierten Sprints, das eine funktionierende Version des Produkts darstellt, die nach den Anforderungen und den Zielen des Teams weiterentwickelt wurde. Ein PI sollte potentiell auslieferbar und getestet sein, auch wenn nicht alle Funktionen bereits live sind.

Beispiel:

- PI: Ein funktionierendes Produkt mit der Filterfunktionalität und einigen Bugfixes.

10. Definition of Done (DoD)

Die **Definition of Done** ist eine Vereinbarung, die beschreibt, was es bedeutet, dass eine User Story oder ein Task als "fertig" betrachtet wird. Sie stellt sicher, dass alle erforderlichen Arbeiten erledigt und getestet wurden und dass keine weiteren Aufgaben mehr notwendig sind, um das Feature zu liefern.

Beispiel:

- Eine User Story gilt als "done", wenn:
 - Der Code geschrieben und getestet wurde.
 - Die Dokumentation aktualisiert wurde.
 - Das Feature auf verschiedenen Geräten getestet wurde.
 - Der Code überprüft und im Code-Repository integriert wurde.

11. Definition of Ready (DoR)

Die **Definition of Ready** beschreibt die Kriterien, die eine User Story oder ein Task erfüllen muss, bevor sie in einen Sprint aufgenommen werden kann. Sie stellt sicher, dass das Team alle notwendigen Informationen hat, um die Arbeit zu beginnen.

Beispiel:

- Eine User Story gilt als "ready", wenn:
 - Sie gut beschrieben und verständlich ist.
 - Akzeptanzkriterien definiert sind.
 - Alle notwendigen Abhängigkeiten geklärt sind.
 - Sie in den Sprint aufgenommen werden kann.

12. Work in Progress Limit (WIP Limit)

Ein **Work in Progress Limit** (WIP Limit) ist eine Grenze, die die Anzahl der Aufgaben oder User Stories begrenzt, die ein Team gleichzeitig bearbeiten kann. Das Ziel eines WIP-Limits ist es, die Effizienz zu steigern, indem verhindert wird, dass zu viele Aufgaben gleichzeitig in Bearbeitung sind, was zu einer Überlastung und ineffizienter Arbeit führen könnte.

Beispiel:

- WIP Limit: Maximal 3 Tasks gleichzeitig in der Implementierung.

Diese Konzepte sind ein integraler Bestandteil der agilen Methodiken und helfen Teams, den Arbeitsfluss zu optimieren, die Kommunikation zu verbessern und die Qualität der Arbeit zu sichern. Sie stellen sicher, dass das Team fokussiert bleibt und die Arbeiten iterativ und in kleinen, gut definierten Schritten abgeschlossen werden.

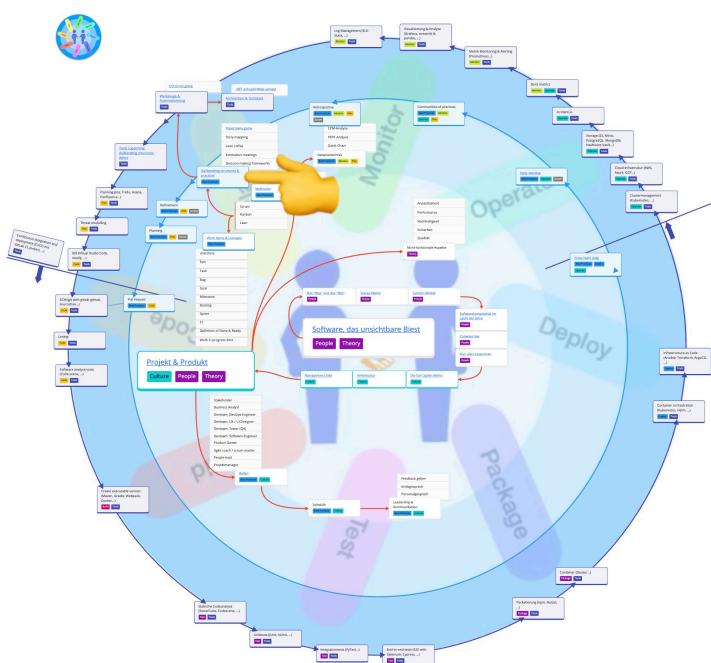
Deliberating_Structures.de

[Produkt & Projekt / Methoden](#)

Deliberating Structures & Practices



- Liberating Structures – Official Website
- Liberating Structures – Innovation durch echte Zusammenarbeit
- Was sind Liberating Structures? (+ Beispiele) – Agile Scrum Group
- Mit Liberating Structures gemeinsam mehr erreichen: Die Übersetzung des ...
- Eintauchen in die Liberating Structures - Facilitation Academy
- Liberating Structures anwenden – mit Vorlagen – business-wissen.de
- Liberating Structures: Innovative Methoden für effektive Meetings
- Liberating Structures Europe
- Liberating Structures | Projekt Agility Lab | Universität Stuttgart
- Alle Liberating Structures alphabetisch sortiert



Formale/informelle Strukturen zur Entscheidungsfindung.

Merkmale von Deliberating Structures: Formalisierte Abläufe, Beteiligung aller Parteien, Transparenz, zielorientierte Diskussionen.

Beispiele im agilen Umfeld:

- **Daily Stand-up:** Tägliches Meeting zur Statusüberprüfung und Problemlösung.
→
 - **Sprint Retrospective:** Reflexion und Verbesserung des Arbeitsprozesses.
 - **Sprint Planning:** Planung der nächsten Aufgaben im Sprint.
 - **Backlog Refinement:** Priorisierung und Klärung von Anforderungen.
 - **Story Mapping Workshops:** Visualisierung der User Journey und Priorisierung von Features.
 - **Communities of Practice:** Fachgruppen zur Wissens- und Problemlösung.
 - **PI Planning (SAFe):** Gemeinsame Planung mehrerer Teams.
 - **Lean Coffee:** Themenbasierte Diskussion ohne feste Agenda.
 - **Estimation Meetings:** Aufwandsschätzungen durch Planning Poker oder T-Shirt Sizing.

- **Decision-making Frameworks (z. B. Fist of Five):** Schnelle Entscheidungsfindung durch Abstimmungsmechanismen.

Beschreibung

Deliberating Structures, auch als „beratende Strukturen“ oder „strukturierte Entscheidungsfindung“ bezeichnet, beziehen sich auf formelle oder informelle Gruppen- oder Organisationsstrukturen, die dazu dienen, Diskussionen und Entscheidungsprozesse effizienter zu gestalten. Diese Strukturen bieten Rahmenbedingungen, um eine geregelte, offene und durchdachte Debatte zu ermöglichen, die zu konsensbasierten Entscheidungen führt.

Typische Merkmale von Deliberating Structures sind:

1. **Formalisierte Abläufe:** Es gibt klare Regeln oder Prozeduren, wie Diskussionen ablaufen, wer spricht, und wie Entscheidungen getroffen werden.
2. **Beteiligung aller Parteien:** Deliberierende Strukturen ermutigen die Teilnahme aller relevanten Akteure oder Interessengruppen, um sicherzustellen, dass alle Perspektiven berücksichtigt werden.
3. **Transparenz und Rechenschaftspflicht:** Entscheidungen werden auf eine Art und Weise getroffen, die nachvollziehbar und für die Beteiligten verständlich ist, was die Rechenschaftspflicht stärkt.
4. **Zielorientierte Diskussionen:** Der Fokus liegt darauf, fundierte, durchdachte und häufig konsensbasierte Entscheidungen zu treffen, die das beste Ergebnis für die gesamte Gruppe oder Organisation darstellen.

Diese Strukturen werden oft in verschiedenen Kontexten eingesetzt, z.B. in der Politik, in Unternehmen oder in gemeinnützigen Organisationen, um komplexe Entscheidungsfindungsprozesse zu steuern. Ein Beispiel wäre ein Rat oder eine Kommission, die über wichtige Fragen debattiert und Lösungen auf der Basis von Diskussion und Konsens entwickelt.

Deliberating structures im agilen Umfeld

Im agilen Umfeld gibt es verschiedene deliberating structures, die speziell dazu dienen, Teams bei der Entscheidungsfindung, Problemlösung und kontinuierlichen Verbesserung zu unterstützen. Hier sind einige Beispiele:

1. Daily Stand-up (Daily Scrum)

- **Zweck:** Ein kurzes tägliches Treffen, bei dem jedes Teammitglied berichtet, was es am vorherigen Tag getan hat, was es heute tun wird und ob es auf Hindernisse gestoßen ist.

- **Deliberation:** Diese Struktur hilft dem Team, tägliche Entscheidungen darüber zu treffen, wie die Arbeit voranschreitet, Hindernisse zu identifizieren und gegebenenfalls gemeinsam Lösungen zu finden.

2. Sprint Retrospective

- **Zweck:** Am Ende jedes Sprints treffen sich die Teammitglieder, um den letzten Sprint zu reflektieren, herauszufinden, was gut funktioniert hat, was verbessert werden kann und konkrete Verbesserungsmaßnahmen zu erarbeiten.
- **Deliberation:** Diese Struktur fördert eine offene Diskussion darüber, wie der Prozess optimiert werden kann. Das Team debattiert die besten Verbesserungsmaßnahmen und trifft gemeinsam Entscheidungen für den nächsten Sprint.

3. Sprint Planning

- **Zweck:** Dieses Meeting findet zu Beginn jedes Sprints statt, um zu planen, welche Arbeit das Team im kommenden Sprint leisten kann.
- **Deliberation:** Hier werden die Anforderungen und User Stories durch das gesamte Team diskutiert, um sicherzustellen, dass sie verstanden und richtig priorisiert werden. Entscheidungen darüber, wie viel Arbeit in einem Sprint erledigt werden kann, werden auf der Grundlage dieser Diskussionen getroffen.

4. Backlog Refinement (Backlog Grooming)

- **Zweck:** In regelmäßigen Abständen geht das Team durch das Product Backlog, um sicherzustellen, dass es priorisiert, verfeinert und verständlich ist.
- **Deliberation:** Dies ist eine Gelegenheit für das Team, die Anforderungen und User Stories detailliert zu besprechen und gegebenenfalls Anpassungen vorzunehmen. Entscheidungen werden über die Priorität und den Umfang zukünftiger Arbeiten getroffen.

5. Story Mapping Workshops

- **Zweck:** Ein Workshop, bei dem das Team die User Journey durch eine Anwendung visualisiert und die entsprechenden Features und Stories in eine sinnvolle Reihenfolge bringt.
- **Deliberation:** In dieser Struktur wird das Team dazu angeregt, gemeinsam darüber nachzudenken, wie die Benutzererfahrung priorisiert und die wichtigsten Funktionalitäten entwickelt werden sollen. Dies hilft, die Richtung des Produkts auf der Grundlage der Debatten zu bestimmen.

6. Communities of Practice (CoP)

- **Zweck:** Dies sind freiwillige, fachlich orientierte Gruppen, die sich regelmäßig

treffen, um Wissen zu teilen, Best Practices zu diskutieren und gemeinsame Probleme zu lösen.

- **Deliberation:** Eine Community of Practice bietet einen Raum, in dem Experten in einem bestimmten Bereich zusammenkommen, um Themen zu besprechen und gemeinsam fundierte Entscheidungen über neue Techniken, Tools oder Prozesse zu treffen.

7. Product Increment (PI) Planning (bei SAFe - Scaled Agile Framework)

- **Zweck:** Ein großes Planungsmeeting, bei dem mehrere Teams, die an demselben Produkt arbeiten, gemeinsam die kommenden Arbeitszyklen planen.
- **Deliberation:** Diese deliberative Struktur stellt sicher, dass alle beteiligten Teams zusammenarbeiten, um die übergreifende Planung und Priorisierung durch Diskussion und Zusammenarbeit abzustimmen.

8. Lean Coffee

- **Zweck:** Eine Struktur, in der das Team ein Treffen organisiert, ohne eine vorgegebene Agenda, und währenddessen gemeinsam entscheidet, welche Themen priorisiert und besprochen werden.
- **Deliberation:** Die Gruppe diskutiert, welche Themen am dringendsten sind, stimmt gemeinsam ab, und dann findet eine zielgerichtete Diskussion statt, um fundierte Entscheidungen zu treffen.

9. Estimation Meetings (Planning Poker, T-Shirt Sizing)

- **Zweck:** Eine Methode, um die Komplexität und den Aufwand für User Stories zu schätzen.
- **Deliberation:** Das Team diskutiert die Anforderungen und nutzt strukturierte Abstimmungsmethoden, wie Planning Poker, um Entscheidungen über die Größe und den Aufwand einer Aufgabe zu treffen.

10. Decision-making Frameworks (z. B. Fist of Five)

- **Zweck:** Ein Tool, um schnell Entscheidungen zu treffen, bei dem Teammitglieder durch Handzeichen (1 bis 5 Finger) signalisieren, wie stark sie mit einer Entscheidung übereinstimmen.
- **Deliberation:** Diese Struktur hilft dem Team, schnell Feedback zu Entscheidungen zu sammeln und eine Debatte zu führen, falls keine ausreichende Einigkeit besteht.

Diese deliberating structures tragen dazu bei, dass im agilen Umfeld Entscheidungen durch den Input aller relevanten Teammitglieder getroffen werden, wodurch die Transparenz und Qualität der Entscheidungsfindung gefördert wird.

Deliberate practices und Phasen

Deliberating structures schaffen eine unterstützende Umgebung für Entscheidungen und Zusammenarbeit, während deliberate practices den Fokus auf die gezielte und kontinuierliche Verbesserung individueller Fähigkeiten legen.

Der Unterschied zwischen "deliberating structures" und "deliberate practices" liegt in ihrem Ansatz und ihrer Anwendung:

1. Deliberating Structures

- **Definition:** "Deliberating structures" bezieht sich auf bewusst entworfene Strukturen oder Rahmenbedingungen, die den Entscheidungsfindungsprozess unterstützen. Diese Strukturen zielen darauf ab, durch klare Rahmenbedingungen, Prozesse und Regeln, die Kommunikation und Zusammenarbeit zu lenken und zu verbessern.
- **Ziel:** Diese Strukturen fördern den Austausch von Ideen, ein strukturiertes Vorgehen und die Erarbeitung durchdachter Lösungen innerhalb einer Gruppe oder Organisation.
- **Anwendungsbeispiel:** Ein Unternehmen könnte eine spezielle Meetingstruktur einführen, bei der klare Rollen verteilt und bestimmte Methoden (z.B. „Liberating Structures“) verwendet werden, um Diskussionen zu steuern und Konsens zu erreichen. Diese Strukturen sind darauf ausgelegt, individuelle Beiträge zu bündeln und den Entscheidungsprozess zu fördern.

2. Deliberate Practices

- **Definition:** "Deliberate practices" bezieht sich auf gezielte und bewusste Übungspraktiken, die kontinuierlich darauf abzielen, spezifische Fähigkeiten zu entwickeln oder zu verbessern. Sie zeichnen sich durch Wiederholung, gezieltes Feedback und kontinuierliche Anpassungen aus, um Expertise oder Meisterschaft in einem bestimmten Bereich zu erreichen.
- **Ziel:** Der Fokus liegt auf der individuellen Verbesserung durch gezielte, herausfordernde Aufgaben, die gezielt Schwächen adressieren. Diese Praktiken sind eng mit persönlicher Entwicklung und Lernen verbunden.

1. Plan

- **Backlog Refinement:** Anforderungen verfeinern, Priorisieren, Schätzungen üben.
- **Sprint Planning:** Aufgaben auswählen, Sprintziele definieren, MVP bestimmen.

2. Code

- **Daily Stand-ups:** Fortschritt und Blocker besprechen, synchronisieren.
- **Pair Programming & Code Reviews:** Wissen teilen, Qualität sichern, Feedback üben.

3. Build

- **Continuous Integration:** Automatische Builds und Tests, Engpässe erkennen.
- **Practice:** CI-Pipeline-Optimierung und Rollback-Übungen.

4. Test

- **Sprint Reviews:** Tests zeigen, dass Anforderungen erfüllt sind.
- **Deliberate Practice:** TDD und Automatisierung zur Qualitätssteigerung.

5. Package

- **Sprint Retrospective:** Zusammenarbeit und Paketierungsprozess reflektieren.
- **Practice:** Test-Rollouts und Rollback-Übungen.

6. Deploy

- **Release Planning:** Deployment-Strategien besprechen (z. B. Blau-Grün).
- **Practice:** Chaos Engineering und Dark Launches für resilientes Deployment.

7. Operate

- **Cross Team Daily:** Plattformgesundheit und Incidents besprechen.
- **Practice:** Incident Management-Übungen, Major Incident-Simulationen.

8. Monitor

- **Retrospective Meetings:** Monitoring-Erkenntnisse analysieren und optimieren.
- **Practice:** Monitoring-Schulungen und Alarm-Management trainieren.

Kurzfassung:

Die Schritte des Entwicklungszyklus werden durch agile Meetings synchronisiert und durch deliberate practices kontinuierlich verbessert.

Hier ist eine ausführlichere Erklärung, wie die Phasen „plan, code, build, test, package, deploy, operate, monitor“ mit agilen Meetings und deliberate practices integriert werden können:

1. Plan (Planen)

▪ Backlog Refinement (Backlog-Pflege):

- In regelmäßigen Abständen trifft sich das Team, um den Produkt-Backlog zu prüfen und zu verfeinern. Anforderungen werden analysiert, priorisiert und in kleinere, umsetzbare Aufgaben aufgeteilt. Dies ist ein zentraler Schritt, um sicherzustellen, dass das Team sich auf wertvolle und umsetzbare Features konzentriert.
- **Deliberate Practice:** Teams können Übungen zum Schätzen und zur Priorisierung durchführen. Beispielsweise kann Planning Poker verwendet werden, um Story Points zu schätzen und so ein gemeinsames Verständnis über den Aufwand zu entwickeln.

▪ Sprint Planning:

- Im Sprint Planning legt das Team fest, welche User Stories in den nächsten Sprint aufgenommen werden sollen. Diese Planung ist iterativ und kurzzyklisch, sodass das Team flexibel bleibt und sich an geänderte Anforderungen anpassen kann.
- **Deliberate Practice:** Üben von Sprint Plannings, z. B. durch Rollenspiele, hilft dem Team, effizient zu planen und Prioritäten zu setzen, indem es das "MVP" (Minimum Viable Product) für jeden Sprint identifiziert.

2. Code (Programmieren)

▪ Daily Stand-ups:

- Das tägliche Stand-up bietet dem Team eine Gelegenheit, den aktuellen Fortschritt im Codieren zu besprechen, Hindernisse zu teilen und Unterstützung zu erhalten. Der Fokus liegt darauf, das Team synchronisiert und agil zu halten, damit alle wissen, woran die anderen arbeiten und welche Blocker vorhanden sind.
- **Deliberate Practice:** Ein kurzes, gezieltes Üben von Kommunikationsfähigkeiten und präzisen Status-Updates kann helfen, den Nutzen des Stand-ups zu maximieren und das Meeting effizient zu halten.

▪ Pair Programming und Code Reviews:

- Pair Programming fördert den Wissenstransfer und die Qualitätssicherung. Zwei Entwickler arbeiten an einer Aufgabe und teilen dabei kontinuierlich Wissen und Ideen. Code Reviews dienen dazu, Fehler zu identifizieren und den Code zu verbessern, bevor er integriert wird.
- **Deliberate Practice:** Regelmäßige Code Reviews und gezielte Übungen in Pair Programming helfen, Standards zu etablieren und die

Codequalität zu verbessern. Auch Rollenspiele, bei denen Teammitglieder Feedback geben und nehmen, fördern eine konstruktive Feedback-Kultur.

3. Build (Bauen)

▪ **Continuous Integration (CI):**

- CI sorgt dafür, dass der Code bei jeder Änderung automatisiert gebaut und getestet wird, sodass Fehler frühzeitig erkannt werden. Die CI-Pipeline ist eine wichtige Praxis, um Entwicklung und Qualitätssicherung eng zu verknüpfen.
- **Deliberate Practice:** Üben des CI-Pipeline-Setups und der Optimierung dieser Pipeline (z. B. durch pair programming oder das Durchführen von Workshops) hilft, die Effizienz zu steigern. Teams können gezielt Rollbacks üben, um Sicherheit in der Prozesssteuerung zu gewinnen.

4. Test (Testen)

▪ **Sprint Reviews:**

- Tests sind essenziell, um sicherzustellen, dass die entwickelten Features den Anforderungen entsprechen. In Sprint Reviews werden diese Tests (vor allem Akzeptanztests) genutzt, um zu zeigen, dass die Arbeit des Sprints den Erwartungen entspricht.
- **Deliberate Practice:** Test-Driven Development (TDD) und Übungen zur Automatisierung von Tests sind gezielte deliberate practices, um sicherzustellen, dass der Code von Anfang an auf Qualität ausgerichtet ist und langfristig gewartet werden kann.

▪ **Explorative Tests:**

- Zusätzlich zu automatisierten Tests können Teams explorative Tests verwenden, bei denen sich die Entwickler auf ungeplante und zufällige Nutzungsszenarien konzentrieren, um unvorhergesehene Schwachstellen zu entdecken.
- **Deliberate Practice:** Hier kann das Team durch gezielte explorative Testing-Sessions und Pairing neue Testansätze entwickeln und „out-of-the-box“-Denkweisen fördern.

5. Package (Paketieren)

▪ **Sprint Retrospective:**

- Beim Paketieren von Features kann die Retrospektive genutzt werden, um Feedback zu sammeln, vor allem hinsichtlich der Zusammenarbeit zwischen Entwicklung und Operations. Die Retrospektive bietet Raum,

- um über den Ablauf und mögliche Verbesserungen des Paketierungsprozesses zu sprechen.
- **Deliberate Practice:** Hier können Teams Übungen im Paketieren und in der Zusammenarbeit mit anderen Abteilungen durchführen. Dazu gehören gezielte Rollback-Übungen oder Test-Rollouts, um Fehlerfälle zu identifizieren und den Prozess zu verbessern.

6. Deploy (Bereitstellen)

- **Release Planning:**
 - In agilen Projekten sind kleine, kontinuierliche Releases üblich. Release Planning Meetings legen fest, wann und wie neue Versionen an die Nutzer ausgeliefert werden. Hier können Teams ihre Deployment-Strategien besprechen und verbessern, z. B. durch Blau-Grün- oder Canary-Deployments.
 - **Deliberate Practice:** Übungen wie "Chaos Engineering" (gezielte Einbringung von Fehlern) und „Dark Launches“ (Teilveröffentlichungen für Tests) helfen dem Team, den Deployment-Prozess zu testen und sicherzustellen, dass er stabil und fehlerresistent ist.

7. Operate (Betreiben)

- **Cross Team Daily (wie z. B. von dir erwähnt):**
 - Ein tägliches, teamübergreifendes Meeting, in dem die Plattformgesundheit und Vorfälle besprochen werden, hilft dabei, die Stabilität der Anwendung aufrechtzuerhalten und gemeinsam an Problemen zu arbeiten. Dieses Meeting stellt sicher, dass keine kritischen Blocker übersehen werden und dass alle Teams ihre Betriebsaufgaben im Blick haben.
 - **Deliberate Practice:** Übungen im Incident Management helfen dem Team, in Stresssituationen (z. B. bei Incidents) schneller und sicherer zu reagieren und die Serviceverfügbarkeit zu verbessern. Auch Simulationen von Major Incidents können helfen, den Umgang mit echten Zwischenfällen zu üben.

8. Monitor (Überwachen)

- **Retrospective Meetings:**
 - Monitoring ermöglicht es, die Stabilität und Leistung der Anwendung zu überwachen und Schwachstellen frühzeitig zu erkennen. Diese Erkenntnisse können in der Retrospektive diskutiert werden, um zukünftige Verbesserungen einzuplanen.
 - **Deliberate Practice:** Teams können sich gezielt in Monitoring-Techniken und der Interpretation von Dashboards und Logs schulen, um kritische Metriken zu überwachen und Alarmeinstellungen

anzupassen. Durch das Üben von Szenarien kann das Team lernen, schnell und gezielt auf bestimmte Vorfälle zu reagieren.

Zusammenfassung

Die Integration dieser Schritte in agile Meetings und deliberate practices hilft dem Team, agile Prinzipien wie kontinuierliches Feedback, iteratives Arbeiten und kontinuierliche Verbesserung umzusetzen. Jedes Meeting, jeder Zyklus und jede deliberate practice dient dazu, die Fähigkeiten des Teams zu verfeinern und sicherzustellen, dass die entwickelten Produkte stabil, hochwertig und auf die Bedürfnisse der Nutzer ausgerichtet sind.

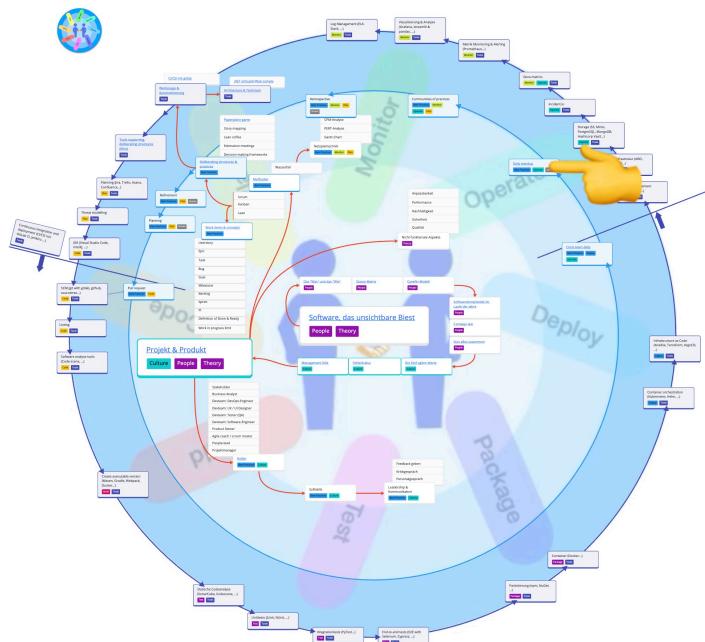
Daily_standup.de

Produkt & Projekt / Methoden / Deliberating structures

Daily Standups



- Das Daily Standup:
Definition, Ablauf & 1 Pro-
Tipp - Agile Academy
- Daily Scrum Meeting: 5
Tipps für eine gute
Besprechung - Agile Scrum
Group
- Daily Standup-Meeting:
Tipps und Vorteile - Agile
Scrum Group
- Daily Standup - Regeln,
Ablauf & Vorlagen |
Consulting LIFE
- What is a stand up meeting
& tips to run one - Atlassian
- Daily Scrum: Definition,
Ablauf und Tipps für ein ...
- acquisa
- Daily Scrum oder Daily
Stand-up (+
Videoanleitung) - Agile
Scrum Group



[Daily Standup Scrum Ritual - Ablauf, Vorteile und Tipps](#)
[Das Daily Scrum oder Daily Standup in Scrum - Agile Academy](#)
[Daily-Scrum-Meeting: Teilnehmer, Inhalte & Praxistipps - business-wissen.de](#)

Sinn und Zweck des Daily Standups

- **Statusüberblick:** Kurzes Update der Teammitglieder über den Fortschritt
- **Frühzeitige Identifikation von Hindernissen:** Probleme direkt ansprechen
- **Synchronisation des Teams:** Alle bleiben informiert und auf dem gleichen Stand
- **Fokussierung auf Sprintziele:** Konzentration auf gemeinsame Ziele und Prioritäten
- **Vermeidung langer Meetings:** Kurze, effiziente Besprechung

Ablauf des Daily Standups:

- **Dauer:** Maximal 15 Minuten
- **Zeitpunkt:** Häufig zu Beginn des Arbeitstags
- **Teilnehmer:** Entwicklungsteam, Scrum Master, ggf. Product Owner
- **Format:**
 - Jedes Teammitglied beantwortet drei Fragen:
 1. Was habe ich seit dem letzten Standup erreicht?
 2. Was werde ich heute tun?
 3. Welche Hindernisse gibt es?
 - **Keine detaillierten Diskussionen:** Tiefergehende Gespräche nach dem Standup

Ein **Daily Standup** (oder **Daily Scrum**) ist ein kurzes, regelmäßiges Treffen in agilen Teams, das typischerweise einmal pro Tag stattfindet. Ziel ist es, den aktuellen Status der Arbeit zu besprechen, Hindernisse zu identifizieren und sicherzustellen, dass das Team auf dem richtigen Weg ist, um die Sprintziele zu erreichen. Der Fokus liegt auf der Transparenz und der Synchronisation des Teams. Es ist ein wichtiger Bestandteil agiler Arbeitsmethoden wie Scrum oder Kanban.

Sinn und Zweck des Daily Standups:

1. **Statusüberblick:** Jedes Teammitglied gibt ein kurzes Update, was in den letzten 24 Stunden erreicht wurde und was als nächstes geplant ist.
2. **Frühzeitige Identifikation von Hindernissen:** Probleme oder Blockaden, die die Arbeit behindern, können sofort angesprochen werden, sodass das Team darauf reagieren kann.

3. **Synchronisation des Teams:** Durch den täglichen Austausch bleiben alle Teammitglieder informiert und können sich gegenseitig unterstützen, insbesondere wenn ihre Aufgaben voneinander abhängen.
4. **Fokussierung auf Sprintziele:** Es hilft, das Team auf das gemeinsame Ziel und die Prioritäten des aktuellen Sprints zu fokussieren.
5. **Vermeidung langer Meetings:** Da es kurz ist, spart es Zeit im Vergleich zu langen Statusmeetings und fördert eine effiziente Zusammenarbeit.

Ablauf des Daily Standups:

- **Dauer:** Das Standup dauert in der Regel 15 Minuten oder weniger.
- **Zeitpunkt:** Es findet oft zu Beginn des Arbeitstags statt, ist aber flexibel, je nach Teampräferenz.
- **Teilnehmer:** Alle Mitglieder des Entwicklungsteams, der Scrum Master und ggf. der Product Owner (je nach Scrum-Praktik).
- **Format:**
 - Jedes Teammitglied beantwortet drei Fragen:
 1. **Was habe ich seit dem letzten Standup erreicht?**
 2. **Was werde ich heute tun?**
 3. **Welche Hindernisse stehen mir im Weg?**
 - Es wird nicht in detaillierte Diskussionen gegangen – falls tiefergehende Gespräche nötig sind, finden sie nach dem Standup mit den betroffenen Personen statt.

Das Daily Standup ist **kein Statusbericht für den Manager**, sondern eine Gelegenheit für das Team, sich selbst zu organisieren und mögliche Probleme rechtzeitig zu adressieren.

Hilfreiche Werkzeuge

Für ein effizientes Daily Standup eignen sich verschiedene Hilfsmittel, die Kommunikation, Transparenz und Nachverfolgbarkeit erleichtern:

1. Digitale Tools für Kollaboration und Aufgaben-Tracking:

- **Jira** oder **Trello**: Um Aufgaben, Fortschritt und Blocker sichtbar zu machen und jedem Teammitglied einen Überblick zu geben.
- **Asana** oder **ClickUp**: Für Aufgaben- und Projektmanagement; ermöglicht es, die geplanten Arbeiten und Blocker einfach zu dokumentieren.
- **Kanban-Boards**: Visualisierung von To-Do, In Progress, Done – optimal für Scrum- und Kanban-Teams, um die Arbeit im Fluss zu halten.

2. Kommunikationstools:

- **Slack, Microsoft Teams** oder **Discord**: Nützlich für kurze Standup-Updates in

Remote-Teams. Spezielle Channels für Daily Updates oder Blocker sorgen für eine klare Kommunikation.

- **Videokonferenz-Tools** wie **Zoom**, **Google Meet** oder **Microsoft Teams**: Für verteilte Teams ist ein kurzes Standup per Video die beste Möglichkeit, sich persönlich auszutauschen.

3. Dokumentationstools:

- **Confluence** oder **Notion**: Um tägliche Updates, Blocker und Entscheidungen festzuhalten und für spätere Referenz zu speichern.
- **Google Docs** oder **OneNote**: Einfache Tools zur schnellen Protokollierung von Standup-Punkten.

4. Monitoring- und Analyse-Tools:

- **Grafana**, **Prometheus** oder **Datadog**: Zur Visualisierung und Überwachung der Plattformgesundheit und Incident-Statistiken – hilfreich für Teams, die technische Incidents im Daily Standup diskutieren.
- **Jaeger** oder **New Relic**: Für die Analyse von Bottlenecks oder Performance-Problemen, um diese im Daily sichtbar zu machen.

5. Task-Timer oder Standup-Timer:

- **Time Timer** oder digitale Stoppuhren: Hilft, das Meeting kurz und fokussiert zu halten und die Redezeit zu begrenzen.
- **Pomodoro-Apps** oder Timer in Tools wie **Slack**: Nützlich, um den Überblick zu behalten und zu verhindern, dass das Meeting in die Länge gezogen wird.

6. Physical Boards oder Whiteboards (für Co-Located Teams):

- **Kanban- oder Scrum-Boards**: Für Teams, die vor Ort arbeiten, helfen physische Boards mit Notizzetteln, den Status von Aufgaben zu verfolgen.
- **Whiteboard oder Flipchart**: Zum schnellen Festhalten von Blockern oder Todos – kann bei Bedarf in digitale Tools übertragen werden.

Jedes dieser Hilfsmittel unterstützt ein klares und effizientes Standup, je nach Bedarf und Standort des Teams.

