

Zehn goldene Regeln

Um dein Softwareprojekt zuverlässig zu ruinieren



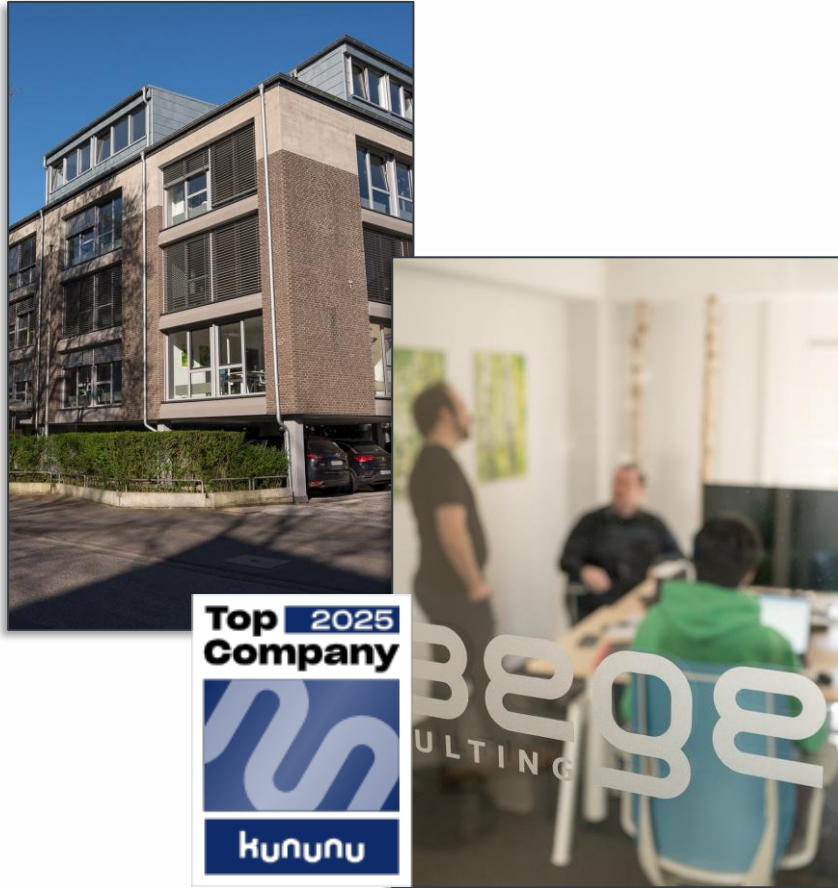
Andreas Monschau

Senior IT Consultant
Head of Traineeprogram

Haeger Consulting GmbH

- ✓ Andreas Monschau
- ✓ IT seit 2000, IT Consultant seit 2012
- ✓ Entwicklung, Architektur, Analyse, Teamleitung, Agilität...



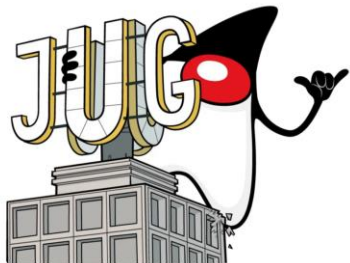
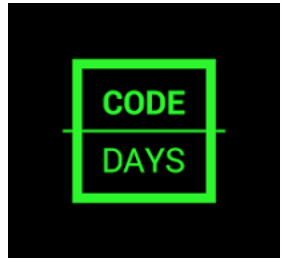


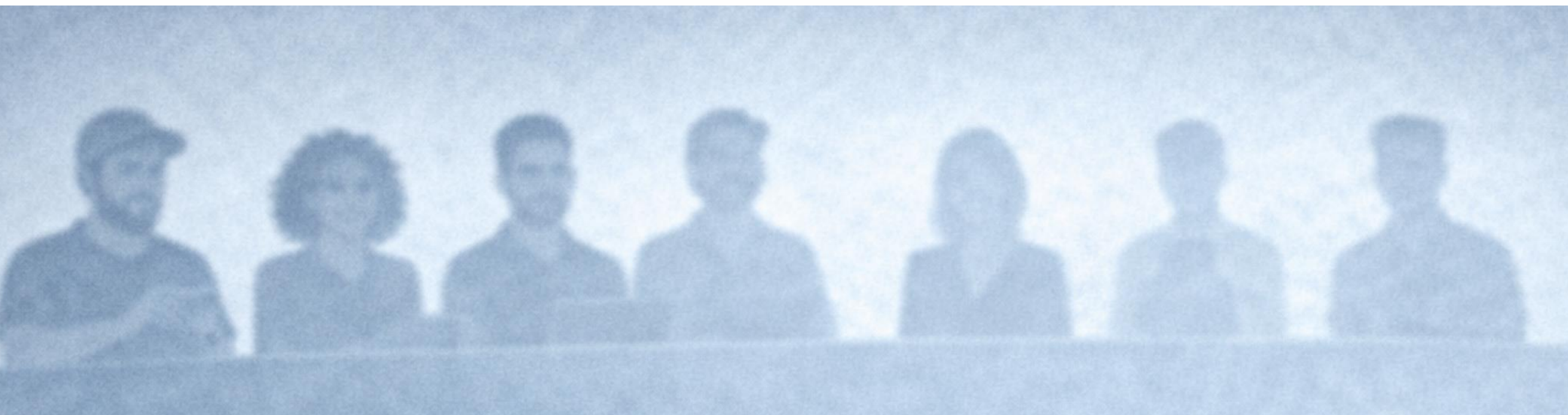
- ✓ Haeger Consulting GmbH
- ✓ Bonn, est. 2006, > 90 MA
- ✓ Menschen entwickeln
- ✓ Projekte in Spur bringen