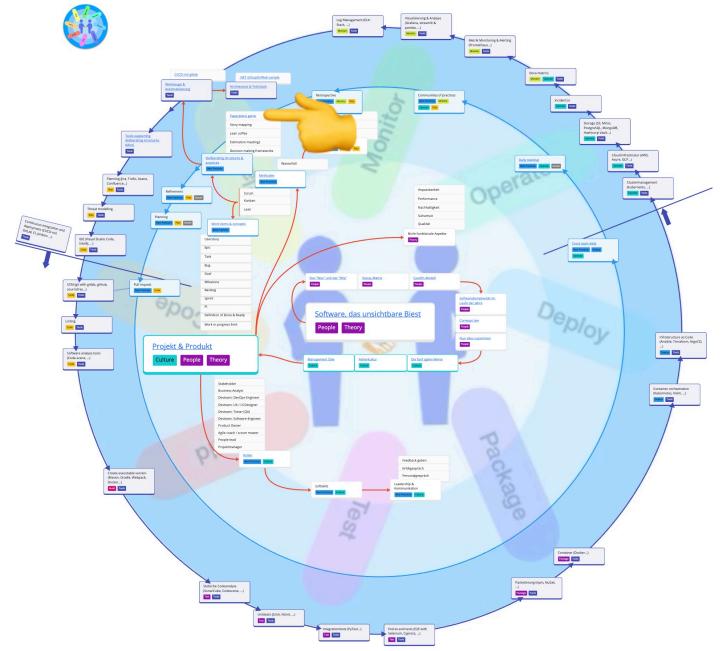
Paperplane_game.de

Produkt & Projekt / Methoden / Deliberating structures

Paperplane Spiel



Top 15 der bekanntesten Agile Games - butterflying.de
 Agile Spiele: Eine Übersicht und Kategorisierung - [Fraunhofer IESE](#)
 25 Agile Games to improve team performance and collaboration - [SessionLab](#)
 Agile games, simulations and learning activities - [Agile Coffee](#)
 25 Agile Spiele zur Steigerung der Leistung und Zusammenarbeit - [ClickUp](#)
 Agile Spiele: warum sie nützlich sind (+ 3 Beispiele) - [Agile Scrum Group](#)
 Agile Spiele und Simulationen in der Teamentwicklung - [Marco Spörl](#)
 25 Agile Games to Boost Team Performance and Collaboration - [ClickUp](#)
 ScrumTale - Scrum Simulation Game
 The 8 Best Agile Games for Improving Team Performance - [Invensis Learning](#)



Hier ist die gesamte Beschreibung des **Paper Plane Games** mit den integrierten Rollen und Veränderungen der Produktanforderungen:

Paper Plane Game – Agile Simulation mit Rollen und Veränderungen der Produktanforderungen

Ziele des Spiels:

- **Verständnis der agilen Prinzipien:** Die Studierenden erleben die Vorteile von Iteration, Feedback und kontinuierlicher Verbesserung.
- **Simulation von Teamrollen:** Einführung der wichtigsten agilen Rollen wie

Product Owner, Frontend-Entwickler, Backend-Entwickler und Scrum Master.

- **Veränderung von Produktanforderungen:** Simulieren von Änderungen während des Projekts, um das Management von sich ändernden Anforderungen zu erleben.

Benötigte Materialien:

- Papier (A4 oder ähnlich) für die Papierflieger.
- Stifte oder Marker zum Beschriften der Flieger.
- Maßband oder Lineal zur Messung der Flugdistanzen.
- (Optional) Notizkarten für Feedback.
- Materialien für verschiedene Designs (z.B. Klebeband, farbiges Papier, etc.).

Rollen im Spiel:

1. Product Owner (PO):

- Verantwortlich für die Produktvision und Priorisierung der Anforderungen.
Gibt die Anforderungen und Änderungen vor und stellt sicher, dass das Team auf das richtige Ziel hinarbeitet.

2. Frontend-Entwickler (FE):

- Fokussiert sich auf das Design des Fliegers und die Benutzererfahrung.
Sorgt dafür, dass der Flieger ästhetisch und ansprechend aussieht.

3. Backend-Entwickler (BE):

- Verantwortlich für die Stabilität des Fliegers und die Flugeigenschaften.
Stellt sicher, dass der Flieger stabil fliegt und strukturell gut funktioniert.

4. Scrum Master (SM):

- Moderiert das Team, hilft bei der Organisation des Prozesses und sorgt dafür, dass das Team effizient arbeitet und agile Prinzipien befolgt.

Ablauf des Spiels:

1. Einführung und Rollenverteilung (20 Minuten)

- **Ziele:** Den Teilnehmern die agilen Rollen und ihre Verantwortlichkeiten erklären.
- Die Teilnehmer werden in Teams eingeteilt, wobei jede Rolle im Team besetzt wird (PO, FE, BE, SM).
- Klärung der Aufgaben der jeweiligen Rollen:
 - **PO:** Gibt Anforderungen vor und priorisiert Änderungen.
 - **FE:** Arbeitet am Design des Fliegers.
 - **BE:** Stellt sicher, dass der Flieger funktional und stabil ist.
 - **SM:** Moderiert den Prozess und sorgt für eine effektive Zusammenarbeit.

2. Erste Iteration (15 Minuten)

- **Aktivität:**
 - Die Teams haben 5 Minuten Zeit, um ihren ersten Papierflieger zu bauen.
 - **PO:** Gibt die grundlegende Anforderung vor (z.B. „Der Flieger muss gut fliegen und stabil sein“).
 - **FE:** Konzentriert sich darauf, den Flieger ästhetisch zu gestalten.
 - **BE:** Sorgt dafür, dass der Flieger stabil genug ist, um zu fliegen.
 - **SM:** Hilft dabei, den Prozess zu moderieren und stellt sicher, dass jeder in seiner Rolle bleibt.
- **Flugversuch:** Jedes Team lässt seinen Flieger fliegen und misst die Distanz.

3. Feedback und Anpassungen (10 Minuten)

- **Aktivität:**
 - Jedes Team gibt sich gegenseitig Feedback über den Flieger.
 - **PO:** Priorisiert das Feedback und bestimmt, welche Änderungen für die nächste Iteration wichtig sind.
 - **FE und BE:** Überlegen, wie sie den Flieger entsprechend den Anforderungen des PO verbessern können.
 - **SM:** Moderiert die Diskussion und stellt sicher, dass die Verbesserungen schnell umgesetzt werden.

4. Einführung von Änderungen der Produktanforderungen (15 Minuten)

- **Aktivität:**
 - Nach der ersten Iteration führt der **PO** eine Änderung der Produktanforderungen ein, z.B.:
 - „Der Flieger muss aus mindestens zwei verschiedenen Papierarten bestehen.“
 - „Der Flieger muss ein Designmerkmal haben (z.B. eine Markierung oder ein Logo).“
 - **FE:** Arbeitet an der ästhetischen Anpassung, um das Design zu ändern.
 - **BE:** Sorgt dafür, dass die Änderungen die Flugeigenschaften des Fliegers nicht beeinträchtigen.
 - **SM:** Stellt sicher, dass die Anforderungen klar sind und dass das Team effizient zusammenarbeitet, um die Änderungen umzusetzen.

5. Weitere Iterationen (30 Minuten)

- **Aktivität:**
 - Weitere Iterationen, bei denen neue Anforderungen oder Änderungen hinzugefügt werden. Jede Runde könnte eine neue Anforderung beinhalten:
 - „Der Flieger muss in der Lage sein, mindestens zwei Kurven in der Luft zu fliegen.“
 - „Der Flieger muss in weniger als 5 Sekunden gefaltet werden.“
 - In jeder Iteration:

- **PO:** Gibt die neue Anforderung vor und sorgt dafür, dass das Team die richtigen Prioritäten setzt.
- **FE und BE:** Passen ihre Arbeit an, um die neuen Anforderungen zu erfüllen.
- **SM:** Moderiert den Prozess, achtet auf den Fortschritt und sorgt für einen effektiven Ablauf.
- **Flugversuche:** Nach jeder Iteration fliegen die Teams ihren Flieger erneut und messen die Distanz.

6. Abschlussdiskussion (15 Minuten)

- **Aktivität:**
 - Reflexion über die Erfahrung:
 - Wie haben die Teams auf die Änderungen reagiert?
 - Welche Herausforderungen gab es bei der Anpassung der Flieger an die neuen Anforderungen?
 - Wie haben sich die Teamrollen auf den Erfolg der Iterationen ausgewirkt?
 - **PO:** Diskutiert, wie die Anforderungen priorisiert wurden und ob der Fokus richtig gesetzt war.
 - **FE und BE:** Teilen ihre Erfahrungen, wie sie zusammengearbeitet haben, um Design und Funktionalität in Einklang zu bringen.
 - **SM:** Reflektiert darüber, wie der agile Prozess während des Spiels unterstützt wurde.
-

Schlussfolgerungen und Agile Prinzipien

- **Iterative Verbesserung:** Die Teams haben gelernt, wie wichtig es ist, regelmäßig Feedback zu erhalten und den Flieger basierend auf diesem Feedback zu verbessern – ähnlich wie bei der kontinuierlichen Verbesserung in agilen Projekten.
- **Zusammenarbeit und Kommunikation:** Die Simulation fördert die Bedeutung der Zusammenarbeit zwischen den verschiedenen Rollen. Der PO sorgt für die Priorisierung der Anforderungen, der FE kümmert sich um das Design, der BE um die Stabilität, und der SM sorgt für den reibungslosen Ablauf.
- **Veränderung der Anforderungen:** Die Einführung von Änderungen der Produktanforderungen in jeder Runde zeigt, wie agile Teams mit wechselnden Anforderungen umgehen und schnell reagieren können.

Mit dieser Simulation können die Studierenden die dynamische und kollaborative Natur agiler Methoden in einer praktischen, interaktiven Weise erleben.

Paperplane Spiel Begleitprogramm

Das folgende **Agile Paper Plane Game**-Programm simuliert den Einsatz agiler Methoden, indem Teams mit verschiedenen Rollen (z.B. Product Owner, Frontend- und Backend-Entwickler, Scrum Master) arbeiten, um Papierflieger zu bauen. In jeder Iteration haben die Teams 30 Sekunden Zeit, einen Flieger zu erstellen, der einer vorgegebenen Anforderung entspricht. Nach jeder Iteration erhalten sie Feedback und müssen ihren Flieger anpassen. Das Programm zeigt den gesamten Prozess, einschließlich der Anzeige von Anforderungen, eines Countdown-Timers für jede Iteration und der Möglichkeit, Feedback zu geben, was die Prinzipien der kontinuierlichen Verbesserung und Anpassung in agilen Projekten simuliert.

```
In [ ]: import time
import random
import ipywidgets as widgets
from IPython.display import display, clear_output

# Define the roles
roles = ["Product Owner", "Frontend Engineer", "Backend Engineer", "Scrum Master"]

# Define the possible changes in requirements
requirements_changes = [
    "The paper plane must be made of at least two different types of paper.",
    "The paper plane must have a design feature (e.g., a logo or a marking).",
    "The paper plane must perform at least two turns in the air.",
    "The paper plane must be folded in less than 5 seconds."
]

# Define the teams
teams = {
    "Team A": {"PO": "", "FE": "", "BE": "", "SM": ""},
    "Team B": {"PO": "", "FE": "", "BE": "", "SM": ""}
}

# Create text widgets to input role names
role_inputs = {}
for team in teams:
    role_inputs[team] = {}
    for role in roles:
        role_inputs[team][role] = widgets.Text(
            description=f'{role} for {team}:',
            layout=widgets.Layout(width='300px')
        )
    display(role_inputs[team][role])

# Time tracking for iterations with countdown
def time_iteration(duration):
    remaining_time = duration
    timer_label = widgets.Label(value=f"Remaining time: {remaining_time}")
    display(timer_label)

    def countdown():
        nonlocal remaining_time
        while remaining_time > 0:
            time.sleep(1)
            remaining_time -= 1
    countdown()
```

```

        timer_label.value = f"Remaining time: {remaining_time}s"

    # Once time is up, display the message for feedback
    timer_label.value = f"Time's up! You can now provide feedback."
    return remaining_time

countdown()

# Function to start the game
def start_game():
    print("Welcome to the Agile Paper Plane Game!\n")

    # Get roles from inputs
    for team in teams:
        for role in roles:
            teams[team][role] = role_inputs[team][role].value

    print("\nGame Starting...\n")

    # Start iterations
    for iteration in range(3):
        print(f"\n--- Iteration {iteration + 1} ---")

        # Display the requirement for this iteration
        new_requirement = random.choice(requirements_changes)
        print(f"\nNew Requirement: {new_requirement}")

        # Simulate time taken to build the plane with a countdown
        print("You have 30 seconds to complete this iteration.")
        time_iteration(30) # 30 seconds countdown per iteration

        # Provide feedback after each iteration
        print("\nFeedback Time:")
        feedback = widgets.Text(description="Feedback:")
        display(feedback)

        def on_submit_change(change):
            print(f"Feedback: {feedback.value}")

        feedback.observe(on_submit_change, names='value')

        # Simulate time taken to build the plane with a countdown
        print("You have 10 seconds to complete the feedback.")
        time_iteration(10) # 10 seconds countdown per feedback

        # After countdown and feedback, start the next iteration
        print("\nStarting next iteration...\n")
        #clear_output(wait=True) # Clears the output to keep it clean a

    # Button to start the game
    start_button = widgets.Button(description="Start Game")
    start_button.on_click(lambda x: start_game())
    display(start_button)

```

Text(value='', description='Product Owner for Team A:', layout=Layout(width='300px'))

```
Text(value='', description='Frontend Engineer for Team A:', layout=Layout(width='300px'))
Text(value='', description='Backend Engineer for Team A:', layout=Layout(width='300px'))
Text(value='', description='Scrum Master for Team A:', layout=Layout(width='300px'))
Text(value='', description='Product Owner for Team B:', layout=Layout(width='300px'))
Text(value='', description='Frontend Engineer for Team B:', layout=Layout(width='300px'))
Text(value='', description='Backend Engineer for Team B:', layout=Layout(width='300px'))
Text(value='', description='Scrum Master for Team B:', layout=Layout(width='300px'))
Button(description='Start Game', style=ButtonStyle())
Welcome to the Agile Paper Plane Game!
```

Game Starting...

--- Iteration 1 ---

New Requirement: The paper plane must perform at least two turns in the air.
You have 30 seconds to complete this iteration.
Label(value='Remaining time: 30s')
Feedback Time:
Text(value='', description='Feedback:')
You have 10 seconds to complete the feedback.
Label(value='Remaining time: 10s')
Starting next iteration...

--- Iteration 2 ---

New Requirement: The paper plane must be made of at least two different types of paper.
You have 30 seconds to complete this iteration.
Label(value='Remaining time: 30s')
Feedback Time:
Text(value='', description='Feedback:')
You have 10 seconds to complete the feedback.
Label(value='Remaining time: 10s')
Starting next iteration...

--- Iteration 3 ---

New Requirement: The paper plane must be folded in less than 5 seconds.
You have 30 seconds to complete this iteration.
Label(value='Remaining time: 30s')
Feedback Time:
Text(value='', description='Feedback:')
You have 10 seconds to complete the feedback.
Label(value='Remaining time: 10s')

Starting next iteration...

Erklärung:

1. **Rollen und Teamaufbau:** Zu Beginn des Spiels müssen die Teilnehmer für jedes Team ihre Rollen festlegen (Product Owner, Frontend-Entwickler, Backend-Entwickler, Scrum Master). Jede Rolle hat spezifische Aufgaben:
 - Der **Product Owner** legt die Anforderungen fest und priorisiert Änderungen.
 - Der **Frontend-Entwickler** ist für das Design des Fliegers verantwortlich.
 - Der **Backend-Entwickler** sorgt für die Flugeigenschaften und Stabilität des Fliegers.
 - Der **Scrum Master** sorgt dafür, dass der Prozess reibungslos abläuft und unterstützt das Team bei der Zusammenarbeit.
 2. **Iterationen und Anforderungen:** In jeder Iteration haben die Teams 30 Sekunden Zeit, um ihren Flieger zu bauen, wobei eine neue Anforderung durch den Product Owner vorgegeben wird. Diese Anforderungen können Änderungen im Design, der Funktionalität oder der Flugeigenschaft des

Fliegers betreffen.

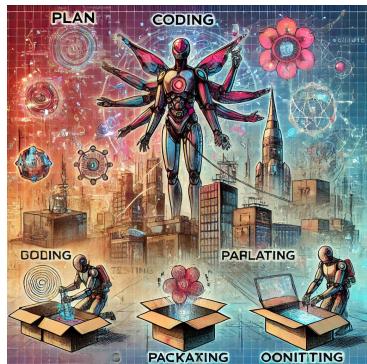
3. **Countdown:** Der Countdown zeigt die verbleibende Zeit für jede Iteration an. Nach Ablauf der Zeit müssen die Teams Feedback zu ihren Fliegern geben, das anschließend in die nächste Iteration integriert wird.
4. **Feedback:** Nach jeder Iteration können die Teams ihr Feedback zu den Anpassungen und dem Bau des Fliegers geben. Dies stellt sicher, dass die Teams auf das Feedback reagieren und die Anforderungen kontinuierlich verbessern.

Das Spiel fördert die Zusammenarbeit im Team, das schnelle Reagieren auf Änderungen und das Anwenden von agilem Feedback in einem interaktiven und spielerischen Rahmen.

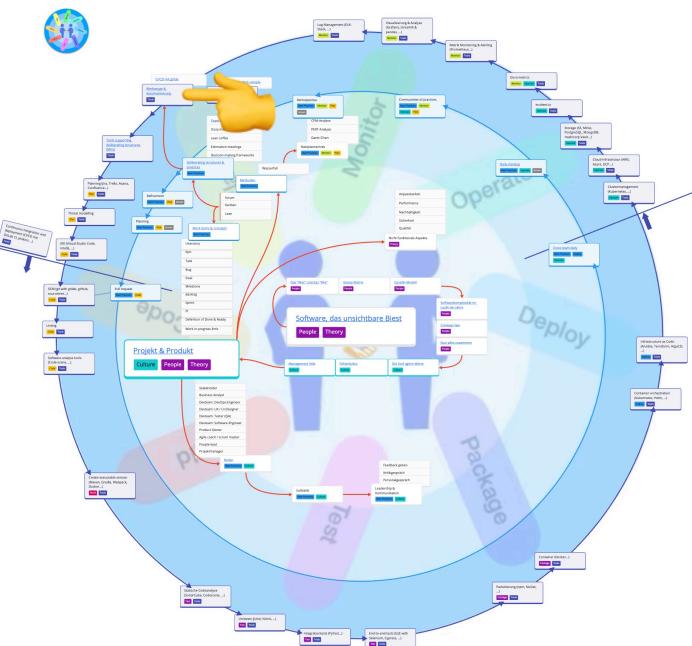
Tools_Automation.de

[Produkt & Projekt / Methoden / Deliberating structures](#)

Werzeuge und Automatisierung



- 33 Best DevOps Automation Tools [2024] - LambdaTest
 - 31 Crucial DevOps Automation Tools (UPDATED 2024) - CloudZero
 - 25 Best DevOps Automation Tools in 2024 - BrowserStack
 - 15 DevOps Automation Tools for SaaS Companies - ClickIT
 - Top 10 DevOps Tools für Automatisierung, Überwachung und ... - ClickUp
 - 15 Best DevOps Tools: The Comprehensive List | 2024 - Katalon
 - Entdecke die 30 besten DevOps Tools, auf die du im Jahr 2024 ... - Kinsta
 - A Complete Guide to 25+ Most Popular DevOps Automation Tools - Veritis
 - DevOps for SaaS Applications in 2024: The Complete Guide - Radixweb
 - 22 Best DevOps Automation Tools: A Comprehensive Guide Of 2024 - The CTO Club



In der Softwareentwicklung und speziell in der DevOps-Pipeline stellen die Phasen *plan, code, build, test, package, deploy, operate* und *monitor* wichtige Schritte im Entwicklungs- und Betriebszyklus dar. Jede dieser Phasen lässt sich durch Automatisierung und spezifische Tools unterstützen, um die Effizienz, Konsistenz und Qualität zu verbessern. Hier sind die Phasen und einige Beispiele für gängige Tools und Automatisierungsansätze:

1. Plan

- **Beschreibung:** In dieser Phase geht es um die Planung und Vorbereitung der Arbeit, oft mithilfe von Projektmanagement- und Kollaborationstools.
 - **Tools:**
 - **Jira:** Aufgaben- und Projektmanagement.
 - **Trello** oder **Asana:** Task-Management und Planung.
 - **Confluence:** Dokumentation und Zusammenarbeit.
 - **Automatisierung:** Ticket-Erstellung für wiederkehrende Aufgaben, Integration von Projektmanagement-Tools mit Versionskontrolle für Statusaktualisierungen.

2. Code

- **Beschreibung:** Schreiben und Versionieren von Code, oft unterstützt durch Versionskontrollsysteme.
- **Tools:**
 - **Git** (GitHub, GitLab, Bitbucket): Versionskontrolle und kollaborative Entwicklung.
 - **IDEs** wie **Visual Studio Code** oder **IntelliJ** mit Plugins für Codequalität und Linting.
- **Automatisierung:**
 - Code-Review-Prozesse automatisieren, z.B. über Pull Requests.
 - Automatische Linting- und Formatierungsprüfungen beim Einchecken von Code.

3. Build

- **Beschreibung:** Erstellen ausführbarer Versionen der Software, z.B. in Form von Binärdateien oder Containern.
- **Tools:**
 - **Maven** oder **Gradle** für Java-Projekte.
 - **Webpack** für JavaScript-Anwendungen.
 - **Docker** für containerisierte Builds.
- **Automatisierung:**
 - CI/CD-Pipelines in **Jenkins** oder **GitLab CI** zur Automatisierung von Builds.
 - Build-Benachrichtigungen oder automatische Abbrüche bei Build-Fehlern.

4. Test

- **Beschreibung:** Sicherstellen, dass die Software wie erwartet funktioniert, durch verschiedene Testarten wie Unit, Integration und End-to-End-Tests.
- **Tools:**
 - **JUnit**, **NUnit** für Unit-Tests.
 - **Selenium** oder **Cypress** für End-to-End-Tests.
 - **SonarQube** für Code-Qualität und statische Code-Analyse.
- **Automatisierung:**
 - Testausführung als Teil der CI/CD-Pipeline.
 - Automatische Benachrichtigung und Testreporting bei Fehlern.

5. Package

- **Beschreibung:** Bündeln der Anwendung und deren Abhängigkeiten in einer distributablen Form.
- **Tools:**
 - **Docker**: Erstellung von Containern.
 - **npm** (für JavaScript-Projekte) und **NuGet** (für .NET-Projekte) für das

Paketmanagement.

- **Automatisierung:**

- Versionierung und Deployment von Packages direkt aus der CI/CD-Pipeline.
- Signieren und Veröffentlichen von Packages in registrierten Repositories.

6. Deploy

- **Beschreibung:** Bereitstellung der Software in der Zielumgebung (z.B. Staging oder Produktion).
- **Tools:**
 - **Kubernetes** und **Helm** für die Bereitstellung in Container-Orchestrierung.
 - **Ansible** und **Terraform** für Infrastruktur als Code (IaC).
 - **ArgoCD** für GitOps-basierte Deployments.
- **Automatisierung:**
 - Automatisierte Rollouts durch CI/CD.
 - Blue-Green- oder Canary-Deployments zur Minimierung von Risiken bei Produktions-Updates.

7. Operate

- **Beschreibung:** Betrieb und Verwaltung der Software im Produktionsumfeld.
- **Tools:**
 - **Kubernetes** für das Cluster-Management und den Betrieb von Containern.
 - **AWS, Azure, GCP** für die Cloud-Infrastruktur.
 - **HashiCorp Vault** zur sicheren Verwaltung von Secrets.
- **Automatisierung:**
 - Autoscaling und Self-Healing-Mechanismen in Kubernetes.
 - Automatisierte Health Checks und Self-Healing-Strategien.

8. Monitor

- **Beschreibung:** Überwachung der Anwendung und Infrastruktur, um frühzeitig auf Probleme reagieren zu können.
- **Tools:**
 - **Prometheus** für Metrik-Monitoring und Alerting.
 - **Grafana** zur Visualisierung und Analyse von Metriken.
 - **ELK Stack** (Elasticsearch, Logstash, Kibana) für das Log-Management.
- **Automatisierung:**
 - Alarme und automatisierte Reaktionssysteme, die bei Anomalien Benachrichtigungen auslösen.
 - Dashboards und automatisierte Berichte zur kontinuierlichen Überwachung und Optimierung.

Diese Tools und Automatisierungen bieten eine nahtlose, durchgängige

Unterstützung entlang des DevOps-Zyklus und tragen zur Effizienz und Konsistenz in der Softwareentwicklung bei. Automatisierte Pipelines sorgen dafür, dass jeder dieser Schritte kontinuierlich und wiederholbar ist, was zu einer höheren Softwarequalität und geringeren Betriebskosten führt.

Software Product Mastering

[See the board](#)



Software Product Mastering ↗

Modified 19 hours ago

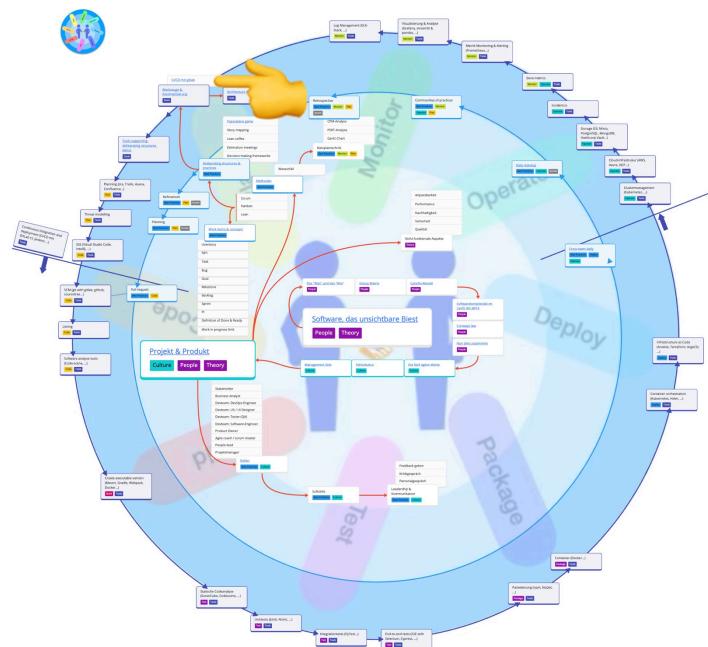
CICD_gitlab.de

Produkt & Projekt / Methoden / Deliberating structures / Werkzeuge &

Software-Projekt mit gitlab



[Get started with GitLab CI/CD - GitLab Documentation](#)
[What is CI/CD? - GitLab](#)
[CI/CD YAML syntax reference - GitLab Documentation](#)
[Tutorial: Create and run your first GitLab CI/CD pipeline](#)
[What is a CI/CD pipeline? - GitLab](#)
[CI/CD pipelines - GitLab Documentation](#)
[Continuous Integration and Delivery - GitLab](#)
[Tutorial: Set up CI/CD steps - GitLab Documentation](#)
[CI/CD development guidelines - GitLab Documentation](#)
[Wie kontinuierliche Integration und kontinuierliche Lieferung zusammenarbeiten - GitLab](#)



WiP

Die hier erklärten Schritte sind aktuell noch ungetestet.

Hier eine kurze und stichpunktartige Zusammenfassung, wie GitLab für ein Webapp-Softwareprojekt mit .NET genutzt werden kann:

1. Repository einrichten:

- Erstelle ein GitLab-Repository für den Code.
- Struktur: `/src`, `/tests`, `/docs` für saubere Organisation.

2. CI/CD-Pipeline:

- Definiere eine `.gitlab-ci.yml` -Datei zur Automatisierung von Build, Test und Deployment.
- Beispiel: `dotnet build`, `dotnet test`, Deployment-Skripte.

3. Branching und Merge Requests:

- Nutze Branches für Features/Tasks.
- Merge Requests für Code-Reviews und Integration.

4. Code-Reviews:

- Verwende Merge Requests für Feedback und Code-Qualitätssicherung.

5. Dokumentation:

- Nutze GitLab Wiki für Projektdokumentation.
- Achte auf eine detaillierte `README.md` -Datei.

6. Aufgabenverwaltung:

- Verwende GitLab Issues zur Verwaltung von Aufgaben, Bugs und Features.
- Visualisiere den Fortschritt über Kanban-Board.

7. Kollaboration und Kommunikation:

- Nutze GitLab Discussions für Teamkommunikation.
- Erstelle Teams für verschiedene Gruppen oder Aufgaben.

8. Testing und Codequalität:

- Integriere Unit-Tests (z.B. mit xUnit, NUnit) in CI/CD.
- Verwende Tools wie SonarCloud zur Codequalität.

9. Deployment:

- Automatisiere das Deployment mit GitLab CI/CD (z.B. auf Server oder Cloud).
- Docker-Integration für containerisierte Anwendungen.

Das sorgt für eine effiziente Zusammenarbeit und kontinuierliche Qualitätssicherung im Projekt.

GitLab ist ein exzellentes Tool für die Zusammenarbeit an Softwareprojekten und kann eine Gruppe optimal unterstützen, wenn sie gemeinsam an einem Webapp-Projekt mit z.B. .NET-Technologie arbeiten. Hier sind einige grundlegende Schritte und Best Practices, wie du GitLab für das Projekt nutzen kannst:

1. Repository einrichten

- **Erstelle ein GitLab-Repository:** Zu Beginn des Projekts solltest du ein GitLab-Repository erstellen, in dem der gesamte Quellcode und die Ressourcen des Projekts gespeichert werden. Gehe dazu in dein GitLab-Profil und erstelle ein

neues Projekt.

- **Repository-Struktur festlegen:** Eine saubere und konsistente Struktur ist entscheidend. Eine typische Struktur für ein .NET-Webanwendungsprojekt könnte so aussehen:

```
/src
  /WebApp
    /Controllers
    /Models
    /Views
    /wwwroot
/tests
/docs
  ▪ /src für den Code der Webanwendung.
  ▪ /tests für Unit-Tests oder Integrationstests.
  ▪ /docs für die Projektdokumentation.
```

2. GitLab CI/CD für Automatisierung

GitLab CI/CD ermöglicht es, Build- und Deployment-Pipelines zu automatisieren.

- **CI/CD Pipeline einrichten:** Du kannst eine `.gitlab-ci.yml` Datei anlegen, um deine Build- und Deployment-Prozesse zu definieren.
- Beispiel einer `.gitlab-ci.yml` für .NET-Projekte:
`image: mcr.microsoft.com/dotnet/aspnet:6.0`

```
stages:
  - build
  - test
  - deploy

variables:
  DOTNET_CLI_HOME: "/tmp"

before_script:
  - apt-get update && apt-get install -y unzip
  - curl -sSL https://aka.ms/InstallAzureCLIDeb | sudo bash

build:
  stage: build
  script:
    - dotnet build WebApp/WebApp.csproj

test:
  stage: test
  script:
    - dotnet test WebApp.Tests/WebApp.Tests.csproj

deploy:
  stage: deploy
  script:
    - echo "Deploying to production..."
```

Hier könnte ein Skript zum Deployen auf einen Server folgen

- **Build:** Baut das .NET-Projekt.
- **Test:** Führt Unit-Tests aus.
- **Deploy:** Kann auf eine Serverumgebung oder auf Docker/Cloud-Services zielen.

3. Branches und Merge Requests

- **Branching-Strategie:** Um die Arbeit an verschiedenen Features und Aufgaben zu organisieren, sollten Branches verwendet werden. Ein typisches Vorgehen ist:
 - **Main-Branch:** Für die stabile Version des Projekts.
 - **Feature-Branches:** Jeder Student oder jedes Team arbeitet an einem Feature in einem eigenen Branch.
 - **Merge Requests:** Feature-Branches sollten über Merge Requests in den Main-Branch integriert werden, was es ermöglicht, Code-Reviews durchzuführen und sicherzustellen, dass der Code in die Hauptentwicklungsline integriert wird.

4. Code-Reviews

- **Merge Requests für Code-Reviews:** Jeder Merge Request sollte von einem anderen Studenten oder Mentor überprüft werden. GitLab ermöglicht es, Kommentare direkt im Code zu hinterlassen und Diskussionen zu führen.
- **Regeln für Pull-Requests:** Stelle sicher, dass der Code vor dem Mergen mit den festgelegten Codestandards übereinstimmt und Unit-Tests erfolgreich sind.

5. Dokumentation

- **Wikis nutzen:** GitLab bietet ein integriertes Wiki, das für die Projektdokumentation genutzt werden kann (z.B. Architektur, Setup-Anweisungen).
- **README-Datei:** Achte darauf, dass das Repository eine gut dokumentierte `README.md` Datei enthält, die den Studenten erklärt, wie sie das Projekt lokal einrichten und ausführen können.

6. Aufgaben und Issues verwalten

- **GitLab Issues:** Nutze GitLab Issues, um Aufgaben, Bugs und Features zu verwalten. Du kannst Labels, Meilensteine und Assignees nutzen, um das Projekt zu organisieren.
- **Kanban-Board:** GitLab bietet ein Kanban-Board, das dir hilft, Aufgaben und den Fortschritt des Projekts zu visualisieren. Du kannst Aufgaben nach "To Do",

"In Progress" und "Done" organisieren.

7. Kollaboration und Kommunikation

- **GitLab Discussions:** Ermutige die Studenten, Diskussionen in GitLab zu führen, wenn sie auf Probleme stoßen oder Fragen haben. Dies ist eine effektive Möglichkeit zur Zusammenarbeit.
- **Teams und Rollen:** Falls du mehrere Gruppen von Studenten hast, kannst du verschiedene Gruppen von GitLab-Teams erstellen, um bestimmte Repositories oder Bereiche zu verwalten.

8. Testing und Codequalität

- **Unit-Tests mit xUnit oder NUnit:** Stelle sicher, dass Unit-Tests mit .NET getestet werden. Dies kann durch Integration in die CI/CD-Pipeline erfolgen.
- **SonarCloud oder CodeClimate für Codequalität:** Du kannst Tools wie SonarCloud verwenden, um die Codequalität zu überwachen und sicherzustellen, dass Best Practices eingehalten werden.

9. Deployment

- **Deployment in Cloud/Server:** Du kannst GitLab CI/CD nutzen, um die Anwendung nach jedem Commit automatisch in eine Entwicklungs- oder Produktionsumgebung zu deployen.
- **Docker-Integration:** Falls die Anwendung containerisiert wird, kannst du Docker und GitLab CI/CD verwenden, um das Deployment zu automatisieren.

Durch diese Schritte wird GitLab zu einem effektiven Werkzeug für das Management und die Zusammenarbeit in einem .NET-basierten Webapp-Projekt. Es fördert das Teamwork, die Automatisierung und die Qualität des Codes.

Umgebungen im studentischen Umfeld

Entwicklungsumgebungen für Studenten

- **Visual Studio Code (VSCode):** Kostenlos, leichtgewichtig, unterstützt .NET, Docker, und DevContainers.
- **Visual Studio Community Edition:** Vollständige IDE, kostenlos für Studenten, bietet erweiterte .NET-Features.

Kostenlose Produktionsumgebungen

- **Azure for Students:** Kostenloses Guthaben für Hosting von Web-Apps, Datenbanken, und mehr.
- **GitHub Pages:** Kostenlos für statische Websites, geeignet für Blazor oder

einfache .NET-Anwendungen.

- **Heroku**: Kostenloser Plan für Web-Apps, einfaches Setup.
- **Vercel**: Kostenloses Hosting für statische und serverseitig gerenderte Web-Apps.
- **Glitch**: Kostenloses Hosting, einfache APIs und Web-Apps.
- **Netlify**: Kostenloses Hosting für statische Seiten, automatische Deployments.

Cloud-Services für Studenten

- **AWS Educate**: Kostenloses Guthaben für Cloud-Dienste wie EC2 und RDS.
- **Google Cloud Platform (GCP)**: Kostenlose Ressourcen für das Hosting von Anwendungen.

Docker und Containerization

- **Docker Desktop**: Lokal für Containerentwicklung, unterstützt Cloud-Deployment.
- **Docker Hub**: Kostenloses Speichern von Container-Images.

CI/CD und Automatisierung

- **GitLab CI/CD**: Automatisiertes Build- und Deployment, Tests und Builds direkt im Repository.
- **GitHub Actions**: Automatisierte Workflows für Builds, Tests und Deployments.

Im studentischen Umfeld gibt es viele kostengünstige oder sogar kostenlose Entwicklung- und Produktionsumgebungen, die ideal für Webanwendungen mit .NET-Technologie geeignet sind. Hier sind einige empfohlene Optionen:

1. Entwicklungsumgebungen (für lokale Entwicklung)

- **Visual Studio Code (VSCode)**:
 - **Kostenlos und Open-Source**: Ideal für .NET Core-Projekte. VSCode unterstützt .NET durch die Erweiterung „C#“ und bietet eine leichtgewichtige, anpassbare Entwicklungsumgebung.
 - **Features**: Code-Completion, Debugging, Git-Integration und vieles mehr.
 - **Docker Support**: Unterstützt Docker-Container für die Entwicklung von Anwendungen in isolierten Umgebungen.
 - **DevContainers**: Mit DevContainers können die Studierenden eine vollständige Entwicklungsumgebung in einer Docker-Umgebung direkt im Editor starten.
- **Visual Studio Community Edition**:
 - **Kostenlos für Studenten**: Visual Studio bietet eine vollständige IDE für .NET-Entwickler, mit erweiterten Funktionen wie Debugging, Refactoring

und Testmanagement.

- **Integration von Azure, Docker und anderen Tools:** Besonders nützlich, wenn du mit Azure oder Docker arbeitest.

2. Produktionsumgebungen (für das Hosting von Webapps)

Kostenlose Optionen für Hosting

- **Azure for Students:**
 - **Kostenloses Guthaben:** Microsoft bietet ein spezielles Programm für Studenten, das kostenlosen Zugang zu Azure-Diensten, wie App Services, Virtual Machines und SQL-Datenbanken bietet.
 - **Dienste:** Du kannst Web-Apps, APIs, Datenbanken und viele andere Services auf Azure nutzen.
 - **Nutzung für Produktions- und Entwicklungsumgebungen:** Azure bietet sowohl Entwicklungs- als auch Produktionslösungen für deine .NET-basierte Web-App.
- **GitHub Pages:**
 - **Kostenlos für statische Websites:** Ideal, wenn du eine einfache statische Webanwendung mit .NET oder Blazor erstellen möchtest.
 - **Deployment aus einem GitHub-Repository:** Erlaube den Studierenden, direkt aus ihrem GitHub-Repo zu deployen.
- **Heroku:**
 - **Kostenlose Web-Heroku Dynos:** Heroku bietet eine kostenlose Stufe, bei der du Webanwendungen und APIs hosten kannst. Diese ist gut geeignet für kleinere Webanwendungen.
 - **Einfaches Setup:** Besonders für Studenten, die nicht mit komplexeren Infrastruktur-Anbietern arbeiten möchten.
 - **.NET-Integration:** Heroku bietet Unterstützung für .NET-Anwendungen über Docker.
- **Vercel:**
 - **Kostenlos für Studenten:** Vercel bietet einen einfach zu bedienenden Service für das Hosting von statischen und serverseitig gerenderten Webanwendungen. Wenn deine Anwendung ein Frontend mit React, Next.js oder Blazor hat, ist Vercel eine gute Wahl.
 - **Integration mit GitHub:** Automatische Deployments direkt aus einem GitHub-Repository.
- **Glitch:**
 - **Kostenloses Hosting für Web-Apps:** Glitch ist eine weitere einfache Möglichkeit für das Hosting von Webanwendungen, wobei Studenten

schnelle Deployments durchführen können.

- **Einfache API-Verwaltung und Fullstack-Unterstützung.**

- **Netlify:**

- **Kostenlose Hosting-Pläne für kleine Apps:** Netlify bietet kostenlose Hosting-Pläne, die besonders gut für statische Seiten oder Jamstack-Architekturen geeignet sind.
- **Automatische Deployments:** Direktes Deployment aus Git-Repositories (GitHub, GitLab, Bitbucket).

Cloud-Services für Studenten

- **Amazon Web Services (AWS) Educate:**

- **Kostenloses Guthaben:** AWS bietet über das AWS Educate-Programm kostenlosen Zugang zu einer Vielzahl von Cloud-Diensten, die für das Hosting von Webanwendungen und Datenbanken genutzt werden können.
- **Dienste wie EC2, RDS und S3:** Diese eignen sich gut für skalierbare Webanwendungen.

- **Google Cloud Platform (GCP) for Education:**

- **Kostenlose Ressourcen und Guthaben:** Google bietet für Studierende kostenlose Cloud-Ressourcen, die du für das Hosting von .NET-Anwendungen nutzen kannst, inklusive Compute Engine und Firebase für Datenbanklösungen.

3. Docker und Containerization (lokale und Cloud-basierte Entwicklung)

- **Docker:**

- **Lokal mit Docker Desktop:** Docker eignet sich hervorragend für die Erstellung von Containerumgebungen, die die Entwicklung und das Testing von .NET-Anwendungen vereinfachen.
- **Cloud-Dienste wie AWS und Azure** bieten umfassende Unterstützung für Docker-Container und ermöglichen eine einfache Bereitstellung in der Cloud.

- **Docker Hub:**

- **Kostenlose Repositories:** Nutze Docker Hub, um Docker-Images zu speichern und in verschiedenen Umgebungen zu verwenden. Dies ist besonders hilfreich, um die Studenten an das Arbeiten mit Containern heranzuführen.

4. CI/CD und Automatisierung (für die Integration und Bereitstellung)

- **GitLab CI/CD:**
 - **Automatisiertes Deployment:** Mit GitLab CI/CD können die Studenten die Anwendung sowohl lokal als auch in die Cloud (z.B. auf Azure, AWS oder Heroku) deployen, sobald sie Änderungen in ihren Repositories vornehmen.
 - **Automatische Tests und Builds:** Führe Tests und Builds direkt im GitLab-Repository durch, bevor du eine Version deployen kannst.
- **GitHub Actions:**
 - **Automatisierte Workflows:** Wenn das Projekt auf GitHub basiert, können GitHub Actions genutzt werden, um Builds, Tests und Deployments zu automatisieren. Es gibt vorgefertigte Actions für .NET, die du direkt verwenden kannst.

Fazit

Für das studentische Umfeld bieten sich kostenlose oder kostengünstige Entwicklungs- und Produktionsumgebungen wie **Azure for Students**, **Heroku**, **GitHub Pages**, **Vercel** und **Docker** an. Diese Dienste ermöglichen eine einfache Einrichtung und Verwaltung von Webanwendungen und eignen sich hervorragend, um Studenten sowohl in der Entwicklung als auch im Deployment zu unterstützen. Ideal ist auch der Einsatz von CI/CD-Tools wie **GitLab** oder **GitHub Actions**, um die Prozesse zu automatisieren und eine professionelle Entwicklungsumgebung zu simulieren.

Gitlab und Azure for Students

Zusammenfassung des Deployments auf Azure for Students mit GitLab CI/CD

1. Azure Setup:

- Erstelle einen **Azure App Service** und ein **Service Principal** (mit den Berechtigungen **Contributor**).
- Hole dir die **Azure Subscription ID**, **Tenant ID**, **Client ID**, und **Client Secret** aus dem Azure-Portal.

2. GitLab CI/CD Konfiguration:

- **Image:** Verwende ein .NET Docker-Image (z.B. `mcr.microsoft.com/dotnet/aspnet:6.0`).
- **Stages:** Definiere `build`, `test`, `deploy`.
- **Before Script:** Installiere die **Azure CLI**.
- **Build Stage:** Baue das .NET-Projekt.
- **Test Stage:** Führe Unit-Tests aus.
- **Deploy Stage:**

- Logge dich mit dem Azure **Service Principal** ein (`az login`).
- Setze die Azure Subscription (`az account set`).
- Veröffentliche die Anwendung (`dotnet publish`).
- Deploye auf Azure App Service (`az webapp deploy`).

3. Secrets verwalten in GitLab:

- Gehe zu **Settings** → **CI / CD** → **Variables**.
- Füge die folgenden Variablen hinzu:
 - `AZURE_SUBSCRIPTION_ID` , `AZURE_TENANT_ID` ,
`AZURE_CLIENT_ID` , `AZURE_CLIENT_SECRET` , `AZURE_APP_NAME` ,
`AZURE_RESOURCE_GROUP` , `AZURE_APP_SERVICE_NAME` .
- Diese Variablen sind während des CI/CD-Prozesses verfügbar und schützen sensible Daten.

4. Sicherheit:

- Speichere keine sensiblen Daten direkt in `.gitlab-ci.yml` .
- GitLab Secrets sind verschlüsselt und sicher während des Deployments verfügbar.

```
image: mcr.microsoft.com/dotnet/aspnet:6.0
```

```
stages:
  - build
  - test
  - deploy

variables:
  DOTNET_CLI_HOME: "/tmp"

before_script:
  - curl -sSL https://aka.ms/InstallAzureCLIDeb | sudo bash

build:
  stage: build
  script:
    - dotnet build src/WebApp/WebApp.csproj

test:
  stage: test
  script:
    - dotnet test src/WebApp.Tests/WebApp.Tests.csproj

deploy:
  stage: deploy
  script:
    - echo $AZURE_CLIENT_SECRET | az login --service-principal -
      -username $AZURE_CLIENT_ID --password $AZURE_CLIENT_SECRET --tenant
      $AZURE_TENANT_ID
    - az account set --subscription $AZURE_SUBSCRIPTION_ID
    - dotnet publish src/WebApp/WebApp.csproj -c Release -o
      ./publish
```

```
- az webapp deploy --resource-group $AZURE_RESOURCE_GROUP --  
name $AZURE_APP_SERVICE_NAME --src-path ./publish --type zip  
only:  
- main
```

Fazit:

- **CI/CD** automatisiert das Builden, Testen und Deployen auf **Azure**.
- **Secrets** werden sicher in GitLab verwaltet.
- Deployment erfolgt durch Azure CLI innerhalb der GitLab Pipeline.

Das Deployment einer .NET-Anwendung auf **Azure for Students** über GitLab CI/CD erfordert einige grundlegende Schritte, um die Anwendung zu erstellen, zu testen und anschließend auf Azure zu deployen. Zusätzlich ist es wichtig, sensible Daten wie API-Keys oder Verbindungszeichenfolgen sicher zu verwalten. Hier ist eine detaillierte Anleitung, wie das in deiner `.gitlab-ci.yml` aussehen könnte und wie du **Secrets** sicher verwaltetest.

1. Azure-Setup

Bevor du mit dem Deployment beginnst, musst du sicherstellen, dass dein Azure-Account korrekt eingerichtet ist. Im Fall von **Azure for Students** erhältst du kostenloses Guthaben, aber du musst ein paar Dinge einrichten:

- **Azure App Service:** Erstelle einen Azure App Service, um deine Anwendung zu hosten.
- **Azure Service Principal:** Erstelle ein Service Principal (eine Identität, die von GitLab verwendet wird, um sich bei Azure zu authentifizieren), das GitLab die Berechtigung zum Deployment gibt.

2. Azure Service Principal erstellen

Um deine Azure App in GitLab zu deployen, musst du ein Service Principal erstellen, das du in deiner GitLab CI/CD Pipeline nutzen kannst:

1. Melde dich bei deinem Azure-Portal an.
2. Gehe zu **Azure Active Directory** → **App Registrations** → **New registration** und registriere eine neue Anwendung.
3. Erstelle ein **Secret** für diese App unter **Certificates & Secrets**.
4. Vergib die Berechtigungen **Contributor** auf der Ressourcengruppe oder App Service Ebene.

Die **Service Principal-Daten** (App-ID, Tenant-ID und Secret) müssen sicher in GitLab hinterlegt werden.

3. GitLab CI/CD Setup

In der `.gitlab-ci.yml` definierst du den gesamten CI/CD-Prozess, einschließlich des Builds, Tests und Deployments auf Azure.

Beispiel einer `.gitlab-ci.yml` Datei:

```
image: mcr.microsoft.com/dotnet/aspnet:6.0

stages:
  - build
  - test
  - deploy

variables:
  DOTNET_CLI_HOME: "/tmp"
  AZURE_SUBSCRIPTION_ID: "your-subscription-id"
  AZURE_TENANT_ID: "your-tenant-id"
  AZURE_CLIENT_ID: "your-client-id"
  AZURE_CLIENT_SECRET: "your-client-secret"
  AZURE_APP_NAME: "your-app-name"
  AZURE_RESOURCE_GROUP: "your-resource-group"
  AZURE_APP_SERVICE_NAME: "your-app-service-name"

before_script:
  - curl -sSL https://aka.ms/InstallAzureCLIDeb | sudo bash

build:
  stage: build
  script:
    - dotnet build src/WebApp/WebApp.csproj
  artifacts:
    paths:
      - bin/

test:
  stage: test
  script:
    - dotnet test src/WebApp.Tests/WebApp.Tests.csproj

deploy:
  stage: deploy
  script:
    # Log into Azure
    - echo $AZURE_CLIENT_SECRET | az login --service-principal \
      --username $AZURE_CLIENT_ID --password $AZURE_CLIENT_SECRET \
      --tenant $AZURE_TENANT_ID
    - az account set --subscription $AZURE_SUBSCRIPTION_ID
    # Publish to Azure App Service
    - dotnet publish src/WebApp/WebApp.csproj -c Release -o \
      ./publish
    - az webapp deploy --resource-group $AZURE_RESOURCE_GROUP \
      --name $AZURE_APP_SERVICE_NAME --src-path ./publish --type zip
    only:
      - main
```

Schritte erklärt:

- **Image:** Wir verwenden das offizielle .NET Docker-Image für die Build- und Test-Phasen.
- **Before Script:** Installiert die Azure CLI, die für das Deployment notwendig ist.
- **Build Stage:** Baut die .NET-Anwendung.
- **Test Stage:** Führt Unit-Tests aus.
- **Deploy Stage:** Loggt sich mit dem Service Principal in Azure ein, setzt die Azure Subscription und deployt die Anwendung auf den Azure App Service.

4. Verwalten von Secrets in GitLab

Die sensiblen Daten wie **Azure Client ID**, **Client Secret**, **Tenant ID** und **Subscription ID** sollten nicht direkt in der `.gitlab-ci.yml` Datei gespeichert werden. Stattdessen speicherst du sie sicher in GitLab CI/CD **Secrets** (Umgebungsvariablen), die nur zur Laufzeit verfügbar sind.

Schritte, um Secrets in GitLab zu verwalten:

1. Gehe in deinem GitLab-Projekt auf **Settings** → **CI / CD** → **Variables**.
2. Füge neue Variablen hinzu:
 - `AZURE_SUBSCRIPTION_ID` : Deine Azure Subscription ID.
 - `AZURE_TENANT_ID` : Deine Azure Tenant ID.
 - `AZURE_CLIENT_ID` : Deine Azure Application (Client) ID.
 - `AZURE_CLIENT_SECRET` : Dein Azure Application Secret.
 - `AZURE_APP_NAME` : Dein Azure App Name.
 - `AZURE_RESOURCE_GROUP` : Die Ressourcengruppe für den Azure App Service.
 - `AZURE_APP_SERVICE_NAME` : Der Name des Azure App Services, auf dem die App läuft.

Die Werte für diese Variablen erhältst du aus dem Azure-Portal, wenn du den Service Principal einrichtest.

5. Sicherheit

Achte darauf, dass du niemals sensible Daten direkt in deinem Code oder der `.gitlab-ci.yml` speicherst. GitLab CI/CD Secrets sind verschlüsselt und nur zur Laufzeit zugänglich, was die Sicherheit erhöht. Zudem solltest du darauf achten, dass nur die benötigten Personen oder Gruppen Zugriff auf die Secrets und das Projekt haben.

6. Überprüfen des Deployments

Nachdem das Deployment abgeschlossen ist, kannst du in Azure unter deinem App Service überprüfen, ob die Anwendung erfolgreich deployed wurde. Du kannst auch

Log-Daten und Fehlerprotokolle einsehen, um sicherzustellen, dass alles korrekt läuft.

Zusammenfassung

Mit dieser Konfiguration wird die .NET-Anwendung auf **Azure for Students** deployt, wobei GitLab CI/CD für die Automatisierung des Build-, Test- und Deployment-Prozesses genutzt wird. Die Nutzung von GitLab **CI/CD Secrets** stellt sicher, dass sensible Daten sicher verwaltet werden.

Techstack_Architecture.de

Produkt & Projekt / Methoden / Deliberating structures / Werkzeuge & Automatisierung

Architektur & Techstack von Webanwendungen



A Quick Guide to the Right SaaS Tech Stacks for Decision-Makers - Binmile

SaaS Tech Stack: A Comprehensive Guide for Decision Makers - Linearloop

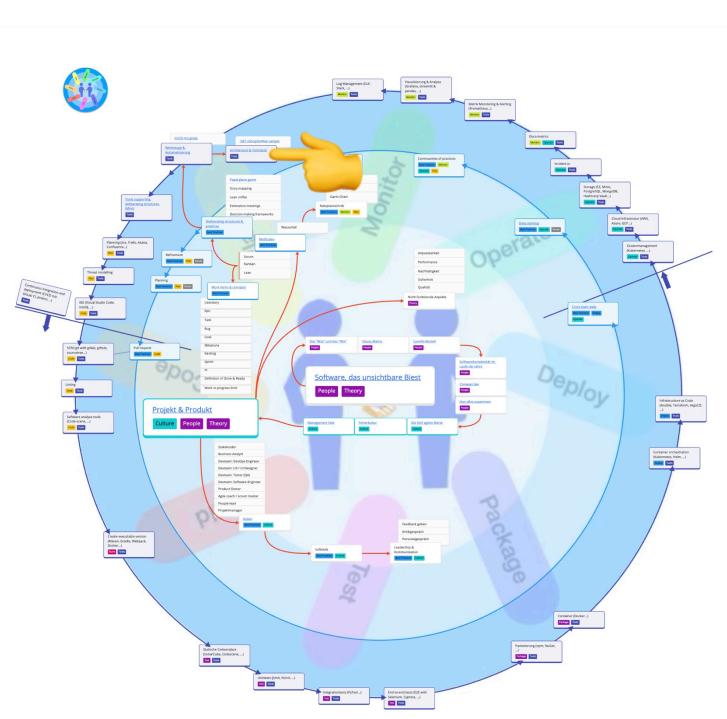
Best SaaS Tech Stack: Tools You Need to Win in 2024 - ScaleupAlly

Best Tech Stack for SaaS Development in 2024 - Brights Blog

How to Choose the Best Tech Stack for Your SaaS - Imaginary Cloud

What Is SaaS Architecture? 10 Best Practices In 2024 - CloudZero

How to Choose the Best Tech Stack for SaaS Development? - MYCPLUS



[SaaS Tech Stack: Choosing the Best for Your Project - CognitiveClouds](#)
[The Architecture Advantage: Choosing the Right Tech Stack for Your SaaS App - DEV Community](#)
[The Perfect SaaS Tech Stack Every CEO, CTO, and VP Should Know - ClickIT](#)

Architekturen

Im Kontext von Software bezeichnet Architektur die grundlegende Struktur und Organisation eines Softwaresystems. Sie umfasst Entscheidungen über Komponenten, deren Beziehungen zueinander, die Kommunikationswege und die übergreifenden Entwurfsmuster. Die Softwarearchitektur legt fest, wie die einzelnen Teile des Systems interagieren, welche Technologien und Frameworks eingesetzt werden, und stellt sicher, dass das System wartbar, erweiterbar und effizient bleibt. Architekturentscheidungen berücksichtigen häufig auch nicht-funktionale Anforderungen wie Skalierbarkeit, Sicherheit, Zuverlässigkeit und Performance.

Hier ist eine kurze Zusammenfassung der Architekturen für Webanwendungen:

1. Monolithische Architektur

- **Beispiel:** Traditionelles E-Commerce-System.
- **Vorteile:** Einfachheit, geringere Infrastruktukosten.
- **Nachteile:** Schwer skalierbar, langsame Entwicklung.

2. Microservices-Architektur

- **Beispiel:** Social-Media-Netzwerk mit separaten Diensten.
- **Vorteile:** Hohe Skalierbarkeit, Flexibilität, Resilienz.
- **Nachteile:** Hohe Komplexität, Orchestrierung nötig, Datenkonsistenz schwierig.

3. Serverless-Architektur

- **Beispiel:** Echtzeit-Datenverarbeitung (z. B. AWS Lambda).
- **Vorteile:** Kosteneffizient, automatische Skalierung, schnelle Entwicklung.
- **Nachteile:** Abhängigkeit von Cloud-Anbietern, Latenzzeiten (Cold Starts).

4. Single-Page Application (SPA)

- **Beispiel:** Projektmanagement-Tool (z. B. Trello).
- **Vorteile:** Schnelle UX, offline-fähig, leicht auf Mobile übertragbar.
- **Nachteile:** SEO-Herausforderungen, längere Initial-Ladezeit.

5. Progressive Web Application (PWA)

- **Beispiel:** Twitter Lite.
- **Vorteile:** Plattformunabhängig, offline-fähig, kosteneffizient.
- **Nachteile:** Eingeschränkter API-Zugriff, Unterschiede je nach Browser.

6. Jamstack-Architektur

- **Beispiel:** Statischer Blog mit dynamischen Inhalten über API.
- **Vorteile:** Schnelle Ladezeiten, sicher, gut skalierbar.
- **Nachteile:** Weniger dynamisch, lange Build-Zeiten.

7. Multi-Page Application (MPA)

- **Beispiel:** CMS-Website wie WordPress.
- **Vorteile:** SEO-freundlich, einfache Navigation, hohe Browser-Kompatibilität.
- **Nachteile:** Langsame Benutzererfahrung, hohe Serverlast.

Die Wahl der Architektur hängt von den spezifischen Anforderungen und Einschränkungen der Anwendung ab.

Es gibt verschiedene Architekturansätze für Webanwendungen, die sich in Struktur, Funktionalität und Entwicklungsansatz unterscheiden. Jede Architektur hat ihre eigenen Vor- und Nachteile und eignet sich je nach Anwendungsfall unterschiedlich gut. Hier sind die wichtigsten Architekturen von Webanwendungen:

1. Monolithische Architektur

In der monolithischen Architektur sind alle Teile der Anwendung (UI, Business-Logik, Datenbank) in einem einzigen, zusammenhängenden Codebasis vereint und werden als ein einziges Deployment-Paket bereitgestellt.

Beispiel

Ein traditionelles E-Commerce-System, bei dem das gesamte System in einer einzigen Anwendung läuft und von einem einzigen Server bereitgestellt wird.

Vorteile

- **Einfachheit:** Einfach zu entwickeln, testen und bereitstellen, da nur eine einzige Anwendung verwaltet wird.
- **Geringere Komplexität:** Alle Module sind in einem einzigen Code-Repository, was die Verwaltung und das Debuggen vereinfacht.
- **Geringere Infrastrukturkosten:** Da alle Komponenten auf einem Server laufen, werden oft weniger Ressourcen benötigt.

Nachteile

- **Skalierbarkeit:** Schwer skalierbar, da die gesamte Anwendung repliziert werden muss, selbst wenn nur ein Teil (z. B. die Datenbank) mehr Ressourcen benötigt.
 - **Deployment:** Ein Fehler in einem Teil der Anwendung kann die gesamte Anwendung lahmlegen.
 - **Langsame Entwicklung:** Änderungen an einem Modul können unbeabsichtigte Auswirkungen auf andere Module haben, was zu einer langsameren Entwicklung führen kann.
-

2. Microservices-Architektur

In einer Microservices-Architektur wird die Anwendung in unabhängige, lose gekoppelte Dienste aufgeteilt, die separat entwickelt, bereitgestellt und skaliert werden können. Jeder Service hat in der Regel seine eigene Datenbank und ist für eine bestimmte Funktion verantwortlich.

Beispiel

Ein Social-Media-Netzwerk, bei dem die Dienste für Nutzerprofile, Posts, Benachrichtigungen und Suchfunktion separat entwickelt und verwaltet werden.

Vorteile

- **Skalierbarkeit:** Jeder Microservice kann unabhängig skaliert werden, basierend auf seinen Anforderungen.
- **Flexibilität:** Verschiedene Technologien können für verschiedene Services verwendet werden, die am besten für deren Bedürfnisse geeignet sind.
- **Resilienz:** Ein Fehler in einem Microservice beeinträchtigt nicht notwendigerweise die gesamte Anwendung.

Nachteile

- **Komplexität:** Erhöhte Komplexität bei der Entwicklung und Verwaltung von mehreren Services, einschließlich der Notwendigkeit eines guten Kommunikationsmechanismus.
 - **Deployment:** Mehrere Services bedeuten oft, dass eine umfassendere Orchestrierung wie Kubernetes oder Docker Swarm notwendig ist.
 - **Datenkonsistenz:** Herausforderungen bei der Aufrechterhaltung der Datenkonsistenz über verschiedene Datenbanken hinweg.
-

3. Serverless-Architektur

In einer Serverless-Architektur wird die Infrastruktur von einem Cloud-Anbieter verwaltet, und der Entwickler konzentriert sich auf das Schreiben der Geschäftslogik. Anwendungen bestehen aus Funktionen, die nur ausgeführt

werden, wenn sie benötigt werden (Event-Driven-Architecture).

Beispiel

Ein System zur Echtzeit-Datenverarbeitung, das auf AWS Lambda basiert und bei bestimmten Ereignissen (z. B. wenn eine Datei hochgeladen wird) Funktionen ausführt.

Vorteile

- **Kosteneffizienz:** Abrechnung erfolgt nur für die tatsächliche Ausführungszeit, was Kosten spart.
- **Skalierbarkeit:** Automatische Skalierung, da der Anbieter für die Verwaltung der Infrastruktur sorgt.
- **Schnelle Entwicklung:** Der Fokus liegt nur auf der Geschäftslogik, wodurch Entwicklungszeit gespart wird.

Nachteile

- **Abhängigkeit von Cloud-Anbietern:** Bindung an bestimmte Cloud-Anbieter und deren API-Struktur.
 - **Eingeschränkte Kontrolle:** Weniger Kontrolle über die Infrastruktur und begrenzte Anpassungsmöglichkeiten.
 - **Latenzzeit:** Bei nicht häufig genutzten Funktionen kann es eine kleine Verzögerung geben, die als „Cold Start“ bezeichnet wird.
-

4. Single-Page Application (SPA)

SPAs laden die gesamte Anwendung in einem einzigen HTML-Dokument, und die Navigation erfolgt durch dynamische Aktualisierungen von Teilbereichen der Seite. SPAs kommunizieren oft über eine API mit dem Backend und nutzen Technologien wie AJAX oder WebSockets.

Beispiel

Ein Projektmanagement-Tool wie Trello oder Asana, das in einer einzigen Seite läuft und verschiedene Komponenten (z. B. Boards, Listen, Karten) dynamisch nachlädt.

Vorteile

- **Benutzererfahrung:** Reibungslose, schnelle Benutzererfahrung ohne ständige Nachladen der Seite.
- **Offline-Fähigkeit:** SPAs können mit lokalen Caches und Service-Worker-Technologien auch offline funktionieren.
- **Mobile Anwendungen:** Einfacher Übergang zu mobilen Plattformen, da die Logik im Frontend zentriert ist.

Nachteile

- **SEO:** Schwieriger zu optimieren für Suchmaschinen, da der Inhalt dynamisch geladen wird.
 - **Ladezeit:** Initiale Ladezeit kann höher sein, da die gesamte Anwendung geladen wird.
 - **Browser-Kompatibilität:** Komplexe SPAs können je nach Browser Unterschiede im Verhalten aufweisen.
-

5. Progressive Web Application (PWA)

PWAs kombinieren die Vorteile von SPAs mit mobilen Anwendungen. Sie laufen im Browser, können aber wie native Apps auf dem Smartphone installiert und offline verwendet werden.

Beispiel

Twitter Lite, eine PWA-Version von Twitter, die im Browser oder als App genutzt werden kann und offline funktioniert.

Vorteile

- **Plattformunabhängigkeit:** PWAs laufen in jedem modernen Browser und funktionieren auf mobilen Geräten wie eine native App.
- **Offline-Fähigkeit:** Durch Nutzung von Service Workern können PWAs Inhalte auch offline bereitstellen.
- **Kostensparnis:** Einmalige Entwicklung für alle Plattformen anstelle von separaten mobilen und Web-Versionen.

Nachteile

- **Eingeschränkter Zugriff auf Geräte-APIs:** Nicht alle Funktionen (z. B. Hardwarezugriff) sind auf allen Geräten oder Browsern möglich.
 - **Browser-Abhängigkeit:** Unterschiedliche Browser können PWAs unterschiedlich unterstützen.
 - **SEO und Performance:** Ähnlich wie bei SPAs können auch PWAs Herausforderungen in der SEO und Performance haben.
-

6. Jamstack-Architektur

Jamstack basiert auf der Trennung von Frontend und Backend, wobei der Frontend-Code statisch generiert wird und Inhalte über APIs und Microservices eingebunden werden. Die Buchstaben „JAM“ stehen für JavaScript, APIs und Markup.

Beispiel

Ein Blog oder Content-Website, die auf statischen Seiten basiert und nur für dynamische Inhalte API-Aufrufe durchführt, z. B. mit Next.js oder Gatsby.

Vorteile

- **Performance:** Sehr schnell, da statische Seiten direkt vom CDN bereitgestellt werden.
- **Sicherheit:** Geringere Angriffsfläche, da kein Backend direkt verfügbar ist.
- **Skalierbarkeit:** Automatische Skalierbarkeit, da nur statische Dateien ausgeliefert werden.

Nachteile

- **Einschränkungen bei dynamischen Inhalten:** Für häufige Updates oder sehr dynamische Inhalte ist Jamstack weniger geeignet.
 - **Build-Zeit:** Bei großen Websites kann die Build-Zeit, um neue Inhalte zu generieren, hoch sein.
 - **Komplexität bei hybriden Ansätzen:** Die Kombination von statischen und dynamischen Inhalten kann zusätzliche Komplexität erfordern.
-

7. Multi-Page Application (MPA)

Eine Multi-Page Application lädt für jede Seite eine neue HTML-Datei. Jede Interaktion, die eine andere Seite benötigt, führt zu einem vollständigen Neuladen der Seite.

Beispiel

Traditionelle CMS-basierte Websites wie WordPress-Seiten, bei denen jede Seite separat geladen wird.

Vorteile

- **SEO-freundlich:** Jede Seite hat eine eigene URL und kann leicht für Suchmaschinen optimiert werden.
- **Einfache Navigation:** Benutzer können mithilfe der URL gezielt auf jede Seite zugreifen.
- **Browser-Kompatibilität:** Standardstruktur für Webanwendungen, die in allen Browsern gut funktioniert.

Nachteile

- **Langsamere Benutzererfahrung:** Da die Seite für jede Navigation neu geladen wird, kann die Benutzererfahrung langsamer sein.
- **Erhöhter Server-Aufwand:** Jede Anfrage erfordert einen neuen Seitenaufruf und kann die Serverlast erhöhen.
- **Komplexere Benutzerinteraktionen:** Dynamische und interaktive

Jede Architektur hat ihre Stärken und Schwächen, und die Wahl hängt von den spezifischen Anforderungen der Anwendung, dem erwarteten Nutzerverhalten, den Budget- und Zeitvorgaben ab.

Beliebte Techstacks

Ein *Techstack* im Software-Kontext bezeichnet die Kombination von Technologien, Programmiersprachen, Frameworks, Tools und Diensten, die verwendet werden, um eine Softwareanwendung oder -plattform zu entwickeln, zu betreiben und zu verwalten. Er setzt sich meist aus dem Backend (Server-seitige Logik, z.B. Node.js, Python, .NET), dem Frontend (Client-seitige Darstellung, z.B. React, Angular, Vue.js) und der Datenbank (z.B. PostgreSQL, MongoDB) zusammen. Der Techstack gibt Aufschluss darüber, welche Technologien zusammenarbeiten, um die Funktionalität, Performance und Skalierbarkeit der Software sicherzustellen.

Hier ist eine kurze, stichpunktartige Zusammenfassung der aktuellen beliebten Tech-Stacks:

1. MEAN Stack (MongoDB, Express.js, Angular, Node.js)

- **Backend:** Node.js, Express.js
- **Frontend:** Angular
- **Datenbank:** MongoDB (NoSQL)
- **Vorteile:** Einheitliche Programmiersprache (JavaScript), moderne Architektur, hohe Skalierbarkeit
- **Einsatz:** Echtzeitkommunikation, Single-Page-Anwendungen, APIs

2. MERN Stack (MongoDB, Express.js, React, Node.js)

- **Backend:** Node.js, Express.js
- **Frontend:** React
- **Datenbank:** MongoDB (NoSQL)
- **Vorteile:** Flexibilität bei der UI, Wiederverwendbare Komponenten, schnelle Datenaktualisierung
- **Einsatz:** Social Media, Dashboards, interaktive UIs

3. LAMP Stack (Linux, Apache, MySQL, PHP)

- **Backend:** PHP
- **Webserver:** Apache
- **Datenbank:** MySQL (relational)
- **Betriebssystem:** Linux

- **Vorteile:** Stabilität, Open-Source, bewährte Lösung
- **Einsatz:** CMS, Blogs, Unternehmenswebsites

4. JAMstack (JavaScript, APIs, Markup)

- **Frontend:** JavaScript
- **Backend:** APIs (serverlos)
- **Markup:** Statische Seiten
- **Vorteile:** Schnelle Ladezeiten, hohe Skalierbarkeit, bessere Sicherheit
- **Einsatz:** Blogs, Marketingseiten, Portfolio-Seiten

5. .NET Stack (C#, ASP.NET Core, Azure)

- **Backend:** ASP.NET Core
- **Programmiersprache:** C#
- **Cloud:** Azure
- **Vorteile:** Hohe Performance, Cross-Platform, Sicherheitsfunktionen
- **Einsatz:** Unternehmensanwendungen, APIs, komplexe Backend-Systeme

Fazit:

- **MERN/MEAN:** Für moderne Webanwendungen mit JavaScript.
- **JAMstack:** Für schnelle, skalierbare, statische Websites.
- **LAMP:** Für klassische Webanwendungen und Content-Management-Systeme.
- **.NET:** Für robuste, skalierbare Unternehmenslösungen.