DIE GOLDENEN REGELN

WIE VERMIESE ICH NEULINGEN
DEN EINSTIEG IN MEIN PROJEKT?







Developer

...

Tester

Agile Practioneer

Teamleiter

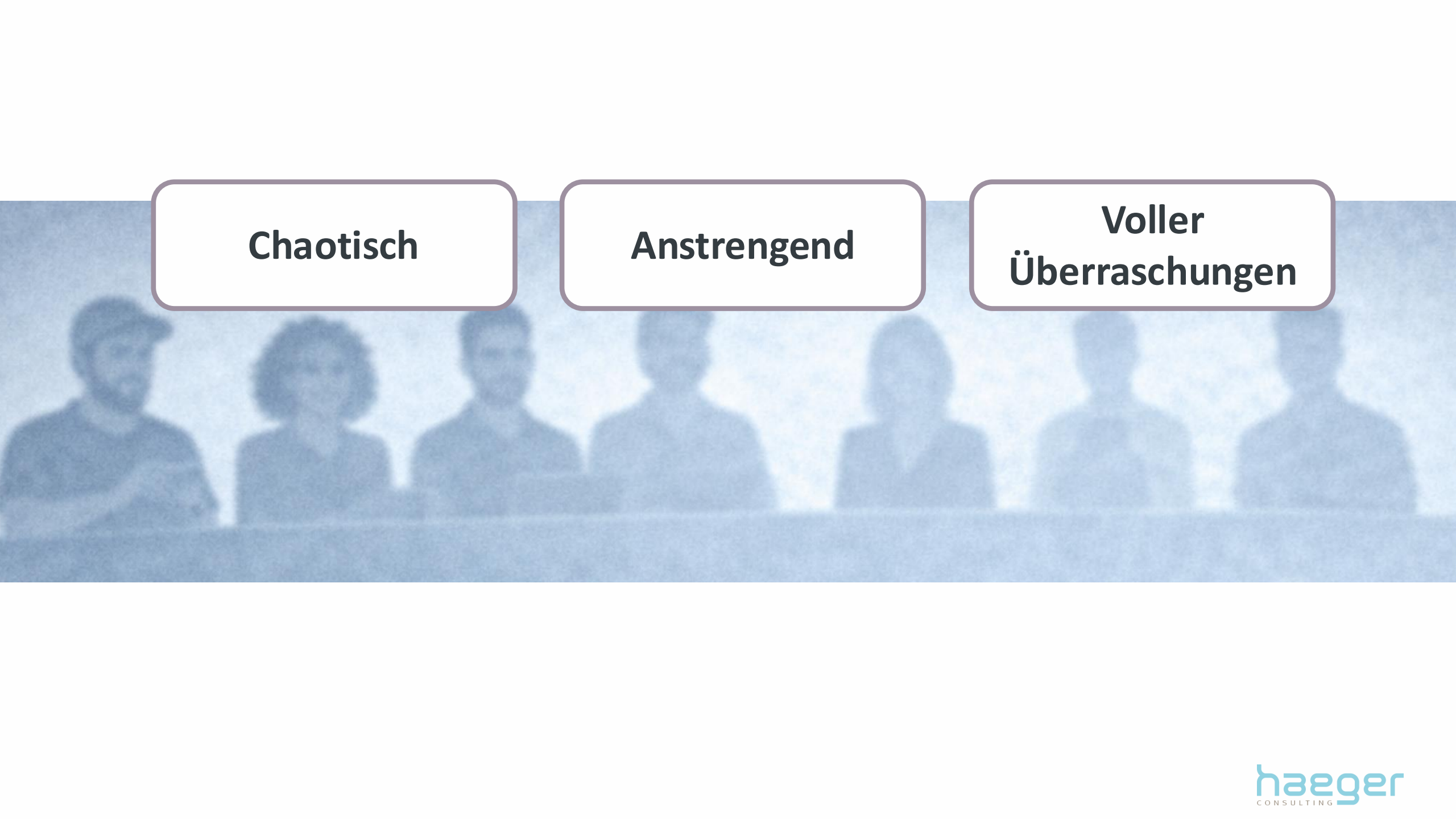
Projektmanager

DevOps