Die Wahl des Tech-Stacks ist eine der wichtigsten Entscheidungen bei der Softwareentwicklung, da sie direkten Einfluss auf die Skalierbarkeit, Wartbarkeit, Sicherheit und Performance einer Anwendung hat. Hier ist eine detailliertere Erklärung der aktuell beliebtesten Tech-Stacks und ihrer Anwendungsfälle:

1. MEAN Stack (MongoDB, Express.js, Angular, Node.js)

- **Beschreibung:**
 - **MongoDB:** Eine NoSQL-Datenbank, die sich durch Flexibilität und Skalierbarkeit auszeichnet. Sie speichert Daten in JSON-ähnlichen Dokumenten und eignet sich hervorragend für dynamische Datenmodelle.
 - **Express.js:** Ein minimalistisches und flexibles Web-Framework für Node.js, das eine einfache Verwaltung von Routen und Middleware ermöglicht.
 - **Angular:** Ein Frontend-Framework von Google, das eine strukturierte Lösung für die Erstellung von Single-Page-Anwendungen (SPAs) bietet. Es verwendet TypeScript und ist bekannt für seine Zwei-Wege-Datenbindung und umfangreiche Entwickler-Tools.
 - **Node.js:** Ein JavaScript-basiertes Backend-Framework, das serverseitige JavaScript-Anwendungen ausführt und aufgrund seiner hohen

Performance und Skalierbarkeit oft für Echtzeitanwendungen und APIs verwendet wird.

- **Vorteile:**
 - **Einheitliche Sprache:** Da sowohl das Frontend als auch das Backend in JavaScript entwickelt werden, müssen Entwickler nur eine Sprache beherrschen, was die Lernkurve reduziert.
 - **Moderne Architektur:** Der Stack ist ideal für moderne Webanwendungen und SPAs, bei denen schnelle Interaktivität und Datenaktualisierungen erforderlich sind.
 - **Skalierbarkeit:** Besonders geeignet für Anwendungen, die schnell skalieren müssen, da Node.js eine hohe Performance in stark parallelen Prozessen bietet.
- **Einsatzgebiete:** Echtzeitkommunikation (Chats, Benachrichtigungen), Single-Page-Anwendungen (SPAs), RESTful APIs.

2. MERN Stack (MongoDB, Express.js, React, Node.js)

- **Beschreibung:**
 - **MongoDB:** Siehe oben.
 - **Express.js:** Siehe oben.
 - **React.js:** Eine JavaScript-Bibliothek von Facebook, die zur Erstellung von Benutzeroberflächen verwendet wird. React ist komponentenbasiert und ermöglicht es Entwicklern, wiederverwendbare UI-Elemente zu erstellen. React ist bekannt für seine Virtual DOM-Technologie, die die Performance von dynamischen Webanwendungen erheblich verbessert.
 - **Node.js:** Siehe oben.
- **Vorteile:**
 - **Flexible UI:** Im Vergleich zu Angular bietet React mehr Flexibilität bei der Gestaltung der Benutzeroberfläche. Entwickler können entscheiden, wie sie die Anwendung strukturieren möchten.
 - **Wiederverwendbare Komponenten:** Die komponentenbasierte Struktur von React erleichtert das Wiederverwenden von Code und spart Entwicklungszeit.
 - **Echtzeit-Datenaktualisierung:** React ermöglicht eine schnelle Reaktion auf Datenänderungen ohne komplettene Seitenaktualisierungen, was es ideal für dynamische Webanwendungen macht.
- **Einsatzgebiete:** Social Media Apps, Dashboard-Anwendungen, interaktive Benutzeroberflächen.

3. LAMP Stack (Linux, Apache, MySQL, PHP)

- **Beschreibung:**
 - **Linux:** Ein Open-Source-Betriebssystem, das als Grundlage für Server- und Webanwendungen dient. Es bietet Stabilität und Sicherheit.

- **Apache:** Ein weit verbreiteter Webserver, der HTTP-Anfragen verarbeitet und mit verschiedenen Programmiersprachen zusammenarbeitet.
- **MySQL:** Eine der beliebtesten relationalen Datenbanken, die für strukturierte Daten und komplexe Abfragen verwendet wird.
- **PHP:** Eine serverseitige Skriptsprache, die hauptsächlich für die Entwicklung von dynamischen Webanwendungen und Content-Management-Systemen (wie WordPress) genutzt wird.
- **Vorteile:**
 - **Stabilität und Sicherheit:** Der LAMP-Stack ist ein bewährtes Setup, das seit vielen Jahren in der Webentwicklung eingesetzt wird. Es bietet Stabilität und viele Tools für die Verwaltung von Webanwendungen.
 - **Kostenlos und Open-Source:** Alle Komponenten des LAMP-Stacks sind Open-Source und kostenlos, was ihn besonders für kleine Unternehmen und Startups attraktiv macht.
 - **Reife und Community:** Es gibt eine riesige Entwicklergemeinschaft, die viele Ressourcen und Tutorials zur Verfügung stellt.
- **Einsatzgebiete:** Content-Management-Systeme (CMS), Blogs, Unternehmenswebsites, Foren.

4. JAMstack (JavaScript, APIs, Markup)

- **Beschreibung:**
 - **JavaScript:** Wie bei anderen modernen Webtechnologien wird JavaScript für die interaktive Benutzeroberfläche und die dynamische Logik auf der Seite verwendet.
 - **APIs:** JAMstack setzt auf serverlose Architekturen, bei denen alle Backend-Operationen über APIs abgewickelt werden. Dies kann mit verschiedenen externen Diensten wie Datenbanken, Authentifizierung und Zahlungsdiensten erfolgen.
 - **Markup:** Statt dynamischer Seiteninhalte werden statische Seiten mit generierten Markup-Dateien ausgeliefert, was zu schnelleren Ladezeiten führt.
- **Vorteile:**
 - **Geschwindigkeit:** Da viele Daten statisch vorgerendert und über APIs geladen werden, sind JAMstack-Anwendungen sehr schnell und bieten eine ausgezeichnete Benutzererfahrung.
 - **Skalierbarkeit:** Da statische Seiten auf Content Delivery Networks (CDNs) ausgeliefert werden, lassen sich JAMstack-Anwendungen sehr leicht skalieren.
 - **Sicherheit:** Weniger serverseitige Logik bedeutet weniger Angriffsflächen.
- **Einsatzgebiete:** Blogs, Marketingseiten, E-Commerce-Seiten, Portfolio-Seiten.

5. .NET Stack (C#, ASP.NET Core, Azure)

- **Beschreibung:**
 - **C#:** Eine objektorientierte Programmiersprache, die von Microsoft entwickelt wurde und sich hervorragend für die Entwicklung von Unternehmenslösungen eignet.
 - **ASP.NET Core:** Ein leistungsstarkes Framework für Webanwendungen, das eine schnelle Entwicklung von APIs und Websites ermöglicht. Es ist plattformübergreifend und kann auf Windows, Linux und macOS betrieben werden.
 - **Azure:** Microsofts Cloud-Plattform, die eine breite Palette von Diensten für Infrastruktur, Datenbanken, Anwendungen und mehr bietet.
- **Vorteile:**
 - **Leistung:** ASP.NET Core ist bekannt für seine hohe Performance und Skalierbarkeit, was es ideal für große Unternehmen und hochperformante Anwendungen macht.
 - **Cross-Platform:** Mit .NET Core können Anwendungen auf verschiedenen Betriebssystemen ausgeführt werden, was die Flexibilität erhöht.
 - **Sicherheit:** Microsoft bietet robuste Sicherheitsfunktionen für die Entwicklung von sicheren Anwendungen.
- **Einsatzgebiete:** Unternehmensanwendungen, SaaS-Produkte, APIs, komplexe Backend-Systeme.

Fazit:

Die Wahl des Tech-Stacks hängt von verschiedenen Faktoren wie Projektanforderungen, Budget, Team-Erfahrung und Skalierbarkeit ab. Aktuell sind insbesondere **MERN** (React.js) und **MEAN** (Angular) für moderne Webanwendungen sehr beliebt, weil sie eine vollständige JavaScript-Lösung für Frontend und Backend bieten. **JAMstack** gewinnt auch an Popularität, insbesondere für Websites, die schnelle Ladezeiten und einfache Skalierbarkeit benötigen. Für komplexe und skalierbare Unternehmenslösungen bleibt der **.NET Stack** eine ausgezeichnete Wahl.

Spielerisch eine Architektur wählen

Hier ist ein Beispiel, das eine spielerische Entscheidungsfindung für eine Webanwendungsarchitektur ermöglicht. Es enthält eine interaktive, Frage-basierte Auswahl, die den Benutzer durch die wichtigsten Merkmale der Architekturen führt und am Ende eine Empfehlung gibt.

```
In [1]: # Architektur-Auswahl-Notebook
# Ziel: Benutzerfreundliche Entscheidungsfindung für eine Webanwendungsa

# Import notwendiger Bibliotheken
from ipywidgets import widgets
from IPython.display import display, Markdown
```

```

# Begrüßung und Einführung
display(Markdown("### Willkommen zur spielerischen Auswahl einer Webanwendung"))
display(Markdown("Beantworte die folgenden Fragen, um eine Architektur zu empfehlen"))

# Eingabefragen
questions = [
    ("Wie wichtig ist die Skalierbarkeit?", ["Sehr wichtig", "Mittelwichtig"]),
    ("Welche Kontrolle benötigst du über die Infrastruktur?", ["Vollständige Kontrolle", "Mittlere Kontrolle"]),
    ("Soll die Anwendung offline funktionieren?", ["Ja", "Nein"]),
    ("Welche Art von Benutzerinteraktion ist vorgesehen?", ["Interaktiv", "Passiv"]),
    ("Wie flexibel muss die Anwendung bezüglich neuer Technologien sein?", ["Flexibel", "Rigid"]),
    ("Ist SEO für die Anwendung entscheidend?", ["Ja", "Nicht so wichtig"])
]

# Antwortverarbeitung
answers = {}

# Fragewidgets erstellen
for i, (question, options) in enumerate(questions):
    print(question)
    answers[i] = widgets.RadioButtons(
        options=options,
        description="",
        disabled=False
    )
    display(answers[i])

# Architekturentscheidungslogik
def recommend_architecture(*args):
    # Antwortwerte sammeln
    user_answers = [answer.value for answer in answers.values()]

    # Empfehlung basierend auf Antworten
    if user_answers == ["Sehr wichtig", "Vollständige Kontrolle", "Ja"]:
        recommendation = "Microservices-Architektur"
    elif user_answers == ["Mittelwichtig", "Mittlere Kontrolle", "Nein"]:
        recommendation = "Single-Page Application (SPA)"
    elif user_answers == ["Sehr wichtig", "Kaum Kontrolle", "Nein", "Wenig Interaktion"]:
        recommendation = "Jamstack-Architektur"
    elif user_answers == ["Sehr wichtig", "Mittlere Kontrolle", "Ja", "Sehr interaktiv"]:
        recommendation = "Progressive Web Application (PWA)"
    elif user_answers == ["Eher unwichtig", "Vollständige Kontrolle", "Nein", "Wenig Interaktion"]:
        recommendation = "Multi-Page Application (MPA)"
    else:
        recommendation = "Monolithische Architektur"

    # Empfehlung ausgeben
    display(Markdown(f"**Empfohlene Architektur:** {recommendation}"))
    display(Markdown(f"Diese Architektur könnte zu deinem Projekt passen"))

# Button zur Entscheidung
decision_button = widgets.Button(description="Entscheidung anzeigen")
decision_button.on_click(recommend_architecture)

# Button anzeigen
display(decision_button)

```

Willkommen zur spielerischen Auswahl einer Webanwendungsarchitektur!

Beantworte die folgenden Fragen, um eine Architektur zu finden, die zu deinem Projekt passt.

Wie wichtig ist die Skalierbarkeit?

RadioButtons(options=('Sehr wichtig', 'Mittelwichtig', 'Eher unwichtig'), value='Sehr wichtig')

Welche Kontrolle benötigst du über die Infrastruktur?

RadioButtons(options=('Vollständige Kontrolle', 'Mittlere Kontrolle', 'Kaum Kontrolle'), value='Vollständige K...')

Soll die Anwendung offline funktionieren?

RadioButtons(options=('Ja', 'Nein'), value='Ja')

Welche Art von Benutzerinteraktion ist vorgesehen?

RadioButtons(options=('Interaktiv mit hoher Leistung', 'Standard', 'Wenig interaktiv'), value='Interaktiv mit ...')

Wie flexibel muss die Anwendung bezüglich neuer Technologien sein?

RadioButtons(options=('Sehr flexibel', 'Mittel', 'Kaum wichtig'), value='Sehr flexibel')

Ist SEO für die Anwendung entscheidend?

RadioButtons(options=('Ja', 'Nicht so wichtig'), value='Ja')

Button(description='Entscheidung anzeigen', style=ButtonStyle())

Empfohlene Architektur: Monolithische Architektur

Diese Architektur könnte zu deinem Projekt passen!

Funktionsweise des Notebooks

1. **Fragen:** Der Benutzer beantwortet mehrere Fragen, die auf den Anforderungen und Zielen der Webanwendung basieren (z. B. Skalierbarkeit, Kontrolle, SEO).
2. **Architekturentscheidungslogik:** Basierend auf den Antworten wird eine Architektur empfohlen.
3. **Ausgabe:** Der Benutzer erhält eine Empfehlung, die den passenden Architekturansatz für das Projekt vorschlägt.

Dieses Notebook kann leicht angepasst werden, um weitere Kriterien und Architekturen hinzuzufügen.

eShopOnWeb.de

eShopOnWeb sample

Das Projekt eShopOnWeb ist eine Beispiel-Webanwendung, die als Referenz für die Entwicklung von E-Commerce-Anwendungen dient. Es nutzt .NET Core und demonstriert eine einfache Architektur mit verschiedenen modernen Webtechnologien. Die Anwendung enthält typische Funktionalitäten wie Produktauswahl, Einkaufswagen und Bestellmanagement und eignet sich

hervorragend, um mehr über den Aufbau von Webanwendungen und die Nutzung von DevOps-Technologien wie Docker und Devcontainers zu lernen.

In diesem Notebook werden wir Schritt für Schritt erklären, wie man das eShopOnWeb-Projekt:

- 1. Klonen
- 2. Lokal ausführen
- 3. Im Devcontainer starten
- 4. In Docker ausführen kann.
- 5. Auf Azure ausrollen kann.

Hier ist eine ausführliche Beschreibung der Schritte, wie du das eShopOnWeb-Projekt lokal, im Devcontainer, in Docker und auf Azure ausrollen kannst:

1. Repository Klonen

Zunächst musst du das eShopOnWeb-Repository von GitHub auf deinen lokalen Rechner klonen. Dies geht einfach über den Git-Befehl:

```
In [1]: !git clone https://github.com/NimblePros/eShopOnWeb.git
```

```
Cloning into 'eShopOnWeb'...
remote: Enumerating objects: 9456, done.
remote: Counting objects: 100% (1104/1104), done.
remote: Compressing objects: 100% (531/531), done.
remote: Total 9456 (delta 603), reused 1020 (delta 555), pack-reused 835
2 (from 1)
Receiving objects: 100% (9456/9456), 62.29 MiB | 8.93 MiB/s, done.
Resolving deltas: 100% (5993/5993), done.
```

Dieser Schritt holt das Projekt in deinen lokalen Ordner.

2. Lokales Starten des Projekts

Um das Projekt lokal auszuführen, musst du sicherstellen, dass du das .NET SDK installiert hast. Falls du es noch nicht hast, kannst du es von der offiziellen .NET-Website herunterladen:

[Download .NET SDK](#).

Um den beschriebenen DIFF in deinem Projekt manuell umzusetzen, solltest du folgende Schritte durchführen:

1. Änderungen an der `docker-compose.yml`:

- **Füge das ^M -Zeichen nach bestimmten Zeilen hinzu** (dies könnte ein Zeilenumbruchproblem sein, das du eventuell in einem bestimmten Editor siehst). Beachte, dass dieses Zeichen auf Windows-basierten Systemen auftreten kann.

- Ändere die `SA_PASSWORD` Umgebungsvariable:
 - Ändere `SA_PASSWORD=@someThingComplicated1234` zu `MSSQL_SA_PASSWORD=Some$Complex1234Password`.

Beispieländerung:

```
services:
  sqlserver:
    image: mcr.microsoft.com/azure-sql-edge
    ports:
      - "1433:1433"
    environment:
      - MSSQL_SA_PASSWORD=Some$Complex1234Password
      - ACCEPT_EULA=Y
```

2. Änderungen an der `src/PublicApi/Dockerfile`:

- Ändere die Basis- und Build-Image-Version von .NET 8.0 auf .NET 9.0. Ändere die Zeilen:
 - `FROM mcr.microsoft.com/dotnet/aspnet:8.0 AS base` zu `FROM mcr.microsoft.com/dotnet/aspnet:9.0 AS base`
 - `FROM mcr.microsoft.com/dotnet/sdk:8.0 AS build` zu `FROM mcr.microsoft.com/dotnet/sdk:9.0 AS build`

Beispieländerung:

```
FROM mcr.microsoft.com/dotnet/aspnet:9.0 AS base
FROM mcr.microsoft.com/dotnet/sdk:9.0 AS build
```

3. Änderungen an der `src/Web/Dockerfile`:

- Ändere ebenfalls die .NET-Versionen in der `Dockerfile` von Version 8.0 auf 9.0:
 - `FROM mcr.microsoft.com/dotnet/sdk:8.0 AS build` zu `FROM mcr.microsoft.com/dotnet/sdk:9.0 AS build`
 - `FROM mcr.microsoft.com/dotnet/aspnet:8.0 AS runtime` zu `FROM mcr.microsoft.com/dotnet/aspnet:9.0 AS runtime`

Beispieländerung:

```
FROM mcr.microsoft.com/dotnet/sdk:9.0 AS build
FROM mcr.microsoft.com/dotnet/aspnet:9.0 AS runtime
```

4. Änderungen an der `src/Web/appsettings.json`:

- Füge einen neuen Abschnitt für die `UseOnlyInMemoryDatabase` Konfiguration hinzu:

Beispieländerung:

```
{  
  "ConnectionStrings": {  
    "DefaultConnection": "..."  
  },  
  "Logging": {  
    "LogLevel": {  
      "Default": "Information",  
      "System": "Warning"  
    }  
  },  
  "AllowedHosts": "*",  
  "UseOnlyInMemoryDatabase": true  
}
```

Zusammenfassung:

- Aktualisiere die `.NET`-Version in den Dockerfiles auf `9.0` statt `8.0`.
- Ändere die Umgebungsvariable für das SQL-Server-Passwort in der `docker-compose.yml`.
- Füge eine neue Konfiguration `UseOnlyInMemoryDatabase` in der `appsettings.json` hinzu.

Das sind die manuellen Änderungen, die du vornehmen musst, um den gewünschten DIFF zu erreichen.

2.2. Abhängigkeiten wiederherstellen

Nach dem Klonen des Projekts musst du zunächst alle Abhängigkeiten wiederherstellen:

```
In [38...]: !dotnet restore eShopOnWeb/eShopOnWeb.sln > /dev/null
```

=

Dieser Befehl lädt alle notwendigen NuGet-Pakete herunter, die das Projekt benötigt.

2.3. Projekt bauen

Nun kannst du das Projekt bauen, um sicherzustellen, dass keine Fehler vorhanden sind:

```
In [39...]: !dotnet build eShopOnWeb/eShopOnWeb.sln > /dev/null
```

= =

2.4. Projekt ausführen

Auf einem Windows basierten System kannst du das Projekt nun lokal starten:

1. Starte das Backend:

Öffne einen Terminal und führe folgende Befehle aus:

```
cd notebooks/samples  
dotnet run --project eShopOnWeb/src/PublicApi
```

Die API sollte nun unter `http://localhost:5099/api` im Browser erreichbar sein.

2. Starte nun das Frontend:

Öffne einen weiteren Terminal und führe folgende Befehle aus:

```
cd notebooks/samples  
dotnet run --project eShopOnWeb/src/Web
```

Das Projekt sollte nun unter `http://localhost:5001` im Browser verfügbar sein.

3. Devcontainer Starten

Ein Devcontainer ist eine Docker-basierte Entwicklungsumgebung, die in Visual Studio Code (VSCode) verwendet werden kann. Um das Projekt in einem Devcontainer zu starten, folge diesen Schritten:

3.1. VSCode Devcontainer Erweiterung installieren

Stelle sicher, dass du die **Dev Containers**-Erweiterung in VSCode installiert hast. Dies ermöglicht es dir, das Projekt direkt im Container zu entwickeln.

3.2. Projekt im Devcontainer öffnen

- Öffne das geklonte `eShopOnWeb` -Projekt in VSCode.
- Wähle in VSCode die Option **Reopen in Container**, um das Projekt im Devcontainer zu starten.

VSCode baut dann den Container und stellt sicher, dass die Entwicklungsumgebung genau der entspricht, die im Projekt konfiguriert ist. Der Devcontainer enthält alle benötigten Tools und Abhängigkeiten.

4. Docker Starten

Falls du das Projekt in einem Docker-Container ausführen möchtest, kannst du dies tun, indem du ein Docker-Image erstellst und es im Container ausführst.

4.1. Docker installieren

Wenn Docker noch nicht auf deinem Computer installiert ist, kannst du es von der offiziellen Website herunterladen:

[Download Docker.](#)

4.2. Docker-Image bauen

Im Projektordner kannst du mit folgendem Befehl ein Docker-Image erstellen:

```
In [40]: !docker compose -f ./eShopOnWeb/docker-compose.yml build > /dev/null
```

WARN[0000] The "Complex1234Password" variable is not set. Defaulting to a blank string.
WARN[0000] /Users/starkj/Documents/2hands/Soproming/Repo/main/notebooks/samples/eShopOnWeb/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion

4.3. Docker-Container starten

Sobald das Image gebaut ist, kannst du den Container starten:

```
In [ ]: !docker compose -f ./eShopOnWeb/docker-compose.yml up > /dev/null
```

WARN[0000] The "Complex1234Password" variable is not set. Defaulting to a blank string.
WARN[0000] /Users/starkj/Documents/2hands/Soproming/Repo/main/notebooks/samples/eShopOnWeb/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 0/0
 ⚡ sqlserver Pulling
0.1s
[+] Running 0/1
 ⚡ sqlserver Pulling
0.2s
[+] Running 0/1
 ⚡ sqlserver Pulling
0.3s
[+] Running 0/1
 ⚡ sqlserver Pulling
0.4s
[+] Running 0/1
 ⚡ sqlserver Pulling
0.5s
[+] Running 0/1
 ⚡ sqlserver Pulling
0.6s
[+] Running 0/1
 ⚡ sqlserver Pulling
0.7s
[+] Running 0/1
 ⚡ sqlserver Pulling
0.8s
[+] Running 0/1
 ⚡ sqlserver Pulling
0.9s
[+] Running 0/1
 ⚡ sqlserver Pulling
1.0s
[+] Running 0/1

```
  ∴ sqlserver Pulling
1.1s
[+] Running 0/1
  ∴ sqlserver [    ] Pulling
1.2s
  ∴ 3d29464607d8 Pulling fs layer
0.0s
  ∴ 3374dd749958 Pulling fs layer
0.0s
  ∴ 18952140cc7b Pulling fs layer
0.0s
[+] Running 0/4
  ∴ sqlserver [    ] Pulling
1.3s
  ∴ 3d29464607d8 Pulling fs layer
0.1s
  ∴ 3374dd749958 Pulling fs layer
0.1s
  ∴ 18952140cc7b Pulling fs layer
0.1s
[+] Running 0/4
  ∴ sqlserver [    ] Pulling
1.4s
  ∴ 3d29464607d8 Pulling fs layer
0.2s
  ∴ 3374dd749958 Pulling fs layer
0.2s
  ∴ 18952140cc7b Pulling fs layer
0.2s
[+] Running 0/4
  ∴ sqlserver [    ] Pulling
1.5s
  ∴ 3d29464607d8 Pulling fs layer
0.3s
  ∴ 3374dd749958 Pulling fs layer
0.3s
  ∴ 18952140cc7b Pulling fs layer
0.3s
[+] Running 0/4
  ∴ sqlserver [    ] Pulling
1.6s
  ∴ 3d29464607d8 Pulling fs layer
0.4s
  ∴ 3374dd749958 Pulling fs layer
0.4s
  ∴ 18952140cc7b Pulling fs layer
0.4s
[+] Running 0/4
  ∴ sqlserver [    ] 1.404MB / 578.2MB Pulling
1.7s
  ∴ 3d29464607d8 Downloading 325.9kB/31.58MB
0.5s
  ∴ 3374dd749958 Downloading 538.9kB/475.4MB
0.5s
  ∴ 18952140cc7b Downloading 538.9kB/71.23MB
0.5s
[+] Running 0/4
```

```
: sqlserver [ ] 2.272MB / 578.2MB Pulling
1.8s
  : 3d29464607d8 Downloading 653.6kB/31.58MB
0.6s
  : 3374dd749958 Downloading 1.08MB/475.4MB
0.6s
  : 18952140cc7b Downloading 538.9kB/71.23MB
0.6s
[+] Running 0/4
  ↵ sqlserver [ ] 2.813MB / 578.2MB Pulling
1.9s
  : 3d29464607d8 Downloading 653.6kB/31.58MB
0.7s
  : 3374dd749958 Downloading 1.08MB/475.4MB
0.7s
  : 18952140cc7b Downloading 1.08MB/71.23MB
0.7s
[+] Running 0/4
  ↵ sqlserver [ ] 4.549MB / 578.2MB Pulling
2.0s
  : 3d29464607d8 Downloading 1.309MB/31.58MB
0.8s
  : 3374dd749958 Downloading 1.62MB/475.4MB
0.8s
  : 18952140cc7b Downloading 1.62MB/71.23MB
0.8s
[+] Running 0/4
  ↵ sqlserver [ ] 5.205MB / 578.2MB Pulling
2.1s
  ↵ 3d29464607d8 Downloading 1.964MB/31.58MB
0.9s
  ↵ 3374dd749958 Downloading 1.62MB/475.4MB
0.9s
  ↵ 18952140cc7b Downloading 1.62MB/71.23MB
0.9s
[+] Running 0/4
  ↵ sqlserver [ ] 7.367MB / 578.2MB Pulling
2.2s
  ↵ 3d29464607d8 Downloading 1.964MB/31.58MB
1.0s
  ↵ 3374dd749958 Downloading 2.702MB/475.4MB
1.0s
  ↵ 18952140cc7b Downloading 2.702MB/71.23MB
1.0s
[+] Running 0/4
  ↵ sqlserver [ ] 9.104MB / 578.2MB Pulling
2.3s
  ↵ 3d29464607d8 Downloading 2.62MB/31.58MB
1.1s
  ↵ 3374dd749958 Downloading 3.242MB/475.4MB
1.1s
  ↵ 18952140cc7b Downloading 3.242MB/71.23MB
1.1s
[+] Running 0/4
  ↵ sqlserver [ ] 10.19MB / 578.2MB Pulling
2.4s
  ↵ 3d29464607d8 Downloading 2.62MB/31.58MB
```

```
1.2s
  ⌈ 3374dd749958 Downloading  3.783MB/475.4MB
1.2s
  ⌈ 18952140cc7b Downloading  3.783MB/71.23MB
1.2s
[+] Running 0/4
  ⌈ sqlserver [    ] 11.38MB / 578.2MB Pulling
2.5s
  ⌈ 3d29464607d8 Downloading  3.275MB/31.58MB
1.3s
  ⌈ 3374dd749958 Downloading  4.324MB/475.4MB
1.3s
  ⌈ 18952140cc7b Downloading  3.783MB/71.23MB
1.3s
[+] Running 0/4
  ⌈ sqlserver [    ] 13.12MB / 578.2MB Pulling
2.6s
  ⌈ 3d29464607d8 Downloading  3.93MB/31.58MB
1.4s
  ⌈ 3374dd749958 Downloading  4.864MB/475.4MB
1.4s
  ⌈ 18952140cc7b Downloading  4.324MB/71.23MB
1.4s
[+] Running 0/4
  ⌈ sqlserver [    ] 13.12MB / 578.2MB Pulling
2.7s
  ⌈ 3d29464607d8 Downloading  3.93MB/31.58MB
1.5s
  ⌈ 3374dd749958 Downloading  4.864MB/475.4MB
1.5s
  ⌈ 18952140cc7b Downloading  4.324MB/71.23MB
1.5s
[+] Running 0/4
  ⌈ sqlserver [..] 15.4MB / 578.2MB Pulling
2.8s
  ⌈ 3d29464607d8 Downloading  4.586MB/31.58MB
1.6s
  ⌈ 3374dd749958 Downloading  5.405MB/475.4MB
1.6s
  ⌈ 18952140cc7b Downloading  5.405MB/71.23MB
1.6s
[+] Running 0/4
  ⌈ sqlserver [..] 16.48MB / 578.2MB Pulling
2.9s
  ⌈ 3d29464607d8 Downloading  4.586MB/31.58MB
1.7s
  ⌈ 3374dd749958 Downloading  5.946MB/475.4MB
1.7s
  ⌈ 18952140cc7b Downloading  5.946MB/71.23MB
1.7s
[+] Running 0/4
  ⌈ sqlserver [..] 18.21MB / 578.2MB Pulling
3.0s
  ⌈ 3d29464607d8 Downloading  5.241MB/31.58MB
1.8s
  ⌈ 3374dd749958 Downloading  6.486MB/475.4MB
1.8s
```

```
: 18952140cc7b Downloading  6.486MB/71.23MB
1.8s
[+] Running 0/4
  ↳ sqlserver [..] 19.3MB / 578.2MB Pulling
3.1s
  ↳ 3d29464607d8 Downloading  5.241MB/31.58MB
1.9s
  ↳ 3374dd749958 Downloading  7.027MB/475.4MB
1.9s
  ↳ 18952140cc7b Downloading  7.027MB/71.23MB
1.9s
[+] Running 0/4
  ↳ sqlserver [..] 21.03MB / 578.2MB Pulling
3.2s
  ↳ 3d29464607d8 Downloading  5.896MB/31.58MB
2.0s
  ↳ 3374dd749958 Downloading  7.568MB/475.4MB
2.0s
  ↳ 18952140cc7b Downloading  7.568MB/71.23MB
2.0s
[+] Running 0/4
  ↳ sqlserver [..] 22.77MB / 578.2MB Pulling
3.3s
  ↳ 3d29464607d8 Downloading  6.552MB/31.58MB
2.1s
  ↳ 3374dd749958 Downloading  8.108MB/475.4MB
2.1s
  ↳ 18952140cc7b Downloading  8.108MB/71.23MB
2.1s
[+] Running 0/4
  ↳ sqlserver [..] 23.29MB / 578.2MB Pulling
3.4s
  ↳ 3d29464607d8 Downloading  6.552MB/31.58MB
2.2s
  ↳ 3374dd749958 Downloading  8.634MB/475.4MB
2.2s
  ↳ 18952140cc7b Downloading  8.108MB/71.23MB
2.2s
[+] Running 0/4
  ↳ sqlserver [..] 25.02MB / 578.2MB Pulling
3.5s
  ↳ 3d29464607d8 Downloading  7.207MB/31.58MB
2.3s
  ↳ 3374dd749958 Downloading  9.175MB/475.4MB
2.3s
  ↳ 18952140cc7b Downloading  8.634MB/71.23MB
2.3s
[+] Running 0/4
  ↳ sqlserver [..] 25.56MB / 578.2MB Pulling
3.6s
  ↳ 3d29464607d8 Downloading  7.207MB/31.58MB
2.4s
  ↳ 3374dd749958 Downloading  9.175MB/475.4MB
2.4s
  ↳ 18952140cc7b Downloading  9.175MB/71.23MB
2.4s
[+] Running 0/4
```

```
  ↳ sqlserver [..] 27.29MB / 578.2MB Pulling
3.7s
    ↳ 3d29464607d8 Downloading  7.863MB/31.58MB
2.5s
    ↳ 3374dd749958 Downloading  9.716MB/475.4MB
2.5s
    ↳ 18952140cc7b Downloading  9.716MB/71.23MB
2.5s
[+] Running 0/4
  ↳ sqlserver [..] 29.03MB / 578.2MB Pulling
3.8s
    ↳ 3d29464607d8 Downloading  8.52MB/31.58MB
2.6s
    ↳ 3374dd749958 Downloading  10.26MB/475.4MB
2.6s
    ↳ 18952140cc7b Downloading  10.26MB/71.23MB
2.6s
[+] Running 0/4
  ↳ sqlserver [..] 30.11MB / 578.2MB Pulling
3.9s
    ↳ 3d29464607d8 Downloading  8.52MB/31.58MB
2.7s
    ↳ 3374dd749958 Downloading  10.8MB/475.4MB
2.7s
    ↳ 18952140cc7b Downloading  10.8MB/71.23MB
2.7s
[+] Running 0/4
  ↳ sqlserver [..] 30.11MB / 578.2MB Pulling
4.0s
    ↳ 3d29464607d8 Downloading  8.52MB/31.58MB
2.8s
    ↳ 3374dd749958 Downloading  10.8MB/475.4MB
2.8s
    ↳ 18952140cc7b Downloading  10.8MB/71.23MB
2.8s
[+] Running 0/4
  ↳ sqlserver [..] 31.85MB / 578.2MB Pulling
4.1s
    ↳ 3d29464607d8 Downloading  9.175MB/31.58MB
2.9s
    ↳ 3374dd749958 Downloading  11.34MB/475.4MB
2.9s
    ↳ 18952140cc7b Downloading  11.34MB/71.23MB
2.9s
[+] Running 0/4
  ↳ sqlserver [..] 33.05MB / 578.2MB Pulling
4.2s
    ↳ 3d29464607d8 Downloading  9.83MB/31.58MB
3.0s
    ↳ 3374dd749958 Downloading  11.34MB/475.4MB
3.0s
    ↳ 18952140cc7b Downloading  11.88MB/71.23MB
3.0s
[+] Running 0/4
  ↳ sqlserver [..] 35.32MB / 578.2MB Pulling
4.3s
    ↳ 3d29464607d8 Downloading  10.49MB/31.58MB
```

```
3.1s
  ⌈ 3374dd749958 Downloading 12.42MB/475.4MB
3.1s
  ⌈ 18952140cc7b Downloading 12.42MB/71.23MB
3.1s
[+] Running 0/4
  ⌈ sqlserver [..] 36.41MB / 578.2MB Pulling
4.4s
  ⌈ 3d29464607d8 Downloading 10.49MB/31.58MB
3.2s
  ⌈ 3374dd749958 Downloading 12.96MB/475.4MB
3.2s
  ⌈ 18952140cc7b Downloading 12.96MB/71.23MB
3.2s
[+] Running 0/4
  ⌈ sqlserver [..] 38.14MB / 578.2MB Pulling
4.5s
  ⌈ 3d29464607d8 Downloading 11.14MB/31.58MB
3.3s
  ⌈ 3374dd749958 Downloading 13.5MB/475.4MB
3.3s
  ⌈ 18952140cc7b Downloading 13.5MB/71.23MB
3.3s
[+] Running 0/4
  ⌈ sqlserver [..] 39.22MB / 578.2MB Pulling
4.6s
  ⌈ 3d29464607d8 Downloading 11.14MB/31.58MB
3.4s
  ⌈ 3374dd749958 Downloading 14.04MB/475.4MB
3.4s
  ⌈ 18952140cc7b Downloading 14.04MB/71.23MB
3.4s
[+] Running 0/4
  ⌈ sqlserver [..] 40.96MB / 578.2MB Pulling
4.7s
  ⌈ 3d29464607d8 Downloading 11.8MB/31.58MB
3.5s
  ⌈ 3374dd749958 Downloading 14.58MB/475.4MB
3.5s
  ⌈ 18952140cc7b Downloading 14.58MB/71.23MB
3.5s
[+] Running 0/4
  ⌈ sqlserver [..] 42.04MB / 578.2MB Pulling
4.8s
  ⌈ 3d29464607d8 Downloading 11.8MB/31.58MB
3.6s
  ⌈ 3374dd749958 Downloading 15.12MB/475.4MB
3.6s
  ⌈ 18952140cc7b Downloading 15.12MB/71.23MB
3.6s
[+] Running 0/4
  ⌈ sqlserver [..] 43.24MB / 578.2MB Pulling
4.9s
  ⌈ 3d29464607d8 Downloading 12.45MB/31.58MB
3.7s
  ⌈ 3374dd749958 Downloading 15.66MB/475.4MB
3.7s
```

```
    ⌈ 18952140cc7b Downloading 15.12MB/71.23MB
3.7s
[+] Running 0/4
  ⌈ sqlserver [.. ..] 44.43MB / 578.2MB Pulling
5.0s
  ⌈ 3d29464607d8 Downloading 13.11MB/31.58MB
3.8s
  ⌈ 3374dd749958 Downloading 15.66MB/475.4MB
3.8s
  ⌈ 18952140cc7b Downloading 15.66MB/71.23MB
3.8s
[+] Running 0/4
  ⌈ sqlserver [.. ..] 45.51MB / 578.2MB Pulling
5.1s
  ⌈ 3d29464607d8 Downloading 13.11MB/31.58MB
3.9s
  ⌈ 3374dd749958 Downloading 16.2MB/475.4MB
3.9s
  ⌈ 18952140cc7b Downloading 16.2MB/71.23MB
3.9s
[+] Running 0/4
  ⌈ sqlserver [.. ..] 47.25MB / 578.2MB Pulling
5.2s
  ⌈ 3d29464607d8 Downloading 13.76MB/31.58MB
4.0s
  ⌈ 3374dd749958 Downloading 16.74MB/475.4MB
4.0s
  ⌈ 18952140cc7b Downloading 16.74MB/71.23MB
4.0s
[+] Running 0/4
  ⌈ sqlserver [.. ..] 48.33MB / 578.2MB Pulling
5.3s
  ⌈ 3d29464607d8 Downloading 13.76MB/31.58MB
4.1s
  ⌈ 3374dd749958 Downloading 17.29MB/475.4MB
4.1s
  ⌈ 18952140cc7b Downloading 17.29MB/71.23MB
4.1s
[+] Running 0/4
  ⌈ sqlserver [.. ..] 50.07MB / 578.2MB Pulling
5.4s
  ⌈ 3d29464607d8 Downloading 14.42MB/31.58MB
4.2s
  ⌈ 3374dd749958 Downloading 17.83MB/475.4MB
4.2s
  ⌈ 18952140cc7b Downloading 17.83MB/71.23MB
4.2s
[+] Running 0/4
  ⌈ sqlserver [.. ..] 51.81MB / 578.2MB Pulling
5.5s
  ⌈ 3d29464607d8 Downloading 15.07MB/31.58MB
4.3s
  ⌈ 3374dd749958 Downloading 18.37MB/475.4MB
4.3s
  ⌈ 18952140cc7b Downloading 18.37MB/71.23MB
4.3s
[+] Running 0/4
```

```
  ↳ sqlserver [.. ..] 52.89MB / 578.2MB Pulling
5.6s
  ↳ 3d29464607d8 Downloading 15.07MB/31.58MB
4.4s
  ↳ 3374dd749958 Downloading 18.91MB/475.4MB
4.4s
  ↳ 18952140cc7b Downloading 18.91MB/71.23MB
4.4s
[+] Running 0/4
  ↳ sqlserver [.. ..] 54.08MB / 578.2MB Pulling
5.7s
  ↳ 3d29464607d8 Downloading 15.73MB/31.58MB
4.5s
  ↳ 3374dd749958 Downloading 18.91MB/475.4MB
4.5s
  ↳ 18952140cc7b Downloading 19.45MB/71.23MB
4.5s
[+] Running 0/4
  ↳ sqlserver [.. ..] 54.62MB / 578.2MB Pulling
5.8s
  ↳ 3d29464607d8 Downloading 15.73MB/31.58MB
4.6s
  ↳ 3374dd749958 Downloading 19.45MB/475.4MB
4.6s
  ↳ 18952140cc7b Downloading 19.45MB/71.23MB
4.6s
[+] Running 0/4
  ↳ sqlserver [.. ..] 56.36MB / 578.2MB Pulling
5.9s
  ↳ 3d29464607d8 Downloading 16.38MB/31.58MB
4.7s
  ↳ 3374dd749958 Downloading 19.99MB/475.4MB
4.7s
  ↳ 18952140cc7b Downloading 19.99MB/71.23MB
4.7s
[+] Running 0/4
  ↳ sqlserver [.. ..] 57.44MB / 578.2MB Pulling
6.0s
  ↳ 3d29464607d8 Downloading 16.38MB/31.58MB
4.8s
  ↳ 3374dd749958 Downloading 20.53MB/475.4MB
4.8s
  ↳ 18952140cc7b Downloading 20.53MB/71.23MB
4.8s
[+] Running 0/4
  ↳ sqlserver [.. ..] 59.18MB / 578.2MB Pulling
6.1s
  ↳ 3d29464607d8 Downloading 17.04MB/31.58MB
4.9s
  ↳ 3374dd749958 Downloading 21.07MB/475.4MB
4.9s
  ↳ 18952140cc7b Downloading 21.07MB/71.23MB
4.9s
[+] Running 0/4
  ↳ sqlserver [.. ..] 60.92MB / 578.2MB Pulling
6.2s
  ↳ 3d29464607d8 Downloading 17.69MB/31.58MB
```

```
5.0s
  ⌈ 3374dd749958 Downloading 21.61MB/475.4MB
5.0s
  ⌈ 18952140cc7b Downloading 21.61MB/71.23MB
5.0s
[+] Running 0/4
  ⌈ sqlserver [.. ..] 62MB / 578.2MB Pulling
6.3s
  ⌈ 3d29464607d8 Downloading 17.69MB/31.58MB
5.1s
  ⌈ 3374dd749958 Downloading 22.15MB/475.4MB
5.1s
  ⌈ 18952140cc7b Downloading 22.15MB/71.23MB
5.1s
[+] Running 0/4
  ⌈ sqlserver [.. ..] 63.19MB / 578.2MB Pulling
6.4s
  ⌈ 3d29464607d8 Downloading 18.35MB/31.58MB
5.2s
  ⌈ 3374dd749958 Downloading 22.15MB/475.4MB
5.2s
  ⌈ 18952140cc7b Downloading 22.69MB/71.23MB
5.2s
[+] Running 0/4
  ⌈ sqlserver [.. ..] 63.73MB / 578.2MB Pulling
6.5s
  ⌈ 3d29464607d8 Downloading 18.35MB/31.58MB
5.3s
  ⌈ 3374dd749958 Downloading 22.69MB/475.4MB
5.3s
  ⌈ 18952140cc7b Downloading 22.69MB/71.23MB
5.3s
[+] Running 0/4
  ⌈ sqlserver [.. ..] 65.47MB / 578.2MB Pulling
6.6s
  ⌈ 3d29464607d8 Downloading 19.01MB/31.58MB
5.4s
  ⌈ 3374dd749958 Downloading 23.23MB/475.4MB
5.4s
  ⌈ 18952140cc7b Downloading 23.23MB/71.23MB
5.4s
[+] Running 0/4
  ⌈ sqlserver [.. ..] 67.21MB / 578.2MB Pulling
6.7s
  ⌈ 3d29464607d8 Downloading 19.66MB/31.58MB
5.5s
  ⌈ 3374dd749958 Downloading 23.77MB/475.4MB
5.5s
  ⌈ 18952140cc7b Downloading 23.77MB/71.23MB
5.5s
[+] Running 0/4
  ⌈ sqlserver [.. ..] 67.75MB / 578.2MB Pulling
6.8s
  ⌈ 3d29464607d8 Downloading 19.66MB/31.58MB
5.6s
  ⌈ 3374dd749958 Downloading 23.77MB/475.4MB
5.6s
```

```
    ⌈ 18952140cc7b Downloading 24.31MB/71.23MB
5.6s
[+] Running 0/4
  ⌈ sqlserver [..] 70.03MB / 578.2MB Pulling
6.9s
    ⌈ 3d29464607d8 Downloading 20.32MB/31.58MB
5.7s
    ⌈ 3374dd749958 Downloading 24.85MB/475.4MB
5.7s
    ⌈ 18952140cc7b Downloading 24.85MB/71.23MB
5.7s
[+] Running 0/4
  ⌈ sqlserver [..] 71.76MB / 578.2MB Pulling
7.0s
    ⌈ 3d29464607d8 Downloading 20.97MB/31.58MB
5.8s
    ⌈ 3374dd749958 Downloading 25.4MB/475.4MB
5.8s
    ⌈ 18952140cc7b Downloading 25.4MB/71.23MB
5.8s
[+] Running 0/4
  ⌈ sqlserver [..] 72.3MB / 578.2MB Pulling
7.1s
    ⌈ 3d29464607d8 Downloading 20.97MB/31.58MB
5.9s
    ⌈ 3374dd749958 Downloading 25.4MB/475.4MB
5.9s
    ⌈ 18952140cc7b Downloading 25.94MB/71.23MB
5.9s
[+] Running 0/4
  ⌈ sqlserver [..] 73.5MB / 578.2MB Pulling
7.2s
    ⌈ 3d29464607d8 Downloading 21.63MB/31.58MB
6.0s
    ⌈ 3374dd749958 Downloading 25.94MB/475.4MB
6.0s
    ⌈ 18952140cc7b Downloading 25.94MB/71.23MB
6.0s
[+] Running 0/4
  ⌈ sqlserver [..] 75.56MB / 578.2MB Pulling
7.3s
    ⌈ 3d29464607d8 Downloading 22.61MB/31.58MB
6.1s
    ⌈ 3374dd749958 Downloading 27.02MB/475.4MB
6.1s
    ⌈ 18952140cc7b Downloading 25.94MB/71.23MB
6.1s
[+] Running 0/4
  ⌈ sqlserver [..] 75.56MB / 578.2MB Pulling
7.4s
    ⌈ 3d29464607d8 Downloading 22.61MB/31.58MB
6.2s
    ⌈ 3374dd749958 Downloading 27.02MB/475.4MB
6.2s
    ⌈ 18952140cc7b Downloading 25.94MB/71.23MB
6.2s
[+] Running 0/4
```

```
  ↳ sqlserver [..] 77.84MB / 578.2MB Pulling
7.5s
  ↳ 3d29464607d8 Downloading 23.27MB/31.58MB
6.3s
  ↳ 3374dd749958 Downloading 28.1MB/475.4MB
6.3s
  ↳ 18952140cc7b Downloading 26.48MB/71.23MB
6.3s
[+] Running 0/4
  ↳ sqlserver [..] 79.58MB / 578.2MB Pulling
7.6s
  ↳ 3d29464607d8 Downloading 23.92MB/31.58MB
6.4s
  ↳ 3374dd749958 Downloading 28.64MB/475.4MB
6.4s
  ↳ 18952140cc7b Downloading 27.02MB/71.23MB
6.4s
[+] Running 0/4
  ↳ sqlserver [..] 79.58MB / 578.2MB Pulling
7.7s
  ↳ 3d29464607d8 Downloading 23.92MB/31.58MB
6.5s
  ↳ 3374dd749958 Downloading 28.64MB/475.4MB
6.5s
  ↳ 18952140cc7b Downloading 27.02MB/71.23MB
6.5s
[+] Running 0/4
  ↳ sqlserver [..] 81.31MB / 578.2MB Pulling
7.8s
  ↳ 3d29464607d8 Downloading 24.58MB/31.58MB
6.6s
  ↳ 3374dd749958 Downloading 29.18MB/475.4MB
6.6s
  ↳ 18952140cc7b Downloading 27.56MB/71.23MB
6.6s
[+] Running 0/4
  ↰ sqlserver [..] 83.59MB / 578.2MB Pulling
7.9s
  ↲ 3d29464607d8 Downloading 25.23MB/31.58MB
6.7s
  ↲ 3374dd749958 Downloading 30.26MB/475.4MB
6.7s
  ↲ 18952140cc7b Downloading 28.1MB/71.23MB
6.7s
[+] Running 0/4
  ↰ sqlserver [..] 84.67MB / 578.2MB Pulling
8.0s
  ↲ 3d29464607d8 Downloading 25.23MB/31.58MB
6.8s
  ↲ 3374dd749958 Downloading 30.8MB/475.4MB
6.8s
  ↲ 18952140cc7b Downloading 28.64MB/71.23MB
6.8s
[+] Running 0/4
  ↰ sqlserver [..] 85.87MB / 578.2MB Pulling
8.1s
  ↲ 3d29464607d8 Downloading 25.89MB/31.58MB
```

```
6.9s
  ⌂ 3374dd749958 Downloading  31.34MB/475.4MB
6.9s
  ⌂ 18952140cc7b Downloading  28.64MB/71.23MB
6.9s
[+] Running 0/4
  ⌂ sqlserver [⠦ ..] 87.06MB / 578.2MB Pulling
8.2s
  ⌂ 3d29464607d8 Downloading  26.54MB/31.58MB
7.0s
  ⌂ 3374dd749958 Downloading  31.34MB/475.4MB
7.0s
  ⌂ 18952140cc7b Downloading  29.18MB/71.23MB
7.0s
[+] Running 0/4
  ⌂ sqlserver [⠦ ..] 88.15MB / 578.2MB Pulling
8.3s
  ⌂ 3d29464607d8 Downloading  26.54MB/31.58MB
7.1s
  ⌂ 3374dd749958 Downloading  31.88MB/475.4MB
7.1s
  ⌂ 18952140cc7b Downloading  29.72MB/71.23MB
7.1s
[+] Running 0/4
  ⌂ sqlserver [⠦ ..] 89.34MB / 578.2MB Pulling
8.4s
  ⌂ 3d29464607d8 Downloading  27.2MB/31.58MB
7.2s
  ⌂ 3374dd749958 Downloading  31.88MB/475.4MB
7.2s
  ⌂ 18952140cc7b Downloading  30.26MB/71.23MB
7.2s
[+] Running 0/4
  ⌂ sqlserver [⠦ ..] 91.08MB / 578.2MB Pulling
8.5s
  ⌂ 3d29464607d8 Downloading  27.85MB/31.58MB
7.3s
  ⌂ 3374dd749958 Downloading  32.96MB/475.4MB
7.3s
  ⌂ 18952140cc7b Downloading  30.26MB/71.23MB
7.3s
[+] Running 0/4
  ⌂ sqlserver [⠦ ..] 92.16MB / 578.2MB Pulling
8.6s
  ⌂ 3d29464607d8 Downloading  27.85MB/31.58MB
7.4s
  ⌂ 3374dd749958 Downloading  33.51MB/475.4MB
7.4s
  ⌂ 18952140cc7b Downloading  30.8MB/71.23MB
7.4s
[+] Running 0/4
  ⌂ sqlserver [⠦ ..] 93.9MB / 578.2MB Pulling
8.7s
  ⌂ 3d29464607d8 Downloading  28.51MB/31.58MB
7.5s
  ⌂ 3374dd749958 Downloading  34.05MB/475.4MB
7.5s
```

```
    .. 18952140cc7b Downloading 31.34MB/71.23MB
7.5s
[+] Running 0/4
  : sqlserver [..] 94.55MB / 578.2MB Pulling
8.8s
  .. 3d29464607d8 Downloading 29.16MB/31.58MB
7.6s
  .. 3374dd749958 Downloading 34.05MB/475.4MB
7.6s
  .. 18952140cc7b Downloading 31.34MB/71.23MB
7.6s
[+] Running 0/4
  : sqlserver [..] 96.17MB / 578.2MB Pulling
8.9s
  .. 3d29464607d8 Downloading 29.16MB/31.58MB
7.7s
  .. 3374dd749958 Downloading 35.13MB/475.4MB
7.7s
  .. 18952140cc7b Downloading 31.88MB/71.23MB
7.7s
[+] Running 0/4
  : sqlserver [..] 97.91MB / 578.2MB Pulling
9.0s
  .. 3d29464607d8 Downloading 29.82MB/31.58MB
7.8s
  .. 3374dd749958 Downloading 35.67MB/475.4MB
7.8s
  .. 18952140cc7b Downloading 32.42MB/71.23MB
7.8s
[+] Running 0/4
  : sqlserver [..] 98.45MB / 578.2MB Pulling
9.1s
  .. 3d29464607d8 Downloading 29.82MB/31.58MB
7.9s
  .. 3374dd749958 Downloading 35.67MB/475.4MB
7.9s
  .. 18952140cc7b Downloading 32.96MB/71.23MB
7.9s
[+] Running 0/4
  : sqlserver [..] 100.2MB / 578.2MB Pulling
9.2s
  .. 3d29464607d8 Downloading 30.47MB/31.58MB
8.0s
  .. 3374dd749958 Downloading 36.75MB/475.4MB
8.0s
  .. 18952140cc7b Downloading 32.96MB/71.23MB
8.0s
[+] Running 0/4
  : sqlserver [..] 101.9MB / 578.2MB Pulling
9.3s
  .. 3d29464607d8 Downloading 31.13MB/31.58MB
8.1s
  .. 3374dd749958 Downloading 37.29MB/475.4MB
8.1s
  .. 18952140cc7b Downloading 33.51MB/71.23MB
8.1s
[+] Running 1/4
```

```
  ↳ sqlserver [⠄ ..] 102.5MB / 578.2MB Pulling
9.4s
    ✓ 3d29464607d8 Download complete
8.2s
    ↳ 3374dd749958 Downloading 37.29MB/475.4MB
8.2s
    ↳ 18952140cc7b Downloading 34.05MB/71.23MB
8.2s
[+] Running 0/4
  ↳ sqlserver [⠄ ..] 104.1MB / 578.2MB Pulling
9.5s
    ↳ 3d29464607d8 Extracting 327.7kB/31.58MB
8.3s
    ↳ 3374dd749958 Downloading 38.37MB/475.4MB
8.3s
    ↳ 18952140cc7b Downloading 34.59MB/71.23MB
8.3s
[+] Running 0/4
  ↳ sqlserver [⠄ ..] 105.2MB / 578.2MB Pulling
9.6s
    ↳ 3d29464607d8 Extracting 6.226MB/31.58MB
8.4s
    ↳ 3374dd749958 Downloading 39.45MB/475.4MB
8.4s
    ↳ 18952140cc7b Downloading 34.59MB/71.23MB
8.4s
[+] Running 0/4
  ↳ sqlserver [⠄ ..] 106.3MB / 578.2MB Pulling
9.7s
    ↳ 3d29464607d8 Extracting 9.83MB/31.58MB
8.5s
    ↳ 3374dd749958 Downloading 39.45MB/475.4MB
8.5s
    ↳ 18952140cc7b Downloading 35.67MB/71.23MB
8.5s
[+] Running 0/4
  ↳ sqlserver [⠄ ..] 107.3MB / 578.2MB Pulling
9.8s
    ↳ 3d29464607d8 Extracting 17.04MB/31.58MB
8.6s
    ↳ 3374dd749958 Downloading 40.53MB/475.4MB
8.6s
    ↳ 18952140cc7b Downloading 35.67MB/71.23MB
8.6s
[+] Running 0/4
  ↳ sqlserver [⠄ ..] 109.5MB / 578.2MB Pulling
9.9s
    ↳ 3d29464607d8 Extracting 25.56MB/31.58MB
8.7s
    ↳ 3374dd749958 Downloading 41.62MB/475.4MB
8.7s
    ↳ 18952140cc7b Downloading 36.75MB/71.23MB
8.7s
[+] Running 0/4
  ↳ sqlserver [⠄ ..] 110.6MB / 578.2MB Pulling
10.0s
    ↳ 3d29464607d8 Extracting 25.56MB/31.58MB
```

```
8.8s
  ⌈ 3374dd749958 Downloading  42.16MB/475.4MB
8.8s
  ⌈ 18952140cc7b Downloading  37.29MB/71.23MB
8.8s
[+] Running 0/4
  ⌈ sqlserver [⠄⠄] 111.7MB / 578.2MB Pulling
10.1s
  ⌈ 3d29464607d8 Extracting  27.53MB/31.58MB
8.9s
  ⌈ 3374dd749958 Downloading  43.24MB/475.4MB
8.9s
  ⌈ 18952140cc7b Downloading  37.29MB/71.23MB
8.9s
[+] Running 0/4
  ⌈ sqlserver [⠄⠄] 112.7MB / 578.2MB Pulling
10.2s
  ⌈ 3d29464607d8 Extracting  30.15MB/31.58MB
9.0s
  ⌈ 3374dd749958 Downloading  43.24MB/475.4MB
9.0s
  ⌈ 18952140cc7b Downloading  38.37MB/71.23MB
9.0s
[+] Running 0/4
  ⌈ sqlserver [⠄⠄] 113.8MB / 578.2MB Pulling
10.3s
  ⌈ 3d29464607d8 Extracting  30.15MB/31.58MB
9.1s
  ⌈ 3374dd749958 Downloading  44.32MB/475.4MB
9.1s
  ⌈ 18952140cc7b Downloading  38.37MB/71.23MB
9.1s
[+] Running 0/4
  ⌈ sqlserver [⠄⠄] 116.4MB / 578.2MB Pulling
10.4s
  ⌈ 3d29464607d8 Extracting  31.58MB/31.58MB
9.2s
  ⌈ 3374dd749958 Downloading  45.4MB/475.4MB
9.2s
  ⌈ 18952140cc7b Downloading  39.45MB/71.23MB
9.2s
[+] Running 1/4
  ⌈ sqlserver [⠄⠄] 117MB / 578.2MB Pulling
10.5s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading  45.4MB/475.4MB
9.3s
  ⌈ 18952140cc7b Downloading  39.99MB/71.23MB
9.3s
[+] Running 1/4
  ⌈ sqlserver [⠄⠄] 119.1MB / 578.2MB Pulling
10.6s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading  46.48MB/475.4MB
9.4s
```

```
    .: 18952140cc7b Downloading  41.07MB/71.23MB
9.4s
[+] Running 1/4
  .: sqlserver [⠄ ..] 120.2MB / 578.2MB Pulling
10.7s
    ✓ 3d29464607d8 Pull complete
9.3s
    .: 3374dd749958 Downloading  47.56MB/475.4MB
9.5s
    .: 18952140cc7b Downloading  41.07MB/71.23MB
9.5s
[+] Running 1/4
  .: sqlserver [⠄ ..] 121.3MB / 578.2MB Pulling
10.8s
    ✓ 3d29464607d8 Pull complete
9.3s
    .: 3374dd749958 Downloading  47.56MB/475.4MB
9.6s
    .: 18952140cc7b Downloading  42.16MB/71.23MB
9.6s
[+] Running 1/4
  .: sqlserver [⠄ ..] 122.4MB / 578.2MB Pulling
10.9s
    ✓ 3d29464607d8 Pull complete
9.3s
    .: 3374dd749958 Downloading  48.64MB/475.4MB
9.7s
    .: 18952140cc7b Downloading  42.16MB/71.23MB
9.7s
[+] Running 1/4
  .: sqlserver [⠄ ..] 124.5MB / 578.2MB Pulling
11.0s
    ✓ 3d29464607d8 Pull complete
9.3s
    .: 3374dd749958 Downloading  49.73MB/475.4MB
9.8s
    .: 18952140cc7b Downloading  43.24MB/71.23MB
9.8s
[+] Running 1/4
  .: sqlserver [⠄ ..] 125.6MB / 578.2MB Pulling
11.1s
    ✓ 3d29464607d8 Pull complete
9.3s
    .: 3374dd749958 Downloading  50.81MB/475.4MB
9.9s
    .: 18952140cc7b Downloading  43.24MB/71.23MB
9.9s
[+] Running 1/4
  .: sqlserver [⠄ ..] 126.7MB / 578.2MB Pulling
11.2s
    ✓ 3d29464607d8 Pull complete
9.3s
    .: 3374dd749958 Downloading  50.81MB/475.4MB
10.0s
    .: 18952140cc7b Downloading  44.32MB/71.23MB
10.0s
[+] Running 1/4
```

```
: sqlserver [::] 127.8MB / 578.2MB Pulling
11.3s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading 51.89MB/475.4MB
10.1s
  ⚡ 18952140cc7b Downloading 44.32MB/71.23MB
10.1s
[+] Running 1/4
  ↳ sqlserver [::] 130MB / 578.2MB Pulling
11.4s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading 52.97MB/475.4MB
10.2s
  ⚡ 18952140cc7b Downloading 45.4MB/71.23MB
10.2s
[+] Running 1/4
  ↳ sqlserver [::] 130.5MB / 578.2MB Pulling
11.5s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading 52.97MB/475.4MB
10.3s
  ⚡ 18952140cc7b Downloading 45.94MB/71.23MB
10.3s
[+] Running 1/4
  ↳ sqlserver [::] 131.6MB / 578.2MB Pulling
11.6s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading 54.05MB/475.4MB
10.4s
  ⚡ 18952140cc7b Downloading 45.94MB/71.23MB
10.4s
[+] Running 1/4
  ↳ sqlserver [::] 133.7MB / 578.2MB Pulling
11.7s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading 55.13MB/475.4MB
10.5s
  ⚡ 18952140cc7b Downloading 47.02MB/71.23MB
10.5s
[+] Running 1/4
  ↳ sqlserver [::] 134.8MB / 578.2MB Pulling
11.8s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading 55.13MB/475.4MB
10.6s
  ⚡ 18952140cc7b Downloading 48.1MB/71.23MB
10.6s
[+] Running 1/4
  ↳ sqlserver [::] 135.9MB / 578.2MB Pulling
11.9s
  ✓ 3d29464607d8 Pull complete
```

```
9.3s
  ⌈ 3374dd749958 Downloading  56.21MB/475.4MB
10.7s
  ⌈ 18952140cc7b Downloading   48.1MB/71.23MB
10.7s
[+] Running 1/4
  ⌈ sqlserver [⠄⠄] 138.1MB / 578.2MB Pulling
12.0s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading  57.29MB/475.4MB
10.8s
  ⌈ 18952140cc7b Downloading   49.18MB/71.23MB
10.8s
[+] Running 1/4
  ⌈ sqlserver [⠄⠄] 138.1MB / 578.2MB Pulling
12.1s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading  57.29MB/475.4MB
10.9s
  ⌈ 18952140cc7b Downloading   49.18MB/71.23MB
10.9s
[+] Running 1/4
  ⌈ sqlserver [⠄⠄] 140.2MB / 578.2MB Pulling
12.2s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading  58.38MB/475.4MB
11.0s
  ⌈ 18952140cc7b Downloading   50.27MB/71.23MB
11.0s
[+] Running 1/4
  ⌈ sqlserver [⠄⠄] 141.3MB / 578.2MB Pulling
12.3s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading  59.46MB/475.4MB
11.1s
  ⌈ 18952140cc7b Downloading   50.27MB/71.23MB
11.1s
[+] Running 1/4
  ⌈ sqlserver [⠄⠄] 142.4MB / 578.2MB Pulling
12.4s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading  59.46MB/475.4MB
11.2s
  ⌈ 18952140cc7b Downloading   51.35MB/71.23MB
11.2s
[+] Running 1/4
  ⌈ sqlserver [⠄⠄] 143.5MB / 578.2MB Pulling
12.5s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading  60.54MB/475.4MB
11.3s
```

```
: 18952140cc7b Downloading 51.35MB/71.23MB
11.3s
[+] Running 1/4
  ↳ sqlserver [⠄⠄] 145.6MB / 578.2MB Pulling
12.6s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading 61.62MB/475.4MB
11.4s
  ↳ 18952140cc7b Downloading 52.43MB/71.23MB
11.4s
[+] Running 1/4
  ↳ sqlserver [⠄⠄] 146.7MB / 578.2MB Pulling
12.7s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading 61.62MB/475.4MB
11.5s
  ↳ 18952140cc7b Downloading 53.51MB/71.23MB
11.5s
[+] Running 1/4
  ↳ sqlserver [⠄⠄] 147.8MB / 578.2MB Pulling
12.8s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading 62.7MB/475.4MB
11.6s
  ↳ 18952140cc7b Downloading 53.51MB/71.23MB
11.6s
[+] Running 1/4
  ↳ sqlserver [⠄⠄] 150MB / 578.2MB Pulling
12.9s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading 63.78MB/475.4MB
11.7s
  ↳ 18952140cc7b Downloading 54.59MB/71.23MB
11.7s
[+] Running 1/4
  ↳ sqlserver [⠄⠄] 150MB / 578.2MB Pulling
13.0s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading 63.78MB/475.4MB
11.8s
  ↳ 18952140cc7b Downloading 54.59MB/71.23MB
11.8s
[+] Running 1/4
  ↳ sqlserver [⠄⠄] 152.1MB / 578.2MB Pulling
13.1s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading 64.86MB/475.4MB
11.9s
  ↳ 18952140cc7b Downloading 55.67MB/71.23MB
11.9s
[+] Running 1/4
```

```
  : sqlserver [..] 153.2MB / 578.2MB Pulling
13.2s
    ✓ 3d29464607d8 Pull complete
9.3s
    : 3374dd749958 Downloading 65.95MB/475.4MB
12.0s
    : 18952140cc7b Downloading 55.67MB/71.23MB
12.0s
[+] Running 1/4
  : sqlserver [..] 154.3MB / 578.2MB Pulling
13.3s
    ✓ 3d29464607d8 Pull complete
9.3s
    : 3374dd749958 Downloading 65.95MB/475.4MB
12.1s
    : 18952140cc7b Downloading 56.75MB/71.23MB
12.1s
[+] Running 1/4
  : sqlserver [..] 156.4MB / 578.2MB Pulling
13.4s
    ✓ 3d29464607d8 Pull complete
9.3s
    : 3374dd749958 Downloading 67.03MB/475.4MB
12.2s
    : 18952140cc7b Downloading 57.84MB/71.23MB
12.2s
[+] Running 1/4
  : sqlserver [..] 157.5MB / 578.2MB Pulling
13.5s
    ✓ 3d29464607d8 Pull complete
9.3s
    : 3374dd749958 Downloading 68.11MB/475.4MB
12.3s
    : 18952140cc7b Downloading 57.84MB/71.23MB
12.3s
[+] Running 1/4
  : sqlserver [..] 158.6MB / 578.2MB Pulling
13.6s
    ✓ 3d29464607d8 Pull complete
9.3s
    : 3374dd749958 Downloading 68.11MB/475.4MB
12.4s
    : 18952140cc7b Downloading 58.92MB/71.23MB
12.4s
[+] Running 1/4
  : sqlserver [..] 159.7MB / 578.2MB Pulling
13.7s
    ✓ 3d29464607d8 Pull complete
9.3s
    : 3374dd749958 Downloading 69.19MB/475.4MB
12.5s
    : 18952140cc7b Downloading 58.92MB/71.23MB
12.5s
[+] Running 1/4
  : sqlserver [..] 161.9MB / 578.2MB Pulling
13.8s
    ✓ 3d29464607d8 Pull complete
```

```
9.3s
  ⌈ 3374dd749958 Downloading  70.27MB/475.4MB
12.6s
  ⌈ 18952140cc7b Downloading      60MB/71.23MB
12.6s
[+] Running 1/4
  ⌈ sqlserver [..] 162.4MB / 578.2MB Pulling
13.9s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading  70.27MB/475.4MB
12.7s
  ⌈ 18952140cc7b Downloading  60.54MB/71.23MB
12.7s
[+] Running 1/4
  ⌈ sqlserver [..] 163.5MB / 578.2MB Pulling
14.0s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading  71.35MB/475.4MB
12.8s
  ⌈ 18952140cc7b Downloading  60.54MB/71.23MB
12.8s
[+] Running 1/4
  ⌈ sqlserver [..] 165.6MB / 578.2MB Pulling
14.1s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading  72.43MB/475.4MB
12.9s
  ⌈ 18952140cc7b Downloading  61.62MB/71.23MB
12.9s
[+] Running 1/4
  ⌈ sqlserver [..] 166.7MB / 578.2MB Pulling
14.2s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading  72.43MB/475.4MB
13.0s
  ⌈ 18952140cc7b Downloading     62.7MB/71.23MB
13.0s
[+] Running 1/4
  ⌈ sqlserver [..] 167.8MB / 578.2MB Pulling
14.3s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading  73.52MB/475.4MB
13.1s
  ⌈ 18952140cc7b Downloading     62.7MB/71.23MB
13.1s
[+] Running 1/4
  ⌈ sqlserver [..] 170MB / 578.2MB Pulling
14.4s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading  74.6MB/475.4MB
13.2s
```



```
  ⚡ sqlserver [●.●] 178.6MB / 578.2MB Pulling
15.1s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading 78.92MB/475.4MB
13.9s
    ⚡ 18952140cc7b Downloading 68.11MB/71.23MB
13.9s
[+] Running 1/4
  ⚡ sqlserver [●.●] 179.7MB / 578.2MB Pulling
15.2s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      80MB/475.4MB
14.0s
    ⚡ 18952140cc7b Downloading 68.11MB/71.23MB
14.0s
[+] Running 1/4
  ⚡ sqlserver [●.●] 181.9MB / 578.2MB Pulling
15.3s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading 81.08MB/475.4MB
14.1s
    ⚡ 18952140cc7b Downloading 69.19MB/71.23MB
14.1s
[+] Running 1/4
  ⚡ sqlserver [●.●] 181.9MB / 578.2MB Pulling
15.4s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading 81.08MB/475.4MB
14.2s
    ⚡ 18952140cc7b Downloading 69.19MB/71.23MB
14.2s
[+] Running 1/4
  ⚡ sqlserver [●.●] 184MB / 578.2MB Pulling
15.5s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading 82.17MB/475.4MB
14.3s
    ⚡ 18952140cc7b Downloading 70.27MB/71.23MB
14.3s
[+] Running 1/4
  ⚡ sqlserver [●.●] 185.1MB / 578.2MB Pulling
15.6s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading 83.25MB/475.4MB
14.4s
    ⚡ 18952140cc7b Downloading 70.27MB/71.23MB
14.4s
[+] Running 2/4
  ⚡ sqlserver [●.●] 186.2MB / 578.2MB Pulling
15.7s
    ✓ 3d29464607d8 Pull complete
```

```
9.3s
  ∴ 3374dd749958 Downloading      84.33MB/475....
14.5s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ∴ sqlserver [..] 186.2MB / 578.2MB Pulling
15.8s
  ✓ 3d29464607d8 Pull complete
9.3s
  ∴ 3374dd749958 Downloading      84.33MB/475....
14.6s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ∴ sqlserver [..] 187.8MB / 578.2MB Pulling
15.9s
  ✓ 3d29464607d8 Pull complete
9.3s
  ∴ 3374dd749958 Downloading      85.95MB/475....
14.7s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ∴ sqlserver [..] 190MB / 578.2MB Pulling
16.0s
  ✓ 3d29464607d8 Pull complete
9.3s
  ∴ 3374dd749958 Downloading      88.11MB/475....
14.8s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ∴ sqlserver [..] 191.6MB / 578.2MB Pulling
16.1s
  ✓ 3d29464607d8 Pull complete
9.3s
  ∴ 3374dd749958 Downloading      89.74MB/475....
14.9s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ∴ sqlserver [..] 191.6MB / 578.2MB Pulling
16.2s
  ✓ 3d29464607d8 Pull complete
9.3s
  ∴ 3374dd749958 Downloading      89.74MB/475....
15.0s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ∴ sqlserver [..] 193.2MB / 578.2MB Pulling
16.3s
  ✓ 3d29464607d8 Pull complete
9.3s
  ∴ 3374dd749958 Downloading      91.36MB/475....
15.1s
```

```
✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [●.●] 194.8MB / 578.2MB Pulling
16.4s
✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      92.98MB/475...
15.2s
✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [●.●] 196.5MB / 578.2MB Pulling
16.5s
✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      94.6MB/475...
15.3s
✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [●.●] 198.1MB / 578.2MB Pulling
16.6s
✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      96.22MB/475...
15.4s
✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [●.●] 198.1MB / 578.2MB Pulling
16.7s
✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      96.22MB/475...
15.5s
✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [●.●] 199.7MB / 578.2MB Pulling
16.8s
✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      97.85MB/475...
15.6s
✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [●.●] 201.3MB / 578.2MB Pulling
16.9s
✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      99.47MB/475...
15.7s
✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
```

```
  ⚡ sqlserver [..] 202.9MB / 578.2MB Pulling
17.0s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      101.1MB/475....
15.8s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [..] 204.6MB / 578.2MB Pulling
17.1s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      102.7MB/475....
15.9s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [..] 204.6MB / 578.2MB Pulling
17.2s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      102.7MB/475....
16.0s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [..] 206.2MB / 578.2MB Pulling
17.3s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      104.3MB/475....
16.1s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [..] 207.8MB / 578.2MB Pulling
17.4s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      106MB/475....
16.2s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [..] 207.8MB / 578.2MB Pulling
17.5s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      106MB/475....
16.3s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [..] 209.4MB / 578.2MB Pulling
17.6s
    ✓ 3d29464607d8 Pull complete
```

```
9.3s
  ↳ 3374dd749958 Downloading      107.6MB/475....
16.4s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [..] 211.1MB / 578.2MB Pulling
17.7s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading      109.2MB/475....
16.5s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [..] 212.7MB / 578.2MB Pulling
17.8s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading      110.8MB/475....
16.6s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [..] 214.3MB / 578.2MB Pulling
17.9s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading      112.4MB/475....
16.7s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [..] 214.3MB / 578.2MB Pulling
18.0s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading      112.4MB/475....
16.8s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [..] 215.9MB / 578.2MB Pulling
18.1s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading      114.1MB/475....
16.9s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [..] 217.5MB / 578.2MB Pulling
18.2s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading      115.7MB/475....
17.0s
```

```
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 219.2MB / 578.2MB Pulling
18.3s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      117.3MB/475.....
17.1s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 220.8MB / 578.2MB Pulling
18.4s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      118.9MB/475.....
17.2s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 220.8MB / 578.2MB Pulling
18.5s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      118.9MB/475.....
17.3s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 222.4MB / 578.2MB Pulling
18.6s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      120.6MB/475.....
17.4s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 224MB / 578.2MB Pulling
18.7s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      122.2MB/475.....
17.5s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 225.7MB / 578.2MB Pulling
18.8s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      123.8MB/475.....
17.6s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
```

```
  ⚡ sqlserver [⠄⠄⠄] 227.3MB / 578.2MB Pulling
18.9s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      125.4MB/475.....
17.7s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠄⠄⠄] 228.9MB / 578.2MB Pulling
19.0s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      127MB/475.....
17.8s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠄⠄⠄] 228.9MB / 578.2MB Pulling
19.1s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      127MB/475.....
17.9s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠄⠄⠄] 230.5MB / 578.2MB Pulling
19.2s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      128.7MB/475.....
18.0s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠄⠄⠄] 232.1MB / 578.2MB Pulling
19.3s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      130.3MB/475.....
18.1s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠄⠄⠄] 233.8MB / 578.2MB Pulling
19.4s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      131.9MB/475.....
18.2s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠄⠄⠄] 235.4MB / 578.2MB Pulling
19.5s
    ✓ 3d29464607d8 Pull complete
```

```
9.3s
  : 3374dd749958 Downloading      133.5MB/475....
18.3s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  : sqlserver [■■■] 235.4MB / 578.2MB Pulling
19.6s
  ✓ 3d29464607d8 Pull complete
9.3s
  : 3374dd749958 Downloading      133.5MB/475....
18.4s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  : sqlserver [■■■] 237MB / 578.2MB Pulling
19.7s
  ✓ 3d29464607d8 Pull complete
9.3s
  : 3374dd749958 Downloading      135.2MB/475....
18.5s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  : sqlserver [■■■] 238.6MB / 578.2MB Pulling
19.8s
  ✓ 3d29464607d8 Pull complete
9.3s
  : 3374dd749958 Downloading      136.8MB/475....
18.6s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  : sqlserver [■■■] 240.3MB / 578.2MB Pulling
19.9s
  ✓ 3d29464607d8 Pull complete
9.3s
  : 3374dd749958 Downloading      138.4MB/475....
18.7s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  : sqlserver [■■■] 241.9MB / 578.2MB Pulling
20.0s
  ✓ 3d29464607d8 Pull complete
9.3s
  : 3374dd749958 Downloading      140MB/475....
18.8s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  : sqlserver [■■■] 241.9MB / 578.2MB Pulling
20.1s
  ✓ 3d29464607d8 Pull complete
9.3s
  : 3374dd749958 Downloading      140MB/475....
18.9s
```

```
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚠ sqlserver [⠄⠄⠄] 243.5MB / 578.2MB Pulling
20.2s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚠ 3374dd749958 Downloading      141.6MB/475....
19.0s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚠ sqlserver [⠄⠄⠄] 244MB / 578.2MB Pulling
20.3s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚠ 3374dd749958 Downloading      142.2MB/475....
19.1s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚠ sqlserver [⠄⠄⠄] 245.7MB / 578.2MB Pulling
20.4s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚠ 3374dd749958 Downloading      143.8MB/475....
19.2s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚠ sqlserver [⠄⠄⠄] 247.3MB / 578.2MB Pulling
20.5s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚠ 3374dd749958 Downloading      145.4MB/475....
19.3s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚠ sqlserver [⠄⠄⠄] 248.9MB / 578.2MB Pulling
20.6s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚠ 3374dd749958 Downloading      147MB/475....
19.4s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚠ sqlserver [⠄⠄⠄] 250.5MB / 578.2MB Pulling
20.7s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚠ 3374dd749958 Downloading      148.7MB/475....
19.5s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
```

```
: sqlserver [■■■] 250.5MB / 578.2MB Pulling
20.8s
    ✓ 3d29464607d8 Pull complete
9.3s
    : 3374dd749958 Downloading      148.7MB/475....
19.6s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
: sqlserver [■■■] 252.1MB / 578.2MB Pulling
20.9s
    ✓ 3d29464607d8 Pull complete
9.3s
    : 3374dd749958 Downloading      150.3MB/475....
19.7s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
: sqlserver [■■■] 253.8MB / 578.2MB Pulling
21.0s
    ✓ 3d29464607d8 Pull complete
9.3s
    : 3374dd749958 Downloading      151.9MB/475....
19.8s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
: sqlserver [■■■] 255.4MB / 578.2MB Pulling
21.1s
    ✓ 3d29464607d8 Pull complete
9.3s
    : 3374dd749958 Downloading      153.5MB/475....
19.9s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
: sqlserver [■■■] 257MB / 578.2MB Pulling
21.2s
    ✓ 3d29464607d8 Pull complete
9.3s
    : 3374dd749958 Downloading      155.2MB/475....
20.0s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
: sqlserver [■■■] 258.6MB / 578.2MB Pulling
21.3s
    ✓ 3d29464607d8 Pull complete
9.3s
    : 3374dd749958 Downloading      156.8MB/475....
20.1s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
: sqlserver [■■■] 258.6MB / 578.2MB Pulling
21.4s
    ✓ 3d29464607d8 Pull complete
```



```
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 268.4MB / 578.2MB Pulling
22.1s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      166.5MB/475....
20.9s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 270MB / 578.2MB Pulling
22.2s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      168.1MB/475....
21.0s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 271.6MB / 578.2MB Pulling
22.3s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      169.8MB/475....
21.1s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 271.6MB / 578.2MB Pulling
22.4s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      169.8MB/475....
21.2s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 273.2MB / 578.2MB Pulling
22.5s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      171.4MB/475....
21.3s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 274.9MB / 578.2MB Pulling
22.6s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      173MB/475....
21.4s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
```

```
  ↳ sqlserver [■■■] 276.5MB / 578.2MB Pulling
22.7s
    ✓ 3d29464607d8 Pull complete
9.3s
    ↳ 3374dd749958 Downloading      174.6MB/475.....
21.5s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [■■■] 278.1MB / 578.2MB Pulling
22.8s
    ✓ 3d29464607d8 Pull complete
9.3s
    ↳ 3374dd749958 Downloading      176.2MB/475.....
21.6s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [■■■] 278.1MB / 578.2MB Pulling
22.9s
    ✓ 3d29464607d8 Pull complete
9.3s
    ↳ 3374dd749958 Downloading      176.2MB/475.....
21.7s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [■■■] 279.7MB / 578.2MB Pulling
23.0s
    ✓ 3d29464607d8 Pull complete
9.3s
    ↳ 3374dd749958 Downloading      177.9MB/475.....
21.8s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [■■■] 281.3MB / 578.2MB Pulling
23.1s
    ✓ 3d29464607d8 Pull complete
9.3s
    ↳ 3374dd749958 Downloading      179.5MB/475.....
21.9s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [■■■] 283MB / 578.2MB Pulling
23.2s
    ✓ 3d29464607d8 Pull complete
9.3s
    ↳ 3374dd749958 Downloading      181.1MB/475.....
22.0s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [■■■] 284.6MB / 578.2MB Pulling
23.3s
    ✓ 3d29464607d8 Pull complete
```

```
9.3s
  ⌈ 3374dd749958 Downloading      182.7MB/475....
22.1s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [■■■] 284.6MB / 578.2MB Pulling
23.4s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading      182.7MB/475....
22.2s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [■■■] 286.2MB / 578.2MB Pulling
23.5s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading      184.4MB/475....
22.3s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [■■■] 287.8MB / 578.2MB Pulling
23.6s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading      186MB/475....
22.4s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [■■■] 289.5MB / 578.2MB Pulling
23.7s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading      187.6MB/475....
22.5s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [■■■] 291.1MB / 578.2MB Pulling
23.8s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading      189.2MB/475....
22.6s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [■■■] 292.7MB / 578.2MB Pulling
23.9s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading      190.8MB/475....
22.7s
```

```
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 292.7MB / 578.2MB Pulling
24.0s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      190.8MB/475.....
22.8s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 294.3MB / 578.2MB Pulling
24.1s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      192.5MB/475.....
22.9s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 295.9MB / 578.2MB Pulling
24.2s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      194.1MB/475.....
23.0s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 295.9MB / 578.2MB Pulling
24.3s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      194.1MB/475.....
23.1s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 297.6MB / 578.2MB Pulling
24.4s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      195.7MB/475.....
23.2s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 299.7MB / 578.2MB Pulling
24.5s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      197.9MB/475.....
23.3s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
```

```
  ↳ sqlserver [..] 301.3MB / 578.2MB Pulling
24.6s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading      199.5MB/475....
23.4s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [..] 303MB / 578.2MB Pulling
24.7s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading      201.1MB/475....
23.5s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [..] 304.6MB / 578.2MB Pulling
24.8s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading      202.7MB/475....
23.6s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [..] 304.6MB / 578.2MB Pulling
24.9s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading      202.7MB/475....
23.7s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [..] 306.2MB / 578.2MB Pulling
25.0s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading      204.4MB/475....
23.8s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [..] 307.8MB / 578.2MB Pulling
25.1s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading      206MB/475....
23.9s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [..] 309.5MB / 578.2MB Pulling
25.2s
  ✓ 3d29464607d8 Pull complete
```

```
9.3s
  ⚡ 3374dd749958 Downloading          207.6MB/475...
24.0s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠠⠠⠠] 311.1MB / 578.2MB Pulling
25.3s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading          209.2MB/475...
24.1s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠠⠠⠠] 312.7MB / 578.2MB Pulling
25.4s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading          210.8MB/475...
24.2s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠠⠠⠠] 312.7MB / 578.2MB Pulling
25.5s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading          210.8MB/475...
24.3s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠠⠠⠠] 314.3MB / 578.2MB Pulling
25.6s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading          212.5MB/475...
24.4s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠠⠠⠠] 315.9MB / 578.2MB Pulling
25.7s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading          214.1MB/475...
24.5s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠠⠠⠠] 317.6MB / 578.2MB Pulling
25.8s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading          215.7MB/475...
24.6s
```

```
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚠ sqlserver [⠄⠄⠄] 319.2MB / 578.2MB Pulling
25.9s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      217.3MB/475.....
24.7s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚠ sqlserver [⠄⠄⠄] 319.2MB / 578.2MB Pulling
26.0s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      217.3MB/475.....
24.8s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚠ sqlserver [⠄⠄⠄] 320.8MB / 578.2MB Pulling
26.1s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚠ 3374dd749958 Downloading      219MB/475.....
24.9s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚠ sqlserver [⠄⠄⠄] 322.4MB / 578.2MB Pulling
26.2s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚠ 3374dd749958 Downloading      220.6MB/475.....
25.0s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚠ sqlserver [⠄⠄⠄] 324.1MB / 578.2MB Pulling
26.3s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚠ 3374dd749958 Downloading      222.2MB/475.....
25.1s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚠ sqlserver [⠄⠄⠄] 325.7MB / 578.2MB Pulling
26.4s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚠ 3374dd749958 Downloading      223.8MB/475.....
25.2s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
```

```
  ↳ sqlserver [..] 327.3MB / 578.2MB Pulling
26.5s
    ✓ 3d29464607d8 Pull complete
9.3s
    ↳ 3374dd749958 Downloading      225.4MB/475....
25.3s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [..] 327.3MB / 578.2MB Pulling
26.6s
    ✓ 3d29464607d8 Pull complete
9.3s
    ↳ 3374dd749958 Downloading      225.4MB/475....
25.4s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [..] 328.9MB / 578.2MB Pulling
26.7s
    ✓ 3d29464607d8 Pull complete
9.3s
    ↳ 3374dd749958 Downloading      227.1MB/475....
25.5s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [..] 330.5MB / 578.2MB Pulling
26.8s
    ✓ 3d29464607d8 Pull complete
9.3s
    ↳ 3374dd749958 Downloading      228.7MB/475....
25.6s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [..] 332.2MB / 578.2MB Pulling
26.9s
    ✓ 3d29464607d8 Pull complete
9.3s
    ↳ 3374dd749958 Downloading      230.3MB/475....
25.7s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [..] 333.8MB / 578.2MB Pulling
27.0s
    ✓ 3d29464607d8 Pull complete
9.3s
    ↳ 3374dd749958 Downloading      231.9MB/475....
25.8s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [..] 333.8MB / 578.2MB Pulling
27.1s
    ✓ 3d29464607d8 Pull complete
```

```
9.3s
  ⚡ 3374dd749958 Downloading      231.9MB/475....
25.9s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠄⠄⠄] 335.4MB / 578.2MB Pulling
27.2s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      233.6MB/475....
26.0s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠄⠄⠄] 337MB / 578.2MB Pulling
27.3s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      235.2MB/475....
26.1s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠄⠄⠄] 338.7MB / 578.2MB Pulling
27.4s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      236.8MB/475....
26.2s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠄⠄⠄] 340.3MB / 578.2MB Pulling
27.5s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      238.4MB/475....
26.3s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠄⠄⠄] 340.3MB / 578.2MB Pulling
27.6s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      238.4MB/475....
26.4s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠄⠄⠄] 341.9MB / 578.2MB Pulling
27.7s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      240MB/475....
26.5s
```

```
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 344.1MB / 578.2MB Pulling
27.8s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      242.2MB/475.....
26.7s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 345.7MB / 578.2MB Pulling
27.9s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      243.8MB/475.....
26.7s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 345.7MB / 578.2MB Pulling
28.0s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      243.8MB/475.....
26.8s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 347.3MB / 578.2MB Pulling
28.1s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      245.4MB/475.....
26.9s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 348.9MB / 578.2MB Pulling
28.2s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      247.1MB/475.....
27.0s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 350.5MB / 578.2MB Pulling
28.3s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      248.7MB/475.....
27.1s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
```

```
  ↳ sqlserver [::] 352.2MB / 578.2MB Pulling
28.4s
    ✓ 3d29464607d8 Pull complete
9.3s
    ↳ 3374dd749958 Downloading      250.3MB/475....
27.2s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [::] 352.2MB / 578.2MB Pulling
28.5s
    ✓ 3d29464607d8 Pull complete
9.3s
    ↳ 3374dd749958 Downloading      250.3MB/475....
27.3s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [::] 353.8MB / 578.2MB Pulling
28.6s
    ✓ 3d29464607d8 Pull complete
9.3s
    ↳ 3374dd749958 Downloading      251.9MB/475....
27.4s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [::] 355.4MB / 578.2MB Pulling
28.7s
    ✓ 3d29464607d8 Pull complete
9.3s
    ↳ 3374dd749958 Downloading      253.6MB/475....
27.5s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [::] 357MB / 578.2MB Pulling
28.8s
    ✓ 3d29464607d8 Pull complete
9.3s
    ↳ 3374dd749958 Downloading      255.2MB/475....
27.6s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [::] 358.1MB / 578.2MB Pulling
28.9s
    ✓ 3d29464607d8 Pull complete
9.3s
    ↳ 3374dd749958 Downloading      256.3MB/475....
27.7s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [::] 358.1MB / 578.2MB Pulling
29.0s
    ✓ 3d29464607d8 Pull complete
```

```
9.3s
  ⌂ 3374dd749958 Downloading      256.3MB/475....
27.8s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌂ sqlserver [⠄⠄⠄] 360.3MB / 578.2MB Pulling
29.1s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌂ 3374dd749958 Downloading      258.4MB/475....
27.9s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌂ sqlserver [⠄⠄⠄] 361.9MB / 578.2MB Pulling
29.2s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌂ 3374dd749958 Downloading      260MB/475....
28.0s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌂ sqlserver [⠄⠄⠄] 363.5MB / 578.2MB Pulling
29.3s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌂ 3374dd749958 Downloading      261.7MB/475....
28.1s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌂ sqlserver [⠄⠄⠄] 365.1MB / 578.2MB Pulling
29.4s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌂ 3374dd749958 Downloading      263.3MB/475....
28.2s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌂ sqlserver [⠄⠄⠄] 366.8MB / 578.2MB Pulling
29.5s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌂ 3374dd749958 Downloading      264.9MB/475....
28.3s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌂ sqlserver [⠄⠄⠄] 366.8MB / 578.2MB Pulling
29.6s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌂ 3374dd749958 Downloading      264.9MB/475....
28.4s
```

```
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 368.4MB / 578.2MB Pulling
29.7s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      266.5MB/475.....
28.5s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 370MB / 578.2MB Pulling
29.8s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      268.2MB/475.....
28.6s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 371.6MB / 578.2MB Pulling
29.9s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      269.8MB/475.....
28.7s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 373.3MB / 578.2MB Pulling
30.0s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      271.4MB/475.....
28.8s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 374.9MB / 578.2MB Pulling
30.1s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      273MB/475.....
28.9s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 374.9MB / 578.2MB Pulling
30.2s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      273MB/475.....
29.0s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
```

```
: sqlserver [■■■] 376.5MB / 578.2MB Pulling
30.3s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      274.6MB/475....
29.1s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [■■■] 378.1MB / 578.2MB Pulling
30.4s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      276.3MB/475....
29.2s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [■■■] 379.7MB / 578.2MB Pulling
30.5s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      277.9MB/475....
29.3s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [■■■] 381.4MB / 578.2MB Pulling
30.6s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      279.5MB/475....
29.4s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [■■■] 381.4MB / 578.2MB Pulling
30.7s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      279.5MB/475....
29.5s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [■■■] 383MB / 578.2MB Pulling
30.8s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      281.1MB/475....
29.6s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [■■■] 384.6MB / 578.2MB Pulling
30.9s
  ✓ 3d29464607d8 Pull complete
```

```
9.3s
  ⌈ 3374dd749958 Downloading      282.8MB/475....
29.7s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [⠄⠄⠄] 386.2MB / 578.2MB Pulling
31.0s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading      284.4MB/475....
29.8s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [⠄⠄⠄] 387.9MB / 578.2MB Pulling
31.1s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading      286MB/475....
29.9s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [⠄⠄⠄] 387.9MB / 578.2MB Pulling
31.2s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading      286MB/475....
30.0s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [⠄⠄⠄] 389.5MB / 578.2MB Pulling
31.3s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading      287.6MB/475....
30.1s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [⠄⠄⠄] 391.1MB / 578.2MB Pulling
31.4s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading      289.2MB/475....
30.2s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [⠄⠄⠄] 392.7MB / 578.2MB Pulling
31.5s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading      290.9MB/475....
30.3s
```

```
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 394.3MB / 578.2MB Pulling
31.6s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      292.5MB/475.....
30.4s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 396MB / 578.2MB Pulling
31.7s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      294.1MB/475.....
30.5s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 396MB / 578.2MB Pulling
31.8s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      294.1MB/475.....
30.6s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 397.6MB / 578.2MB Pulling
31.9s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      295.7MB/475.....
30.7s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 399.2MB / 578.2MB Pulling
32.0s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      297.4MB/475.....
30.8s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 400.8MB / 578.2MB Pulling
32.1s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      299MB/475.....
30.9s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
```

```
  ⌂ sqlserver [■■■] 402.5MB / 578.2MB Pulling
32.2s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⌂ 3374dd749958 Downloading      300.6MB/475....
31.0s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌂ sqlserver [■■■] 402.5MB / 578.2MB Pulling
32.3s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⌂ 3374dd749958 Downloading      300.6MB/475....
31.1s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌂ sqlserver [■■■] 404.1MB / 578.2MB Pulling
32.4s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⌂ 3374dd749958 Downloading      302.2MB/475....
31.2s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌂ sqlserver [■■■] 405.7MB / 578.2MB Pulling
32.5s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⌂ 3374dd749958 Downloading      303.8MB/475....
31.3s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌂ sqlserver [■■■] 407.3MB / 578.2MB Pulling
32.6s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⌂ 3374dd749958 Downloading      305.5MB/475....
31.4s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌂ sqlserver [■■■] 408.9MB / 578.2MB Pulling
32.7s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⌂ 3374dd749958 Downloading      307.1MB/475....
31.5s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌂ sqlserver [■■■] 410.6MB / 578.2MB Pulling
32.8s
    ✓ 3d29464607d8 Pull complete
```

```
9.3s
  ⌈ 3374dd749958 Downloading      308.7MB/475....
31.6s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [⠄⠄⠄] 410.6MB / 578.2MB Pulling
32.9s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading      308.7MB/475....
31.7s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [⠄⠄⠄] 412.2MB / 578.2MB Pulling
33.0s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading      310.3MB/475....
31.8s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [⠄⠄⠄] 413.8MB / 578.2MB Pulling
33.1s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading      312MB/475....
31.9s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [⠄⠄⠄] 415.4MB / 578.2MB Pulling
33.2s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading      313.6MB/475....
32.0s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [⠄⠄⠄] 417.1MB / 578.2MB Pulling
33.3s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading      315.2MB/475....
32.1s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [⠄⠄⠄] 418.7MB / 578.2MB Pulling
33.4s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading      316.8MB/475....
32.2s
```

```
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠄⠄⠄] 418.7MB / 578.2MB Pulling
33.5s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      316.8MB/475.....
32.3s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠄⠄⠄] 420.3MB / 578.2MB Pulling
33.6s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      318.4MB/475.....
32.4s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠄⠄⠄] 421.9MB / 578.2MB Pulling
33.7s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      320.1MB/475.....
32.5s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠄⠄⠄] 423.5MB / 578.2MB Pulling
33.8s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      321.7MB/475.....
32.6s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠄⠄⠄] 425.2MB / 578.2MB Pulling
33.9s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      323.3MB/475.....
32.7s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠄⠄⠄] 425.2MB / 578.2MB Pulling
34.0s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      323.3MB/475.....
32.8s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
```

```
  ⚡ sqlserver [⠄⠄⠄] 426.8MB / 578.2MB Pulling
34.1s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      324.9MB/475.....
32.9s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠄⠄⠄] 428.4MB / 578.2MB Pulling
34.2s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      326.5MB/475.....
33.0s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠄⠄⠄] 429.5MB / 578.2MB Pulling
34.3s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      327.6MB/475.....
33.1s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠄⠄⠄] 429.5MB / 578.2MB Pulling
34.4s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      327.6MB/475.....
33.2s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠄⠄⠄] 432.2MB / 578.2MB Pulling
34.5s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      330.3MB/475.....
33.3s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠄⠄⠄] 433.8MB / 578.2MB Pulling
34.6s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      332MB/475.....
33.4s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠄⠄⠄] 435.4MB / 578.2MB Pulling
34.7s
    ✓ 3d29464607d8 Pull complete
```

```
9.3s
  ↳ 3374dd749958 Downloading      333.6MB/475....
33.5s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [⠄⠄⠄] 437.1MB / 578.2MB Pulling
34.8s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading      335.2MB/475....
33.6s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [⠄⠄⠄] 437.1MB / 578.2MB Pulling
34.9s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading      335.2MB/475....
33.7s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [⠄⠄⠄] 438.7MB / 578.2MB Pulling
35.0s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading      336.8MB/475....
33.8s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [⠄⠄⠄] 440.3MB / 578.2MB Pulling
35.1s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading      338.4MB/475....
33.9s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [⠄⠄⠄] 441.9MB / 578.2MB Pulling
35.2s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading      340.1MB/475....
34.0s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [⠄⠄⠄] 443.5MB / 578.2MB Pulling
35.3s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading      341.7MB/475....
34.1s
```

```
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠄⠄⠄] 445.2MB / 578.2MB Pulling
35.4s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      343.3MB/475.....
34.2s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠄⠄⠄] 445.2MB / 578.2MB Pulling
35.5s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      343.3MB/475.....
34.3s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠄⠄⠄] 446.8MB / 578.2MB Pulling
35.6s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      344.9MB/475.....
34.4s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠄⠄⠄] 448.4MB / 578.2MB Pulling
35.7s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      346.6MB/475.....
34.5s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠄⠄⠄] 450MB / 578.2MB Pulling
35.8s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      348.2MB/475.....
34.6s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [⠄⠄⠄] 451.7MB / 578.2MB Pulling
35.9s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      349.8MB/475.....
34.7s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
```

```
  sqlserver [■■■] 453.3MB / 578.2MB Pulling
36.0s
  ✓ 3d29464607d8 Pull complete
9.3s
  : 3374dd749958 Downloading      351.4MB/475.....
34.8s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  sqlserver [■■■] 453.3MB / 578.2MB Pulling
36.1s
  ✓ 3d29464607d8 Pull complete
9.3s
  : 3374dd749958 Downloading      351.4MB/475.....
34.9s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  sqlserver [■■■] 454.9MB / 578.2MB Pulling
36.2s
  ✓ 3d29464607d8 Pull complete
9.3s
  : 3374dd749958 Downloading      353MB/475.....
35.0s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  sqlserver [■■■] 456.5MB / 578.2MB Pulling
36.3s
  ✓ 3d29464607d8 Pull complete
9.3s
  : 3374dd749958 Downloading      354.7MB/475.....
35.1s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  sqlserver [■■■] 458.1MB / 578.2MB Pulling
36.4s
  ✓ 3d29464607d8 Pull complete
9.3s
  : 3374dd749958 Downloading      356.3MB/475.....
35.2s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  sqlserver [■■■] 459.8MB / 578.2MB Pulling
36.5s
  ✓ 3d29464607d8 Pull complete
9.3s
  : 3374dd749958 Downloading      357.9MB/475.....
35.3s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  sqlserver [■■■] 459.8MB / 578.2MB Pulling
36.6s
  ✓ 3d29464607d8 Pull complete
```

```
9.3s
  ↳ 3374dd749958 Downloading      357.9MB/475....
35.4s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [⠄⠄⠄] 461.4MB / 578.2MB Pulling
36.7s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading      359.5MB/475....
35.5s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [⠄⠄⠄] 463MB / 578.2MB Pulling
36.8s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading      361.2MB/475....
35.6s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [⠄⠄⠄] 464.6MB / 578.2MB Pulling
36.9s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading      362.8MB/475....
35.7s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [⠄⠄⠄] 466.3MB / 578.2MB Pulling
37.0s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading      364.4MB/475....
35.8s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [⠄⠄⠄] 466.3MB / 578.2MB Pulling
37.1s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading      364.4MB/475....
35.9s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [⠄⠄⠄] 467.9MB / 578.2MB Pulling
37.2s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading      366MB/475....
36.0s
```

```
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████]  469MB / 578.2MB Pulling
37.3s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      367.1MB/475.....
36.1s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████]  471.1MB / 578.2MB Pulling
37.4s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      369.3MB/475.....
36.2s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████]  472.7MB / 578.2MB Pulling
37.5s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      370.9MB/475.....
36.3s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████]  474.4MB / 578.2MB Pulling
37.6s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      372.5MB/475.....
36.4s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████]  474.4MB / 578.2MB Pulling
37.7s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      372.5MB/475.....
36.5s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████]  476MB / 578.2MB Pulling
37.8s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      374.1MB/475.....
36.6s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
```

```
  ⌂ sqlserver [####] 477.6MB / 578.2MB Pulling
37.9s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⌂ 3374dd749958 Downloading      375.8MB/475.....
36.7s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌂ sqlserver [####] 479.2MB / 578.2MB Pulling
38.0s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⌂ 3374dd749958 Downloading      377.4MB/475.....
36.8s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌂ sqlserver [####] 480.9MB / 578.2MB Pulling
38.1s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⌂ 3374dd749958 Downloading      379MB/475.....
36.9s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌂ sqlserver [####] 482.5MB / 578.2MB Pulling
38.2s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⌂ 3374dd749958 Downloading      380.6MB/475.....
37.0s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌂ sqlserver [####] 482.5MB / 578.2MB Pulling
38.3s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⌂ 3374dd749958 Downloading      380.6MB/475.....
37.1s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌂ sqlserver [####] 484.1MB / 578.2MB Pulling
38.4s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⌂ 3374dd749958 Downloading      382.2MB/475.....
37.2s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌂ sqlserver [####] 485.7MB / 578.2MB Pulling
38.5s
    ✓ 3d29464607d8 Pull complete
```

```
9.3s
  : 3374dd749958 Downloading      383.9MB/475....
37.3s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  : sqlserver [####] 487.3MB / 578.2MB Pulling
38.6s
  ✓ 3d29464607d8 Pull complete
9.3s
  : 3374dd749958 Downloading      385.5MB/475....
37.4s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  : sqlserver [####] 489MB / 578.2MB Pulling
38.7s
  ✓ 3d29464607d8 Pull complete
9.3s
  : 3374dd749958 Downloading      387.1MB/475....
37.5s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  : sqlserver [####] 489MB / 578.2MB Pulling
38.8s
  ✓ 3d29464607d8 Pull complete
9.3s
  : 3374dd749958 Downloading      387.1MB/475....
37.6s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  : sqlserver [####] 490.6MB / 578.2MB Pulling
38.9s
  ✓ 3d29464607d8 Pull complete
9.3s
  : 3374dd749958 Downloading      388.7MB/475....
37.7s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  : sqlserver [####] 492.2MB / 578.2MB Pulling
39.0s
  ✓ 3d29464607d8 Pull complete
9.3s
  : 3374dd749958 Downloading      390.3MB/475....
37.8s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  : sqlserver [####] 493.8MB / 578.2MB Pulling
39.1s
  ✓ 3d29464607d8 Pull complete
9.3s
  : 3374dd749958 Downloading      392MB/475....
37.9s
```

```
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 495.4MB / 578.2MB Pulling
39.2s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      393.6MB/475.....
38.0s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 497.1MB / 578.2MB Pulling
39.3s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      395.2MB/475.....
38.1s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 497.1MB / 578.2MB Pulling
39.4s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      395.2MB/475.....
38.2s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 498.7MB / 578.2MB Pulling
39.5s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      396.8MB/475.....
38.3s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 500.3MB / 578.2MB Pulling
39.6s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      398.5MB/475.....
38.4s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 501.9MB / 578.2MB Pulling
39.7s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Downloading      400.1MB/475.....
38.5s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
```

```
: sqlserver [■■■] 503.6MB / 578.2MB Pulling
39.8s
    ✓ 3d29464607d8 Pull complete
9.3s
    .. 3374dd749958 Downloading      401.7MB/475....
38.6s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
: sqlserver [■■■] 503.6MB / 578.2MB Pulling
39.9s
    ✓ 3d29464607d8 Pull complete
9.3s
    .. 3374dd749958 Downloading      401.7MB/475....
38.7s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
: sqlserver [■■■] 505.2MB / 578.2MB Pulling
40.0s
    ✓ 3d29464607d8 Pull complete
9.3s
    .. 3374dd749958 Downloading      403.3MB/475....
38.8s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
: sqlserver [■■■] 507.3MB / 578.2MB Pulling
40.1s
    ✓ 3d29464607d8 Pull complete
9.3s
    .. 3374dd749958 Downloading      405.5MB/475....
38.9s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
: sqlserver [■■■] 507.3MB / 578.2MB Pulling
40.2s
    ✓ 3d29464607d8 Pull complete
9.3s
    .. 3374dd749958 Downloading      405.5MB/475....
39.0s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
: sqlserver [■■■] 509MB / 578.2MB Pulling
40.3s
    ✓ 3d29464607d8 Pull complete
9.3s
    .. 3374dd749958 Downloading      407.1MB/475....
39.1s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
: sqlserver [■■■] 510.6MB / 578.2MB Pulling
40.4s
    ✓ 3d29464607d8 Pull complete
```

```
9.3s
  ⌈ 3374dd749958 Downloading      408.7MB/475....
39.2s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [⠄⠄⠄] 512.2MB / 578.2MB Pulling
40.5s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading      410.4MB/475....
39.3s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [⠄⠄⠄] 513.8MB / 578.2MB Pulling
40.6s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading      412MB/475....
39.4s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [⠄⠄⠄] 514.9MB / 578.2MB Pulling
40.7s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading      413.1MB/475....
39.5s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [⠄⠄⠄] 516.5MB / 578.2MB Pulling
40.8s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading      414.7MB/475....
39.6s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [⠄⠄⠄] 516.5MB / 578.2MB Pulling
40.9s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading      414.7MB/475....
39.7s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [⠄⠄⠄] 518.2MB / 578.2MB Pulling
41.0s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading      416.3MB/475....
39.8s
```

```
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [████] 519.8MB / 578.2MB Pulling
41.1s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      417.9MB/475.....
39.9s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [████] 521.4MB / 578.2MB Pulling
41.2s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      419.5MB/475.....
40.0s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [████] 523MB / 578.2MB Pulling
41.3s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      421.2MB/475.....
40.1s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [████] 523MB / 578.2MB Pulling
41.4s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      421.2MB/475.....
40.2s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [████] 524.6MB / 578.2MB Pulling
41.5s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      422.8MB/475.....
40.3s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [████] 526.3MB / 578.2MB Pulling
41.6s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      424.4MB/475.....
40.4s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
```

```
└─sqlserver [■■■] 527.9MB / 578.2MB Pulling
41.7s
  ✓ 3d29464607d8 Pull complete
9.3s
  .. 3374dd749958 Downloading          426MB/475....
40.5s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
└─sqlserver [■■■] 529.5MB / 578.2MB Pulling
41.8s
  ✓ 3d29464607d8 Pull complete
9.3s
  .. 3374dd749958 Downloading          427.7MB/475....
40.6s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
└─sqlserver [■■■] 531.1MB / 578.2MB Pulling
41.9s
  ✓ 3d29464607d8 Pull complete
9.3s
  .. 3374dd749958 Downloading          429.3MB/475....
40.7s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
└─sqlserver [■■■] 531.1MB / 578.2MB Pulling
42.0s
  ✓ 3d29464607d8 Pull complete
9.3s
  .. 3374dd749958 Downloading          429.3MB/475....
40.8s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
└─sqlserver [■■■] 532.8MB / 578.2MB Pulling
42.1s
  ✓ 3d29464607d8 Pull complete
9.3s
  .. 3374dd749958 Downloading          430.9MB/475....
40.9s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
└─sqlserver [■■■] 534.4MB / 578.2MB Pulling
42.2s
  ✓ 3d29464607d8 Pull complete
9.3s
  .. 3374dd749958 Downloading          432.5MB/475....
41.0s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
└─sqlserver [■■■] 536MB / 578.2MB Pulling
42.3s
  ✓ 3d29464607d8 Pull complete
```

```
9.3s
  ⌈ 3374dd749958 Downloading      434.1MB/475....
41.1s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [####] 537.6MB / 578.2MB Pulling
42.4s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading      435.8MB/475....
41.2s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [####] 539.2MB / 578.2MB Pulling
42.5s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading      437.4MB/475....
41.4s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [####] 539.2MB / 578.2MB Pulling
42.6s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading      437.4MB/475....
41.4s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [####] 540.9MB / 578.2MB Pulling
42.7s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading      439MB/475....
41.5s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [####] 542.5MB / 578.2MB Pulling
42.8s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading      440.6MB/475....
41.6s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [####] 544.1MB / 578.2MB Pulling
42.9s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Downloading      442.3MB/475....
41.7s
```

```
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 545.7MB / 578.2MB Pulling
43.0s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      443.9MB/475.....
41.8s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 545.7MB / 578.2MB Pulling
43.1s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      443.9MB/475.....
41.9s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 547.4MB / 578.2MB Pulling
43.2s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      445.5MB/475.....
42.0s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 549MB / 578.2MB Pulling
43.3s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      447.1MB/475.....
42.1s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 550.6MB / 578.2MB Pulling
43.4s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      448.7MB/475.....
42.2s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 552.2MB / 578.2MB Pulling
43.5s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      450.4MB/475.....
42.3s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
```

```
  ↳ sqlserver [####] 553.8MB / 578.2MB Pulling
43.6s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading          452MB/475.....
42.4s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [####] 553.8MB / 578.2MB Pulling
43.7s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading          452MB/475.....
42.5s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [####] 555.5MB / 578.2MB Pulling
43.8s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading          453.6MB/475.....
42.6s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [####] 557.1MB / 578.2MB Pulling
43.9s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading          455.2MB/475.....
42.7s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [####] 558.7MB / 578.2MB Pulling
44.0s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading          456.9MB/475.....
42.8s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [####] 560.3MB / 578.2MB Pulling
44.1s
  ✓ 3d29464607d8 Pull complete
9.3s
  ↳ 3374dd749958 Downloading          458.5MB/475.....
42.9s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [####] 560.3MB / 578.2MB Pulling
44.2s
  ✓ 3d29464607d8 Pull complete
```

```
9.3s
  ⚡ 3374dd749958 Downloading      458.5MB/475....
43.0s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [████]  562MB / 578.2MB Pulling
44.3s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      460.1MB/475....
43.1s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [████]  563.6MB / 578.2MB Pulling
44.4s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      461.7MB/475....
43.2s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [████]  565.2MB / 578.2MB Pulling
44.5s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      463.3MB/475....
43.3s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [████]  566.8MB / 578.2MB Pulling
44.6s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      465MB/475....
43.4s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [████]  568.4MB / 578.2MB Pulling
44.7s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      466.6MB/475....
43.5s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [████]  568.4MB / 578.2MB Pulling
44.8s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Downloading      466.6MB/475....
43.6s
```

```
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚠ sqlserver [██████] 570.1MB / 578.2MB Pulling
44.9s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⌂ 3374dd749958 Downloading      468.2MB/475.....
43.7s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚠ sqlserver [██████] 571.7MB / 578.2MB Pulling
45.0s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⌂ 3374dd749958 Downloading      469.8MB/475.....
43.8s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚠ sqlserver [██████] 571.7MB / 578.2MB Pulling
45.1s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⌂ 3374dd749958 Downloading      469.8MB/475.....
43.9s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚠ sqlserver [██████] 573.8MB / 578.2MB Pulling
45.2s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⌂ 3374dd749958 Downloading      472MB/475.....
44.0s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚠ sqlserver [██████] 575.5MB / 578.2MB Pulling
45.3s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⌂ 3374dd749958 Downloading      473.6MB/475.....
44.1s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚠ sqlserver [██████] 577.1MB / 578.2MB Pulling
45.4s
    ✓ 3d29464607d8 Pull complete
9.3s
  ⌂ 3374dd749958 Extracting      557.1kB/475.4...
44.2s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
```

```
..: sqlserver [####] 577.1MB / 578.2MB Pulling
45.5s
    ✓ 3d29464607d8 Pull complete
9.3s
    ..: 3374dd749958 Extracting      7.799MB/475.4...
44.3s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
..: sqlserver [####] 577.1MB / 578.2MB Pulling
45.6s
    ✓ 3d29464607d8 Pull complete
9.3s
    ..: 3374dd749958 Extracting      13.93MB/475.4...
44.4s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
..: sqlserver [####] 577.1MB / 578.2MB Pulling
45.7s
    ✓ 3d29464607d8 Pull complete
9.3s
    ..: 3374dd749958 Extracting      18.38MB/475.4...
44.5s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
..: sqlserver [####] 577.1MB / 578.2MB Pulling
45.8s
    ✓ 3d29464607d8 Pull complete
9.3s
    ..: 3374dd749958 Extracting      18.38MB/475.4...
44.6s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
..: sqlserver [####] 577.1MB / 578.2MB Pulling
45.9s
    ✓ 3d29464607d8 Pull complete
9.3s
    ..: 3374dd749958 Extracting      26.74MB/475.4...
44.7s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
..: sqlserver [####] 577.1MB / 578.2MB Pulling
46.0s
    ✓ 3d29464607d8 Pull complete
9.3s
    ..: 3374dd749958 Extracting      33.42MB/475.4...
44.8s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
..: sqlserver [####] 577.1MB / 578.2MB Pulling
46.1s
    ✓ 3d29464607d8 Pull complete
```

```
9.3s
  ⚡ 3374dd749958 Extracting      41.78MB/475.4...
44.9s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [████] 577.1MB / 578.2MB Pulling
46.2s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Extracting      47.35MB/475.4...
45.0s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [████] 577.1MB / 578.2MB Pulling
46.3s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Extracting      54.03MB/475.4...
45.1s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [████] 577.1MB / 578.2MB Pulling
46.4s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Extracting      61.28MB/475.4...
45.2s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [████] 577.1MB / 578.2MB Pulling
46.5s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Extracting      65.18MB/475.4...
45.3s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [████] 577.1MB / 578.2MB Pulling
46.6s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Extracting      70.19MB/475.4...
45.4s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [████] 577.1MB / 578.2MB Pulling
46.7s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Extracting      78.54MB/475.4...
45.5s
```

```
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 577.1MB / 578.2MB Pulling
46.8s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Extracting      99.71MB/475.4...
45.6s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 577.1MB / 578.2MB Pulling
46.9s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Extracting      108.1MB/475.4...
45.7s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 577.1MB / 578.2MB Pulling
47.0s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Extracting      115.9MB/475.4...
45.8s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 577.1MB / 578.2MB Pulling
47.1s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Extracting      124.2MB/475.4...
45.9s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 577.1MB / 578.2MB Pulling
47.2s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Extracting      129.8MB/475.4...
46.0s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 577.1MB / 578.2MB Pulling
47.3s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Extracting      129.8MB/475.4...
46.1s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
```

```
  ↳ sqlserver [####] 577.1MB / 578.2MB Pulling
47.4s
    ✓ 3d29464607d8 Pull complete
9.3s
    ↳ 3374dd749958 Extracting      136.5MB/475.4...
46.2s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [####] 577.1MB / 578.2MB Pulling
47.5s
    ✓ 3d29464607d8 Pull complete
9.3s
    ↳ 3374dd749958 Extracting      143.7MB/475.4...
46.3s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [####] 577.1MB / 578.2MB Pulling
47.6s
    ✓ 3d29464607d8 Pull complete
9.3s
    ↳ 3374dd749958 Extracting      152.6MB/475.4...
46.4s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [####] 577.1MB / 578.2MB Pulling
47.7s
    ✓ 3d29464607d8 Pull complete
9.3s
    ↳ 3374dd749958 Extracting      164.9MB/475.4...
46.5s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [####] 577.1MB / 578.2MB Pulling
47.8s
    ✓ 3d29464607d8 Pull complete
9.3s
    ↳ 3374dd749958 Extracting      177.1MB/475.4...
46.6s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [####] 577.1MB / 578.2MB Pulling
47.9s
    ✓ 3d29464607d8 Pull complete
9.3s
    ↳ 3374dd749958 Extracting      182.2MB/475.4...
46.7s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ↳ sqlserver [####] 577.1MB / 578.2MB Pulling
48.0s
    ✓ 3d29464607d8 Pull complete
```

```
9.3s
  ⌂ 3374dd749958 Extracting      184.9MB/475.4...
46.8s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌂ sqlserver [####] 577.1MB / 578.2MB Pulling
48.1s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌂ 3374dd749958 Extracting      190.5MB/475.4...
46.9s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌂ sqlserver [####] 577.1MB / 578.2MB Pulling
48.2s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌂ 3374dd749958 Extracting      197.2MB/475.4...
47.0s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌂ sqlserver [####] 577.1MB / 578.2MB Pulling
48.3s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌂ 3374dd749958 Extracting      206.1MB/475.4...
47.1s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌂ sqlserver [####] 577.1MB / 578.2MB Pulling
48.4s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌂ 3374dd749958 Extracting      208.9MB/475.4...
47.2s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌂ sqlserver [####] 577.1MB / 578.2MB Pulling
48.5s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌂ 3374dd749958 Extracting      211.7MB/475.4...
47.3s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌂ sqlserver [####] 577.1MB / 578.2MB Pulling
48.6s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌂ 3374dd749958 Extracting      215MB/475.4...
47.4s
```

```
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚒ sqlserver [██████] 577.1MB / 578.2MB Pulling
48.7s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚒ 3374dd749958 Extracting      222.8MB/475.4...
47.5s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚒ sqlserver [██████] 577.1MB / 578.2MB Pulling
48.8s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚒ 3374dd749958 Extracting      231.7MB/475.4...
47.6s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚒ sqlserver [██████] 577.1MB / 578.2MB Pulling
48.9s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚒ 3374dd749958 Extracting      240.1MB/475.4...
47.7s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚒ sqlserver [██████] 577.1MB / 578.2MB Pulling
49.0s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚒ 3374dd749958 Extracting      249MB/475.4...
47.8s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚒ sqlserver [██████] 577.1MB / 578.2MB Pulling
49.1s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚒ 3374dd749958 Extracting      257.9MB/475.4...
47.9s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚒ sqlserver [██████] 577.1MB / 578.2MB Pulling
49.2s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚒ 3374dd749958 Extracting      257.9MB/475.4...
48.0s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
```

```
: sqlserver [####] 577.1MB / 578.2MB Pulling
49.3s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Extracting      265.2MB/475.4...
48.1s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
: sqlserver [####] 577.1MB / 578.2MB Pulling
49.4s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Extracting      265.2MB/475.4...
48.2s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
: sqlserver [####] 577.1MB / 578.2MB Pulling
49.5s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Extracting      272.4MB/475.4...
48.3s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
: sqlserver [####] 577.1MB / 578.2MB Pulling
49.6s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Extracting      279.6MB/475.4...
48.4s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
: sqlserver [####] 577.1MB / 578.2MB Pulling
49.7s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Extracting      288.6MB/475.4...
48.5s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
: sqlserver [####] 577.1MB / 578.2MB Pulling
49.8s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Extracting      299.1MB/475.4...
48.6s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
: sqlserver [####] 577.1MB / 578.2MB Pulling
49.9s
  ✓ 3d29464607d8 Pull complete
```

```
9.3s
  ⌈ 3374dd749958 Extracting      306.4MB/475.4...
48.7s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [####] 577.1MB / 578.2MB Pulling
50.0s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Extracting      312.5MB/475.4...
48.8s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [####] 577.1MB / 578.2MB Pulling
50.1s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Extracting      319.8MB/475.4...
48.9s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [####] 577.1MB / 578.2MB Pulling
50.2s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Extracting      326.4MB/475.4...
49.0s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [####] 577.1MB / 578.2MB Pulling
50.3s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Extracting      333.1MB/475.4...
49.1s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [####] 577.1MB / 578.2MB Pulling
50.4s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Extracting      341.5MB/475.4...
49.2s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [####] 577.1MB / 578.2MB Pulling
50.5s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Extracting      349.8MB/475.4...
49.3s
```

```
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 577.1MB / 578.2MB Pulling
50.6s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Extracting      357.6MB/475.4...
49.4s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 577.1MB / 578.2MB Pulling
50.7s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Extracting      362.6MB/475.4...
49.5s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 577.1MB / 578.2MB Pulling
50.8s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Extracting      370.4MB/475.4...
49.6s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 577.1MB / 578.2MB Pulling
50.9s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Extracting      378.8MB/475.4...
49.7s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 577.1MB / 578.2MB Pulling
51.0s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Extracting      386MB/475.4...
49.8s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 577.1MB / 578.2MB Pulling
51.1s
    ✓ 3d29464607d8 Pull complete
9.3s
    ⚡ 3374dd749958 Extracting      395MB/475.4...
49.9s
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
```

```
 sqlserver [####] 577.1MB / 578.2MB Pulling
51.2s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Extracting      403.3MB/475.4...
50.0s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  sqlserver [####] 577.1MB / 578.2MB Pulling
51.3s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Extracting      408.3MB/475.4...
50.1s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  sqlserver [####] 577.1MB / 578.2MB Pulling
51.4s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Extracting      416.7MB/475.4...
50.2s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  sqlserver [####] 577.1MB / 578.2MB Pulling
51.5s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Extracting      416.7MB/475.4...
50.3s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  sqlserver [####] 577.1MB / 578.2MB Pulling
51.6s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Extracting      425MB/475.4...
50.4s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  sqlserver [####] 577.1MB / 578.2MB Pulling
51.7s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⚡ 3374dd749958 Extracting      431.7MB/475.4...
50.5s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  sqlserver [####] 577.1MB / 578.2MB Pulling
51.8s
  ✓ 3d29464607d8 Pull complete
```

```
9.3s
  ⌈ 3374dd749958 Extracting      439.5MB/475.4...
50.6s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [####] 577.1MB / 578.2MB Pulling
51.9s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Extracting      447.9MB/475.4...
50.7s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [####] 577.1MB / 578.2MB Pulling
52.0s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Extracting      454MB/475.4...
50.8s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [####] 577.1MB / 578.2MB Pulling
52.1s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Extracting      461.8MB/475.4...
50.9s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [####] 577.1MB / 578.2MB Pulling
52.2s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Extracting      470.2MB/475.4...
51.0s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⌈ sqlserver [####] 577.3MB / 578.2MB Pulling
52.3s
  ✓ 3d29464607d8 Pull complete
9.3s
  ⌈ 3374dd749958 Extracting      475.4MB/475.4...
51.1s
  ✓ 18952140cc7b Download complete
14.4s
[+] Running 3/4
  ⌈ sqlserver [####] 577.3MB / 578.2MB Pulling
52.4s
  ✓ 3d29464607d8 Pull complete
9.3s
  ✓ 3374dd749958 Pull complete
51.2s
```

```
    ✓ 18952140cc7b Download complete
14.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 577.3MB / 578.2MB Pulling
52.5s
    ✓ 3d29464607d8 Pull complete
9.3s
    ✓ 3374dd749958 Pull complete
51.2s
  ⚡ 18952140cc7b Extracting  557.1kB/71.23MB
51.3s
[+] Running 2/4
  ⚡ sqlserver [██████] 577.3MB / 578.2MB Pulling
52.6s
    ✓ 3d29464607d8 Pull complete
9.3s
    ✓ 3374dd749958 Pull complete
51.2s
  ⚡ 18952140cc7b Extracting  8.356MB/71.23MB
51.4s
[+] Running 2/4
  ⚡ sqlserver [██████] 577.3MB / 578.2MB Pulling
52.7s
    ✓ 3d29464607d8 Pull complete
9.3s
    ✓ 3374dd749958 Pull complete
51.2s
  ⚡ 18952140cc7b Extracting  21.73MB/71.23MB
51.5s
[+] Running 2/4
  ⚡ sqlserver [██████] 577.3MB / 578.2MB Pulling
52.8s
    ✓ 3d29464607d8 Pull complete
9.3s
    ✓ 3374dd749958 Pull complete
51.2s
  ⚡ 18952140cc7b Extracting  26.74MB/71.23MB
51.6s
[+] Running 2/4
  ⚡ sqlserver [██████] 577.3MB / 578.2MB Pulling
52.9s
    ✓ 3d29464607d8 Pull complete
9.3s
    ✓ 3374dd749958 Pull complete
51.2s
  ⚡ 18952140cc7b Extracting  26.74MB/71.23MB
51.7s
[+] Running 2/4
  ⚡ sqlserver [██████] 577.3MB / 578.2MB Pulling
53.0s
    ✓ 3d29464607d8 Pull complete
9.3s
    ✓ 3374dd749958 Pull complete
51.2s
  ⚡ 18952140cc7b Extracting  30.64MB/71.23MB
51.8s
[+] Running 2/4
```

```
  ▒ sqlserver [██████] 577.3MB / 578.2MB Pulling
53.1s
    ✓ 3d29464607d8 Pull complete
9.3s
    ✓ 3374dd749958 Pull complete
51.2s
    ▒ 18952140cc7b Extracting 33.42MB/71.23MB
51.9s
[+] Running 2/4
  ▒ sqlserver [██████] 577.3MB / 578.2MB Pulling
53.2s
    ✓ 3d29464607d8 Pull complete
9.3s
    ✓ 3374dd749958 Pull complete
51.2s
    ▒ 18952140cc7b Extracting 36.77MB/71.23MB
52.0s
[+] Running 2/4
  ▒ sqlserver [██████] 577.3MB / 578.2MB Pulling
53.3s
    ✓ 3d29464607d8 Pull complete
9.3s
    ✓ 3374dd749958 Pull complete
51.2s
    ▒ 18952140cc7b Extracting 46.79MB/71.23MB
52.1s
[+] Running 2/4
  ▒ sqlserver [██████] 577.3MB / 578.2MB Pulling
53.4s
    ✓ 3d29464607d8 Pull complete
9.3s
    ✓ 3374dd749958 Pull complete
51.2s
    ▒ 18952140cc7b Extracting 56.82MB/71.23MB
52.2s
[+] Running 2/4
  ▒ sqlserver [██████] 577.3MB / 578.2MB Pulling
53.5s
    ✓ 3d29464607d8 Pull complete
9.3s
    ✓ 3374dd749958 Pull complete
51.2s
    ▒ 18952140cc7b Extracting 61.83MB/71.23MB
52.3s
[+] Running 2/4
  ▒ sqlserver [██████] 577.3MB / 578.2MB Pulling
53.6s
    ✓ 3d29464607d8 Pull complete
9.3s
    ✓ 3374dd749958 Pull complete
51.2s
    ▒ 18952140cc7b Extracting 63.5MB/71.23MB
52.4s
[+] Running 2/4
  ▒ sqlserver [██████] 577.3MB / 578.2MB Pulling
53.7s
    ✓ 3d29464607d8 Pull complete
```

```
9.3s
  ✓ 3374dd749958 Pull complete
51.2s
  ⌈ 18952140cc7b Extracting  63.5MB/71.23MB
52.5s
[+] Running 2/4
  ⌈ sqlserver [####] 577.3MB / 578.2MB Pulling
53.8s
  ✓ 3d29464607d8 Pull complete
9.3s
  ✓ 3374dd749958 Pull complete
51.2s
  ⌈ 18952140cc7b Extracting  67.4MB/71.23MB
52.6s
[+] Running 2/4
  ⌈ sqlserver [####] 577.3MB / 578.2MB Pulling
53.9s
  ✓ 3d29464607d8 Pull complete
9.3s
  ✓ 3374dd749958 Pull complete
51.2s
  ⌈ 18952140cc7b Extracting  67.4MB/71.23MB
52.7s
[+] Running 2/4
  ⌈ sqlserver [####] 577.3MB / 578.2MB Pulling
54.0s
  ✓ 3d29464607d8 Pull complete
9.3s
  ✓ 3374dd749958 Pull complete
51.2s
  ⌈ 18952140cc7b Extracting  69.07MB/71.23MB
52.8s
[+] Running 2/4
  ⌈ sqlserver [####] 577.3MB / 578.2MB Pulling
54.1s
  ✓ 3d29464607d8 Pull complete
9.3s
  ✓ 3374dd749958 Pull complete
51.2s
  ⌈ 18952140cc7b Extracting  69.07MB/71.23MB
52.9s
[+] Running 2/4
  ⌈ sqlserver [####] 577.3MB / 578.2MB Pulling
54.2s
  ✓ 3d29464607d8 Pull complete
9.3s
  ✓ 3374dd749958 Pull complete
51.2s
  ⌈ 18952140cc7b Extracting  69.07MB/71.23MB
53.0s
[+] Running 2/4
  ⌈ sqlserver [####] 577.3MB / 578.2MB Pulling
54.3s
  ✓ 3d29464607d8 Pull complete
9.3s
  ✓ 3374dd749958 Pull complete
51.2s
```

```
: 18952140cc7b Extracting 70.19MB/71.23MB
53.1s
[+] Running 2/4
  ⚡ sqlserver [██████] 577.3MB / 578.2MB Pulling
54.4s
  ✓ 3d29464607d8 Pull complete
9.3s
  ✓ 3374dd749958 Pull complete
51.2s
  ⚡ 18952140cc7b Extracting 70.19MB/71.23MB
53.2s
[+] Running 2/4
  ⚡ sqlserver [██████] 578.2MB / 578.2MB Pulling
54.5s
  ✓ 3d29464607d8 Pull complete
9.3s
  ✓ 3374dd749958 Pull complete
51.2s
  ⚡ 18952140cc7b Extracting 71.23MB/71.23MB
53.3s
[+] Running 4/4
  ✓ sqlserver Pulled
54.6s
  ✓ 3d29464607d8 Pull complete
9.3s
  ✓ 3374dd749958 Pull complete
51.2s
  ✓ 18952140cc7b Pull complete
53.4s
[+] Running 2/0
  ✓ Container eshoponweb-eshoppublicapi-1 Stopped
0.0s
  ✓ Container eshoponweb-eshopwebmvc-1 Stopped
0.0s
  ⚡ Container eshoponweb-sqlserver-1 Recreate
0.0s
[+] Running 2/3
  ✓ Container eshoponweb-eshoppublicapi-1 Stopped
0.0s
  ✓ Container eshoponweb-eshopwebmvc-1 Stopped
0.0s
  ⚡ Container eshoponweb-sqlserver-1 Recreate
0.1s
[+] Running 2/3
  ✓ Container eshoponweb-eshoppublicapi-1 Stopped
0.0s
  ✓ Container eshoponweb-eshopwebmvc-1 Stopped
0.0s
  ⚡ Container eshoponweb-sqlserver-1 Recreate
0.2s
[+] Running 3/3
  ✓ Container eshoponweb-eshoppublicapi-1
Created 0.0s
  ⚡ Container eshoponweb-eshopwebmvc-1
Recreate 0.3s
  ✓ Container eshoponweb-sqlserver-1
Recreated 0.3s
```

```
! sqlserver The requested image's platform (linux/amd64) does not match
the detected host platform (linux/arm64/v8) and no specific platform was
requested 0.0s
[+] Running 3/4
✓ Container eshoponweb-eshoppublicapi-1
Created 0.0s
  ! Container eshoponweb-eshopwebmvc-1
Recreate 0.4s
✓ Container eshoponweb-sqlserver-1
Recreated 0.3s
! sqlserver The requested image's platform (linux/amd64) does not match
the detected host platform (linux/arm64/v8) and no specific platform was
requested 0.0s
[+] Running 4/4
✓ Container eshoponweb-eshoppublicapi-1
Created 0.0s
✓ Container eshoponweb-eshopwebmvc-1
Recreated 0.4s
✓ Container eshoponweb-sqlserver-1
Recreated 0.3s
! sqlserver The requested image's platform (linux/amd64) does not match
the detected host platform (linux/arm64/v8) and no specific platform was
requested 0.0s
Attaching to eshoppublicapi-1, eshopwebmvc-1, sqlserver-1
sqlserver-1 | SQL Server 2022 will run as non-root by default.
sqlserver-1 | This container is running as user mssql.
sqlserver-1 | To learn more visit https://go.microsoft.com/fwlink/?linkid=2099216.
eshopwebmvc-1 | Unhandled exception. System.UriFormatException: Invalid URI: The URI is empty.
eshopwebmvc-1 |     at System.Uri.CreateThis(String uri, Boolean dontEscape, UriKind uriKind, UriCreationOptions& creationOptions)
eshopwebmvc-1 |     at System.Uri..ctor(String uriString)
eshopwebmvc-1 |     at Microsoft.eShopWeb.Web.Extensions.ServiceCollectionExtensions.AddDatabaseContexts(IServiceCollection services, IWebHostEnvironment environment, ConfigurationManager configuration) in /app/src/Web/Extensions/ServiceCollectionExtensions.cs:line 28
eshopwebmvc-1 |     at Program.<Main>$(<Main>String[] args) in /app/src/Web/Program.cs:line 16
eshopwebmvc-1 |     at Program.<Main>(String[] args)
eshoppublicapi-1 | fail: PublicApi[0]
eshoppublicapi-1 |     An error was generated for warning 'Microsoft.EntityFrameworkCore.Migrations.PendingModelChangesWarning': The model for context 'CatalogContext' has pending changes. Add a new migration before updating the database. This exception can be suppressed or logged by passing event ID 'RelationalEventId.PendingModelChangesWarning' to the 'ConfigureWarnings' method in 'DbContext.OnConfiguring' or 'AddDbContext'.
eshoppublicapi-1 | fail: PublicApi[0]
eshoppublicapi-1 |     An error was generated for warning 'Microsoft.EntityFrameworkCore.Migrations.PendingModelChangesWarning': The model for context 'CatalogContext' has pending changes. Add a new migration before updating the database. This exception can be suppressed or logged by passing event ID 'RelationalEventId.PendingModelChangesWarning' to the 'ConfigureWarnings' method in 'DbContext.OnConfiguring' or 'AddDbContext'.
eshoppublicapi-1 | fail: PublicApi[0]
```

```
eshoppublicapi-1 |      An error was generated for warning 'Microsoft.EntityFrameworkCore.Migrations.PendingModelChangesWarning': The model for context 'CatalogContext' has pending changes. Add a new migration before updating the database. This exception can be suppressed or logged by passing event ID 'RelationalEventId.PendingModelChangesWarning' to the 'ConfigureWarnings' method in 'DbContext.OnConfiguring' or 'AddDbContext'.
eshoppublicapi-1 | fail: PublicApi[0]
eshoppublicapi-1 |      An error was generated for warning 'Microsoft.EntityFrameworkCore.Migrations.PendingModelChangesWarning': The model for context 'CatalogContext' has pending changes. Add a new migration before updating the database. This exception can be suppressed or logged by passing event ID 'RelationalEventId.PendingModelChangesWarning' to the 'ConfigureWarnings' method in 'DbContext.OnConfiguring' or 'AddDbContext'.
eshoppublicapi-1 | fail: PublicApi[0]
eshoppublicapi-1 |      An error was generated for warning 'Microsoft.EntityFrameworkCore.Migrations.PendingModelChangesWarning': The model for context 'CatalogContext' has pending changes. Add a new migration before updating the database. This exception can be suppressed or logged by passing event ID 'RelationalEventId.PendingModelChangesWarning' to the 'ConfigureWarnings' method in 'DbContext.OnConfiguring' or 'AddDbContext'.
eshoppublicapi-1 | fail: PublicApi[0]
eshoppublicapi-1 |      An error was generated for warning 'Microsoft.EntityFrameworkCore.Migrations.PendingModelChangesWarning': The model for context 'CatalogContext' has pending changes. Add a new migration before updating the database. This exception can be suppressed or logged by passing event ID 'RelationalEventId.PendingModelChangesWarning' to the 'ConfigureWarnings' method in 'DbContext.OnConfiguring' or 'AddDbContext'.
eshoppublicapi-1 | fail: PublicApi[0]
eshoppublicapi-1 |      An error was generated for warning 'Microsoft.EntityFrameworkCore.Migrations.PendingModelChangesWarning': The model for context 'CatalogContext' has pending changes. Add a new migration before updating the database. This exception can be suppressed or logged by passing event ID 'RelationalEventId.PendingModelChangesWarning' to the 'ConfigureWarnings' method in 'DbContext.OnConfiguring' or 'AddDbContext'.
eshoppublicapi-1 | fail: PublicApi[0]
eshoppublicapi-1 |      An error was generated for warning 'Microsoft.EntityFrameworkCore.Migrations.PendingModelChangesWarning': The model for context 'CatalogContext' has pending changes. Add a new migration before updating the database. This exception can be suppressed or logged by passing event ID 'RelationalEventId.PendingModelChangesWarning' to the 'ConfigureWarnings' method in 'DbContext.OnConfiguring' or 'AddDbContext'.
eshoppublicapi-1 | fail: PublicApi[0]
eshoppublicapi-1 |      An error was generated for warning 'Microsoft.EntityFrameworkCore.Migrations.PendingModelChangesWarning': The model for context 'CatalogContext' has pending changes. Add a new migration before updating the database. This exception can be suppressed or logged by passing event ID 'RelationalEventId.PendingModelChangesWarning' to the 'ConfigureWarnings' method in 'DbContext.OnConfiguring' or 'AddDbContext'.
eshoppublicapi-1 | fail: PublicApi[0]
eshoppublicapi-1 |      An error was generated for warning 'Microsoft.EntityFrameworkCore.Migrations.PendingModelChangesWarning': The model for context 'CatalogContext' has pending changes. Add a new migration before updating the database. This exception can be suppressed or logged by passing event ID 'RelationalEventId.PendingModelChangesWarning' to the 'ConfigureWarnings' method in 'DbContext.OnConfiguring' or 'AddDbContext'.
```

```
EntityFrameworkCore.Migrations.PendingModelChangesWarning': The model for context 'CatalogContext' has pending changes. Add a new migration before updating the database. This exception can be suppressed or logged by passing event ID 'RelationalEventId.PendingModelChangesWarning' to the 'ConfigureWarnings' method in 'DbContext.OnConfiguring' or 'AddDbContext'.
eshoppublicapi-1 | fail: PublicApi[0]
eshoppublicapi-1 |     An error occurred seeding the DB.
eshoppublicapi-1 |     System.InvalidOperationException: An error was generated for warning 'Microsoft.EntityFrameworkCore.Migrations.PendingModelChangesWarning': The model for context 'CatalogContext' has pending changes. Add a new migration before updating the database. This exception can be suppressed or logged by passing event ID 'RelationalEventId.PendingModelChangesWarning' to the 'ConfigureWarnings' method in 'DbContext.OnConfiguring' or 'AddDbContext'.
eshoppublicapi-1 |         at Microsoft.EntityFrameworkCore.Diagnostics.EventDefinition`1.Log[TLoggerCategory](IDiagnosticsLogger`1 logger, TParam arg)
eshoppublicapi-1 |             at Microsoft.EntityFrameworkCore.Diagnostics.RelationalLoggerExtensions.PendingModelChangesWarning(IDiagnosticsLogger`1 diagnostics, Type contextType)
eshoppublicapi-1 |                 at Microsoft.EntityFrameworkCore.Migrations.Internal.Migrator.Migrate(String targetMigration)
eshoppublicapi-1 |                     at Microsoft.EntityFrameworkCore.RelationalDatabaseFacadeExtensions.Migrate(DatabaseFacade databaseFacade)
eshoppublicapi-1 |                         at Microsoft.eShopWeb.Infrastructure.Data.CatalogContextSeed.SeedAsync(CatalogContext catalogContext, ILogger logger, Int32 retry) in /app/src/Infrastructure/Data/CatalogContextSeed.cs:line 21
eshoppublicapi-1 |                 at Microsoft.eShopWeb.Infrastructure.Data.CatalogContextSeed.SeedAsync(CatalogContext catalogContext, ILogger logger, Int32 retry) in /app/src/Infrastructure/Data/CatalogContextSeed.cs:line 50
eshoppublicapi-1 |                     at Microsoft.eShopWeb.Infrastructure.Data.CatalogContextSeed.SeedAsync(CatalogContext catalogContext, ILogger logger, Int32 retry) in /app/src/Infrastructure/Data/CatalogContextSeed.cs:line 55
eshoppublicapi-1 |                         at Microsoft.eShopWeb.Infrastructure.Data.CatalogContextSeed.SeedAsync(CatalogContext catalogContext, ILogger logger, Int32 retry) in /app/src/Infrastructure/Data/CatalogContextSeed.cs:line 55
eshoppublicapi-1 |                             at Microsoft.eShopWeb.Infrastructure.Data.CatalogContextSeed.SeedAsync(CatalogContext catalogContext, ILogger logger, Int32 retry) in /app/src/Infrastructure/Data/CatalogContextSeed.cs:line 55
eshoppublicapi-1 |                                 at Microsoft.eShopWeb.Infrastructure.Data.CatalogContextSeed.SeedAsync(CatalogContext catalogContext, ILogger logger, Int32 retry) in /app/src/Infrastructure/Data/CatalogContextSeed.cs:line 55
eshoppublicapi-1 |                                     at Microsoft.eShopWeb.Infrastructure.Data.CatalogContextSeed.SeedAsync(CatalogContext catalogContext, ILogger logger, Int32 retry) in /app/src/Infrastructure/Data/CatalogContextSeed.cs:line 55
```

```
eshoppublicapi-1 |          at Microsoft.eShopWeb.Infrastructure.Data.CatalogContextSeed.SeedAsync(CatalogContext catalogContext, ILogger logger, Int32 retry) in /app/src/Infrastructure/Data/CatalogContextSeed.cs:line 55
eshoppublicapi-1 |          at Microsoft.eShopWeb.Infrastructure.Data.CatalogContextSeed.SeedAsync(CatalogContext catalogContext, ILogger logger, Int32 retry) in /app/src/Infrastructure/Data/CatalogContextSeed.cs:line 55
eshoppublicapi-1 |          at Microsoft.eShopWeb.Infrastructure.Data.CatalogContextSeed.SeedAsync(CatalogContext catalogContext, ILogger logger, Int32 retry) in /app/src/Infrastructure/Data/CatalogContextSeed.cs:line 55
eshoppublicapi-1 |          at Microsoft.eShopWeb.Infrastructure.Data.CatalogContextSeed.SeedAsync(CatalogContext catalogContext, ILogger logger, Int32 retry) in /app/src/Infrastructure/Data/CatalogContextSeed.cs:line 55
eshoppublicapi-1 |          at Microsoft.eShopWeb.PublicApi.Extensions.WebApplicationExtensions.SeedDatabaseAsync(WebApplication app) in /app/src/PublicApi/Extensions/WebApplicationExtensions.cs:line 23
eshoppublicapi-1 | warn: Microsoft.AspNetCore.DataProtection.Repositories.FileSystemXmlRepository[60]
eshoppublicapi-1 |      Storing keys in a directory '/root/.aspnet/DataProtection-Keys' that may not be persisted outside of the container. Protected data will be unavailable when container is destroyed. For more information go to https://aka.ms/aspnet/dataprotectionwarning
sqlserver-1 | 2024-11-17 19:40:09.92 Server      Setup step is copying system data file 'C:\templatedata\master.mdf' to '/var/opt/mssql/data/master.mdf'.
2024-11-17 19:40:09.97 Server      Did not find an existing master data file /var/opt/mssql/data/master.mdf, copying the missing default master and other system database files. If you have moved the database location, but not moved the database files, startup may fail. To repair: shutdown SQL Server, move the master database to configured location, and restart.
2024-11-17 19:40:09.97 Server      Setup step is copying system data file 'C:\templatedata\mastlog.ldf' to '/var/opt/mssql/data/mastlog.ldf'.
2024-11-17 19:40:09.98 Server      Setup step is copying system data file 'C:\templatedata\model.mdf' to '/var/opt/mssql/data/model.mdf'.
2024-11-17 19:40:09.99 Server      Setup step is copying system data file 'C:\templatedata\modellog.ldf' to '/var/opt/mssql/data/modellog.ldf'.
2024-11-17 19:40:10.00 Server      Setup step is copying system data file 'C:\templatedata\msdbdata.mdf' to '/var/opt/mssql/data/msdbdata.mdf'.
2024-11-17 19:40:10.01 Server      Setup step is copying system data file 'C:\templatedata\msdblog.ldf' to '/var/opt/mssql/data/msdblog.ldf'.
2024-11-17 19:40:10.02 Server      Setup step is FORCE copying system data file 'C:\templatedata\model_replicatedmaster.mdf' to '/var/opt/mssql/data/model_replicatedmaster.mdf'.
2024-11-17 19:40:10.03 Server      Setup step is FORCE copying system data file 'C:\templatedata\model_replicatedmaster.ldf' to '/var/opt/mssql/data/model_replicatedmaster.ldf'.
2024-11-17 19:40:10.03 Server      Setup step is FORCE copying system data file 'C:\templatedata\model_msdbdata.mdf' to '/var/opt/mssql/data/model_msdbdata.mdf'.
2024-11-17 19:40:10.06 Server      Setup step is FORCE copying system data file 'C:\templatedata\model_msdblog.ldf' to '/var/opt/mssql/data/model_msdblog.ldf'.
2024-11-17 19:40:10.10 Server      Microsoft SQL Server 2022 (RTM-CU16)
```

(KB5048033) - 16.0.4165.4 (X64)
Nov 6 2024 19:24:49 |
Copyright (C) 2022 Microsoft Corporation
Developer Edition (64-bit) on Linux (Ubuntu 22.04.5 LTS) <X64>
2024-11-17 19:40:10.11 Server UTC adjustment: 0:00
2024-11-17 19:40:10.11 Server (c) Microsoft Corporation.
2024-11-17 19:40:10.11 Server All rights reserved.
2024-11-17 19:40:10.12 Server Server process ID is 416.
2024-11-17 19:40:10.12 Server Logging SQL Server messages in file
'/var/opt/mssql/log/errorlog'.
2024-11-17 19:40:10.12 Server Registry startup parameters:
-d /var/opt/mssql/data/master.mdf
-l /var/opt/mssql/data/mastlog.ldf
-e /var/opt/mssql/log/errorlog
2024-11-17 19:40:10.13 Server SQL Server detected 1 sockets with 4 cores per socket and 4 logical processors per socket, 4 total logical processors; using 4 logical processors based on SQL Server licensing. This is an informational message; no user action is required.
2024-11-17 19:40:10.13 Server SQL Server is starting at normal priority base (=7). This is an informational message only. No user action is required.
2024-11-17 19:40:10.13 Server Detected 3136 MB of RAM, 2659 MB of available memory, 2659 MB of available page file. This is an informational message; no user action is required.
2024-11-17 19:40:10.14 Server Using conventional memory in the memory manager.
2024-11-17 19:40:10.14 Server Detected pause instruction latency: 0 cycles.
2024-11-17 19:40:10.14 Server SQL Server detected the following NUMA node configuration (NUMA Node number 0, Processor Group number 0, CPU Mask 0x000000000000000f).
2024-11-17 19:40:10.15 Server Page exclusion bitmap is enabled.
2024-11-17 19:40:10.18 Server Buffer pool extension is not supported on Linux platform.
2024-11-17 19:40:10.19 Server Buffer Pool: Allocating 4194304 bytes for 417211 hashPages.
2024-11-17 19:40:10.40 Server Buffer pool extension is already disabled. No action is necessary.
2024-11-17 19:40:11.17 Server Installing Client TLS certificates to the store.
2024-11-17 19:40:11.18 Server CPU vectorization level(s) detected: SSE SSE2 SSE3 SSSE3 SSE41 SSE42 POPCNT
2024-11-17 19:40:11.26 Server Successfully initialized the TLS configuration. Allowed TLS protocol versions are ['1.0 1.1 1.2']. Allowed TLS ciphers are ['ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-ECDSA-AES256-SHA:ECDHE-ECDSA-AES128-SHA:AES256-GCM-SHA384:AES128-GCM-SHA256:AES256-SHA256:AES128-SHA256:AES256-SHA:AES128-SHA:!DHE-RSA-AES256-GCM-SHA384:!DHE-RSA-AES128-GCM-SHA256:!DHE-RSA-AES256-SHA:!DHE-RSA-AES128-SHA'].
2024-11-17 19:40:11.34 Server Query Store settings initialized with enabled = 1,
SetThreadAffinity failed: 87m
2024-11-17 19:40:11.38 Server The maximum number of dedicated administrator connections for this instance is '1'
2024-11-17 19:40:11.38 Server Node configuration: node 0: CPU mask: 0x000000000000000f:0 Active CPU mask: 0x000000000000000f:0. This message

provides a description of the NUMA configuration for this computer. This is an informational message only. No user action is required.

2024-11-17 19:40:11.40 Server Using dynamic lock allocation. Initial allocation of 2500 Lock blocks and 5000 Lock Owner blocks per node. This is an informational message only. No user action is required.

2024-11-17 19:40:11.48 Server In-Memory OLTP initialized on lowend machine.

2024-11-17 19:40:11.54 Server CLR version v4.0.30319 loaded.

2024-11-17 19:40:11.63 Server [INFO] Created Extended Events session 'hkenginexesession'

2024-11-17 19:40:11.64 Server Database Instant File Initialization: enabled. For security and performance considerations see the topic 'Database Instant File Initialization' in SQL Server Books Online. This is an informational message only. No user action is required. ForceFlush is enabled for this instance.

2024-11-17 19:40:11.66 Server Total Log Writer threads: 2. This is an informational message; no user action is required.

2024-11-17 19:40:11.67 Server Software Usage Metrics is disabled.

2024-11-17 19:40:11.72 spid18s [1]. Feature Status: PVS: 0. CTR: 0. ConcurrentPFSUpdate: 1. ConcurrentGAMUpdate: 1. ConcurrentSGAMUpdate: 1, CleanupUnderUserTransaction: 0. TranLevelPVS: 0

2024-11-17 19:40:11.75 spid18s Starting up database 'master'. ForceFlush feature is enabled for log durability.

2024-11-17 19:40:12.31 spid18s Converting database 'master' from version 927 to the current version 957.

2024-11-17 19:40:12.31 spid18s Database 'master' running the upgrade step from version 927 to version 928.

2024-11-17 19:40:12.58 spid18s Database 'master' running the upgrade step from version 928 to version 929.

2024-11-17 19:40:12.63 spid18s Database 'master' running the upgrade step from version 929 to version 930.

2024-11-17 19:40:12.65 spid18s Database 'master' running the upgrade step from version 930 to version 931.

2024-11-17 19:40:12.70 spid18s Database 'master' running the upgrade step from version 931 to version 932.

2024-11-17 19:40:12.71 spid18s Database 'master' running the upgrade step from version 932 to version 933.

2024-11-17 19:40:12.72 spid18s Database 'master' running the upgrade step from version 933 to version 934.

2024-11-17 19:40:12.73 spid18s Database 'master' running the upgrade step from version 934 to version 935.

2024-11-17 19:40:12.78 spid18s Database 'master' running the upgrade step from version 935 to version 936.

2024-11-17 19:40:12.79 spid18s Database 'master' running the upgrade step from version 936 to version 937.

2024-11-17 19:40:12.80 spid18s Database 'master' running the upgrade step from version 937 to version 938.

2024-11-17 19:40:12.81 spid18s Database 'master' running the upgrade step from version 938 to version 939.

2024-11-17 19:40:12.82 Server External governance manager initialized

2024-11-17 19:40:12.82 spid18s Database 'master' running the upgrade step from version 939 to version 940.

2024-11-17 19:40:12.83 spid18s Database 'master' running the upgrade step from version 940 to version 941.

2024-11-17 19:40:12.85 spid18s Database 'master' running the upgrade step from version 941 to version 942.

2024-11-17 19:40:12.85 spid18s Database 'master' running the upgrade step from version 942 to version 943.

2024-11-17 19:40:12.90 spid18s Database 'master' running the upgrade step from version 943 to version 944.

2024-11-17 19:40:12.90 spid18s Database 'master' running the upgrade step from version 944 to version 945.

2024-11-17 19:40:12.93 spid18s Database 'master' running the upgrade step from version 945 to version 946.

2024-11-17 19:40:12.95 spid18s Database 'master' running the upgrade step from version 946 to version 947.

2024-11-17 19:40:12.96 spid18s Database 'master' running the upgrade step from version 947 to version 948.

2024-11-17 19:40:12.96 spid18s Database 'master' running the upgrade step from version 948 to version 949.

2024-11-17 19:40:12.98 spid18s Database 'master' running the upgrade step from version 949 to version 950.

2024-11-17 19:40:12.99 spid18s Database 'master' running the upgrade step from version 950 to version 951.

2024-11-17 19:40:13.02 spid18s Database 'master' running the upgrade step from version 951 to version 952.

2024-11-17 19:40:13.02 spid18s Database 'master' running the upgrade step from version 952 to version 953.

2024-11-17 19:40:13.04 spid18s Database 'master' running the upgrade step from version 953 to version 954.

2024-11-17 19:40:13.05 spid18s Database 'master' running the upgrade step from version 954 to version 955.

2024-11-17 19:40:13.06 spid18s Database 'master' running the upgrade step from version 955 to version 956.

2024-11-17 19:40:13.26 spid18s Database 'master' running the upgrade step from version 956 to version 957.

2024-11-17 19:40:13.44 spid19s [32767]. Feature Status: PVS: 0. CTR: 0. ConcurrentPFSUpdate: 1. ConcurrentGAMUpdate: 1. ConcurrentSGAMUpdate: 1, CleanupUnderUserTransaction: 0. TranLevelPVS: 0

2024-11-17 19:40:13.45 spid19s Starting up database 'mssqlsystemresource'.

2024-11-17 19:40:13.46 spid19s The resource database build version is 16.00.4165. This is an informational message only. No user action is required.

2024-11-17 19:40:13.49 spid29s Password policy update was successful.

2024-11-17 19:40:13.52 spid19s [3]. Feature Status: PVS: 0. CTR: 0. ConcurrentPFSUpdate: 1. ConcurrentGAMUpdate: 1. ConcurrentSGAMUpdate: 1, CleanupUnderUserTransaction: 0. TranLevelPVS: 0

2024-11-17 19:40:13.53 spid19s Starting up database 'model'.

2024-11-17 19:40:13.56 spid18s [32762]. Feature Status: PVS: 0. CTR: 0. ConcurrentPFSUpdate: 1. ConcurrentGAMUpdate: 1. ConcurrentSGAMUpdate: 1, CleanupUnderUserTransaction: 0. TranLevelPVS: 0

2024-11-17 19:40:13.57 spid18s Starting up database 'model_replicate_dmaster'.

2024-11-17 19:40:13.64 spid19s Converting database 'model' from version 927 to the current version 957.

2024-11-17 19:40:13.65 spid19s Database 'model' running the upgrade step from version 927 to version 928.

2024-11-17 19:40:13.67 spid19s Database 'model' running the upgrade step from version 928 to version 929.

2024-11-17 19:40:13.78 spid18s Converting database 'model_replicated_master' from version 927 to the current version 957.

2024-11-17 19:40:13.78 spid19s Database 'model' running the upgrade step from version 929 to version 930.

2024-11-17 19:40:13.79 spid18s Database 'model_replicatedmaster' running the upgrade step from version 927 to version 928.

2024-11-17 19:40:13.81 spid19s Database 'model' running the upgrade step from version 930 to version 931.

2024-11-17 19:40:13.81 spid18s Database 'model_replicatedmaster' running the upgrade step from version 928 to version 929.

2024-11-17 19:40:13.82 spid19s Database 'model' running the upgrade step from version 931 to version 932.

2024-11-17 19:40:13.83 spid19s Database 'model' running the upgrade step from version 932 to version 933.

2024-11-17 19:40:13.84 spid19s Database 'model' running the upgrade step from version 933 to version 934.

2024-11-17 19:40:13.86 spid19s Database 'model' running the upgrade step from version 934 to version 935.

2024-11-17 19:40:13.87 spid19s Database 'model' running the upgrade step from version 935 to version 936.

2024-11-17 19:40:13.90 spid19s Database 'model' running the upgrade step from version 936 to version 937.

2024-11-17 19:40:13.91 spid19s Database 'model' running the upgrade step from version 937 to version 938.

2024-11-17 19:40:13.92 spid18s Database 'model_replicatedmaster' running the upgrade step from version 929 to version 930.

2024-11-17 19:40:13.93 spid19s Database 'model' running the upgrade step from version 938 to version 939.

2024-11-17 19:40:13.94 Server Common language runtime (CLR) functionality initialized.

2024-11-17 19:40:13.95 spid19s Database 'model' running the upgrade step from version 939 to version 940.

2024-11-17 19:40:13.97 spid19s Database 'model' running the upgrade step from version 940 to version 941.

2024-11-17 19:40:13.97 spid18s Database 'model_replicatedmaster' running the upgrade step from version 930 to version 931.

2024-11-17 19:40:13.99 spid19s Database 'model' running the upgrade step from version 941 to version 942.

2024-11-17 19:40:13.99 spid18s Database 'model_replicatedmaster' running the upgrade step from version 931 to version 932.

2024-11-17 19:40:14.00 spid19s Database 'model' running the upgrade step from version 942 to version 943.

2024-11-17 19:40:14.03 spid18s Database 'model_replicatedmaster' running the upgrade step from version 932 to version 933.

2024-11-17 19:40:14.05 spid18s Database 'model_replicatedmaster' running the upgrade step from version 933 to version 934.

2024-11-17 19:40:14.05 spid19s Database 'model' running the upgrade step from version 943 to version 944.

2024-11-17 19:40:14.06 spid18s Database 'model_replicatedmaster' running the upgrade step from version 934 to version 935.

2024-11-17 19:40:14.06 spid19s Database 'model' running the upgrade step from version 944 to version 945.

2024-11-17 19:40:14.09 spid18s Database 'model_replicatedmaster' running the upgrade step from version 935 to version 936.

2024-11-17 19:40:14.09 spid19s Database 'model' running the upgrade step from version 945 to version 946.

2024-11-17 19:40:14.10 spid18s Database 'model_replicatedmaster' running the upgrade step from version 936 to version 937.

2024-11-17 19:40:14.11 spid19s Database 'model' running the upgrade

step from version 946 to version 947.

2024-11-17 19:40:14.12 spid18s Database 'model_replicatedmaster' running the upgrade step from version 937 to version 938.

2024-11-17 19:40:14.13 spid19s Database 'model' running the upgrade step from version 947 to version 948.

2024-11-17 19:40:14.14 spid18s Database 'model_replicatedmaster' running the upgrade step from version 938 to version 939.

2024-11-17 19:40:14.14 spid19s Database 'model' running the upgrade step from version 948 to version 949.

2024-11-17 19:40:14.16 spid18s Database 'model_replicatedmaster' running the upgrade step from version 939 to version 940.

2024-11-17 19:40:14.18 spid19s Database 'model' running the upgrade step from version 949 to version 950.

2024-11-17 19:40:14.19 spid18s Database 'model_replicatedmaster' running the upgrade step from version 940 to version 941.

2024-11-17 19:40:14.19 spid19s Database 'model' running the upgrade step from version 950 to version 951.

2024-11-17 19:40:14.21 spid18s Database 'model_replicatedmaster' running the upgrade step from version 941 to version 942.

2024-11-17 19:40:14.21 spid19s Database 'model' running the upgrade step from version 951 to version 952.

2024-11-17 19:40:14.23 spid19s Database 'model' running the upgrade step from version 952 to version 953.

2024-11-17 19:40:14.23 spid18s Database 'model_replicatedmaster' running the upgrade step from version 942 to version 943.

2024-11-17 19:40:14.24 spid19s Database 'model' running the upgrade step from version 953 to version 954.

2024-11-17 19:40:14.26 spid19s Database 'model' running the upgrade step from version 954 to version 955.

2024-11-17 19:40:14.27 spid19s Database 'model' running the upgrade step from version 955 to version 956.

2024-11-17 19:40:14.28 spid18s Database 'model_replicatedmaster' running the upgrade step from version 943 to version 944.

2024-11-17 19:40:14.29 spid18s Database 'model_replicatedmaster' running the upgrade step from version 944 to version 945.

2024-11-17 19:40:14.29 spid29s A self-generated certificate was successfully loaded for encryption.

2024-11-17 19:40:14.30 spid29s Server is listening on ['any' <ipv6> 1433] accept sockets 1.

2024-11-17 19:40:14.31 spid19s Database 'model' running the upgrade step from version 956 to version 957.

2024-11-17 19:40:14.31 spid29s Server is listening on ['any' <ipv4> 1433] accept sockets 1.

2024-11-17 19:40:14.31 spid18s Database 'model_replicatedmaster' running the upgrade step from version 945 to version 946.

2024-11-17 19:40:14.32 Server Server is listening on [::1 <ipv6> 1434] accept sockets 1.

2024-11-17 19:40:14.32 Server Server is listening on [127.0.0.1 <ipv4> 1434] accept sockets 1.

2024-11-17 19:40:14.33 Server Dedicated admin connection support was established for listening locally on port 1434.

2024-11-17 19:40:14.33 spid18s Database 'model_replicatedmaster' running the upgrade step from version 946 to version 947.

2024-11-17 19:40:14.34 spid29s Server is listening on [::1 <ipv6> 1431] accept sockets 1.

2024-11-17 19:40:14.35 spid18s Database 'model_replicatedmaster' running the upgrade step from version 947 to version 948.

2024-11-17 19:40:14.35 spid29s Server is listening on [127.0.0.1 <ipv4> 1431] accept sockets 1.
2024-11-17 19:40:14.36 spid29s SQL Server is now ready for client connections. This is an informational message; no user action is required.
2024-11-17 19:40:14.38 spid18s Database 'model_replicatedmaster' running the upgrade step from version 948 to version 949.
2024-11-17 19:40:14.39 spid18s Database 'model_replicatedmaster' running the upgrade step from version 949 to version 950.
2024-11-17 19:40:14.39 spid18s Database 'model_replicatedmaster' running the upgrade step from version 950 to version 951.
2024-11-17 19:40:14.41 spid18s Database 'model_replicatedmaster' running the upgrade step from version 951 to version 952.
2024-11-17 19:40:14.42 spid18s Database 'model_replicatedmaster' running the upgrade step from version 952 to version 953.
2024-11-17 19:40:14.50 spid18s Database 'model_replicatedmaster' running the upgrade step from version 953 to version 954.
2024-11-17 19:40:14.52 spid18s Database 'model_replicatedmaster' running the upgrade step from version 954 to version 955.
2024-11-17 19:40:14.53 spid18s Database 'model_replicatedmaster' running the upgrade step from version 955 to version 956.
2024-11-17 19:40:14.56 spid18s Database 'model_replicatedmaster' running the upgrade step from version 956 to version 957.
2024-11-17 19:40:14.64 spid18s [32761]. Feature Status: PVS: 0. CTR: 0. ConcurrentPFSUpdate: 1. ConcurrentGAMUpdate: 1. ConcurrentSGAMUpdate: 1, CleanupUnderUserTransaction: 0. TranLevelPVS: 0
2024-11-17 19:40:14.65 spid18s Starting up database 'model_msdb'.
2024-11-17 19:40:14.73 spid18s Converting database 'model_msdb' from version 927 to the current version 957.
2024-11-17 19:40:14.73 spid18s Database 'model_msdb' running the upgrade step from version 927 to version 928.
2024-11-17 19:40:14.74 spid18s Database 'model_msdb' running the upgrade step from version 928 to version 929.
2024-11-17 19:40:14.78 spid18s Database 'model_msdb' running the upgrade step from version 929 to version 930.
2024-11-17 19:40:14.80 spid18s Database 'model_msdb' running the upgrade step from version 930 to version 931.
2024-11-17 19:40:14.81 spid18s Database 'model_msdb' running the upgrade step from version 931 to version 932.
2024-11-17 19:40:14.82 spid18s Database 'model_msdb' running the upgrade step from version 932 to version 933.
2024-11-17 19:40:14.83 spid18s Database 'model_msdb' running the upgrade step from version 933 to version 934.
2024-11-17 19:40:14.84 spid18s Database 'model_msdb' running the upgrade step from version 934 to version 935.
2024-11-17 19:40:14.84 spid18s Database 'model_msdb' running the upgrade step from version 935 to version 936.
2024-11-17 19:40:14.85 spid18s Database 'model_msdb' running the upgrade step from version 936 to version 937.
2024-11-17 19:40:14.85 spid18s Database 'model_msdb' running the upgrade step from version 937 to version 938.
2024-11-17 19:40:14.98 spid18s Database 'model_msdb' running the upgrade step from version 938 to version 939.
2024-11-17 19:40:14.99 spid18s Database 'model_msdb' running the upgrade step from version 939 to version 940.
2024-11-17 19:40:14.99 spid18s Database 'model_msdb' running the upgrade step from version 940 to version 941.
2024-11-17 19:40:15.00 spid18s Database 'model_msdb' running the upgrade step from version 941 to version 942.

rade step from version 941 to version 942.

2024-11-17 19:40:15.02 spid18s Database 'model_msdb' running the upgrade step from version 942 to version 943.

2024-11-17 19:40:15.05 spid18s Database 'model_msdb' running the upgrade step from version 943 to version 944.

2024-11-17 19:40:15.06 spid18s Database 'model_msdb' running the upgrade step from version 944 to version 945.

2024-11-17 19:40:15.08 spid18s Database 'model_msdb' running the upgrade step from version 945 to version 946.

2024-11-17 19:40:15.09 spid18s Database 'model_msdb' running the upgrade step from version 946 to version 947.

2024-11-17 19:40:15.09 spid18s Database 'model_msdb' running the upgrade step from version 947 to version 948.

2024-11-17 19:40:15.10 spid18s Database 'model_msdb' running the upgrade step from version 948 to version 949.

2024-11-17 19:40:15.11 spid18s Database 'model_msdb' running the upgrade step from version 949 to version 950.

2024-11-17 19:40:15.12 spid18s Database 'model_msdb' running the upgrade step from version 950 to version 951.

2024-11-17 19:40:15.14 spid18s Database 'model_msdb' running the upgrade step from version 951 to version 952.

2024-11-17 19:40:15.14 spid18s Database 'model_msdb' running the upgrade step from version 952 to version 953.

2024-11-17 19:40:15.15 spid18s Database 'model_msdb' running the upgrade step from version 953 to version 954.

2024-11-17 19:40:15.16 spid18s Database 'model_msdb' running the upgrade step from version 954 to version 955.

2024-11-17 19:40:15.16 spid18s Database 'model_msdb' running the upgrade step from version 955 to version 956.

2024-11-17 19:40:15.19 spid18s Database 'model_msdb' running the upgrade step from version 956 to version 957.

2024-11-17 19:40:15.23 spid36s Attribute synchronization initialized

2024-11-17 19:40:15.23 spid36s Attribute synchronization manager initialized

2024-11-17 19:40:15.23 spid18s Resource governor reconfiguration succeeded.

2024-11-17 19:40:15.24 spid18s SQL Server Audit is starting the audits. This is an informational message. No user action is required.

2024-11-17 19:40:15.25 spid18s SQL Server Audit has started the audits. This is an informational message. No user action is required.

2024-11-17 19:40:15.37 spid18s SQL Trace ID 1 was started by login "sa".

2024-11-17 19:40:15.38 spid18s Server name is '9b6b36b99b48'. This is an informational message only. No user action is required.

2024-11-17 19:40:15.42 spid19s Clearing tempdb database.

2024-11-17 19:40:15.42 spid51s Always On: The availability replica manager is starting. This is an informational message only. No user action is required.

2024-11-17 19:40:15.43 spid52s [4]. Feature Status: PVS: 0. CTR: 0. ConcurrentPFSUpdate: 1. ConcurrentGAMUpdate: 1. ConcurrentSGAMUpdate: 1, CleanupUnderUserTransaction: 0. TranLevelPVS: 0

2024-11-17 19:40:15.45 spid52s Starting up database 'msdb'.

2024-11-17 19:40:15.46 spid51s Always On: The availability replica manager is waiting for the instance of SQL Server to allow client connections. This is an informational message only. No user action is required.

2024-11-17 19:40:15.47 spid27s ERROR: Unable to set system administrator password: Password validation failed. The password does not meet SQ

L Server password policy requirements because it is too short. The password must be at least 8 characters..

2024-11-17 19:40:15.49 spid27s An error occurred during server setup. See previous errors for more information.

2024-11-17 19:40:15.50 spid27s SQL Trace was stopped due to server shutdown. Trace ID = '1'. This is an informational message only; no user action is required.

sqlserver-1 |

sqlserver-1 exited with code 255
eshoppublicapi-1 exited with code 0
eshopwebmvc-1 exited with code 137

Dies startet das Projekt im Docker-Container, und du kannst die Anwendung unter `localhost:5106` für das Frontend und unter `localhost:5200` im Browser erreichen.

5. Projekt auf Azure ausrollen

Für das Deployment auf Azure benötigst du ein Azure-Konto und die **Azure CLI**.

Falls du sie noch nicht installiert hast, folge diesen Anweisungen:

[Installiere Azure CLI.](#)

5.1. Voraussetzungen

- Ein Azure-Konto: [Azure Kostenloses Konto](#)
- Azure CLI installiert: [Azure CLI Installieren](#)
- Docker (wenn du Docker für das Deployment verwenden möchtest).
- Ein App Service Plan auf Azure (für das Hosting der Webanwendung).
- (Optional) Eine registrierte Anwendung in Azure Active Directory, falls du Authentifizierung benötigst.

5.2. Projekt für das Azure-Deployment vorbereiten

Bevor du das Projekt auf Azure ausrollst, solltest du sicherstellen, dass alle Umgebungsvariablen wie Datenbank-Verbindungszeichenfolgen und geheime Werte korrekt gesetzt sind. Du solltest diese nicht im Code speichern, sondern über **App Settings** in Azure verwalten.

5.3. Azure Deployment mit Azure CLI

Nun kannst du das Projekt mit Azure CLI auf Azure bereitstellen. Folge diesen Schritten (diese sind aktuell ungetestet):

5.3.1. Anmelden bei Azure

Melde dich bei deinem Azure-Konto an:

```
az login
```

5.3.2. Erstellen einer Ressourcengruppe

Erstelle eine neue Ressourcengruppe, die alle Ressourcen für das Projekt enthält:

```
az group create --name eShopOnWebResourceGroup --location "East US"
```

5.3.3. Erstellen des App Service Plans

Der App Service Plan definiert die Infrastruktur, auf der deine Web-App laufen wird. Erstelle ihn mit:

```
az appservice plan create --name eShopOnWebPlan --resource-group eShopOnWebResourceGroup --sku B1 --is-linux
```

5.3.4. Erstellen der Web-App

Erstelle eine neue Web-App, die deine Anwendung hosten wird. Hier kannst du die Laufzeit angeben (z. B. .NET 6):

```
az webapp create --resource-group eShopOnWebResourceGroup --plan eShopOnWebPlan --name eShopOnWebApp --runtime "DOTNET|6.0"
```

5.3.5. Docker-Deployment (Optional)

Wenn du Docker verwendest, kannst du das Docker-Image auf Azure Web App for Containers hochladen:

```
az webapp config container set --name eShopOnWebApp --resource-group eShopOnWebResourceGroup --docker-custom-image-name eshoponweb
```

5.3.6. Git-Deployment (Für lokale .NET-Anwendung)

Falls du kein Docker verwendest, kannst du das Projekt auch direkt mit Git auf Azure deployen:

```
az webapp deployment source config-local-git --name eShopOnWebApp --resource-group eShopOnWebResourceGroup
```

5.4. Überprüfung der Bereitstellung

Nach erfolgreichem Deployment kannst du die Anwendung über die folgende URL aufrufen:

<https://<dein-app-name>.azurewebsites.net>

6. Fehlerbehebung und mögliche Probleme

- **Bauen des Projekts:** Wenn beim Bauen des Projekts Fehler auftreten, überprüfe, ob alle Abhängigkeiten korrekt installiert sind und ob du das richtige .NET SDK verwendest.
- **Docker:** Wenn Docker nicht funktioniert, überprüfe, ob Docker korrekt installiert und der Docker-Daemon läuft.
- **Azure Deployment:** Wenn das Azure Deployment fehlschlägt, stelle sicher, dass du alle Ressourcen korrekt erstellt hast und die Umgebungsvariablen für die Web-App richtig konfiguriert sind. Prüfe auch, ob du die richtige Runtime