Chaotisch

Chaotisch

Anstrengend



Chaotisch

Anstrengend

**Voller
Überraschungen**

Chaotisch

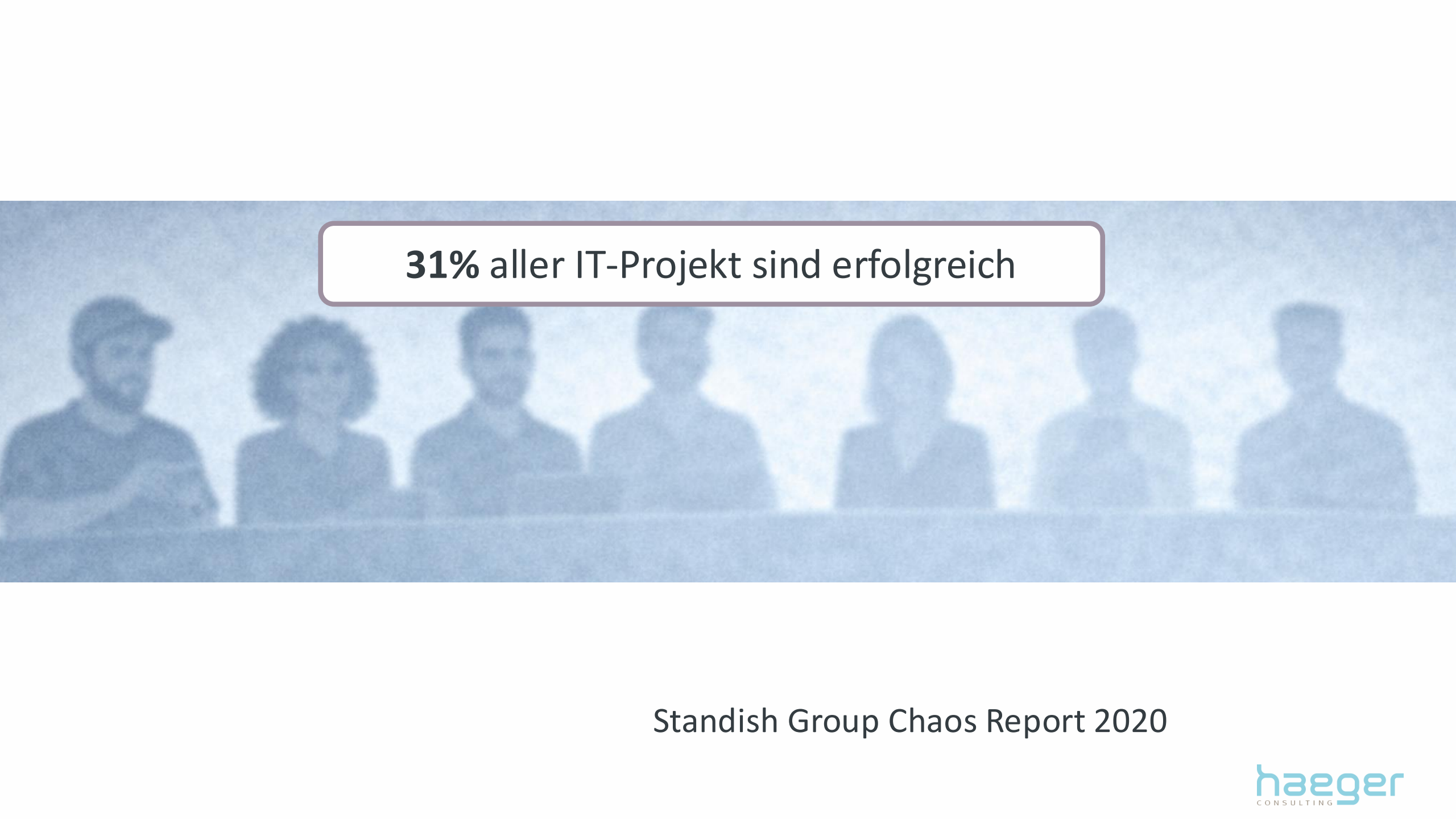
Anstrengend

**Voller
Überraschungen**

**So kann es gerne
bleiben**

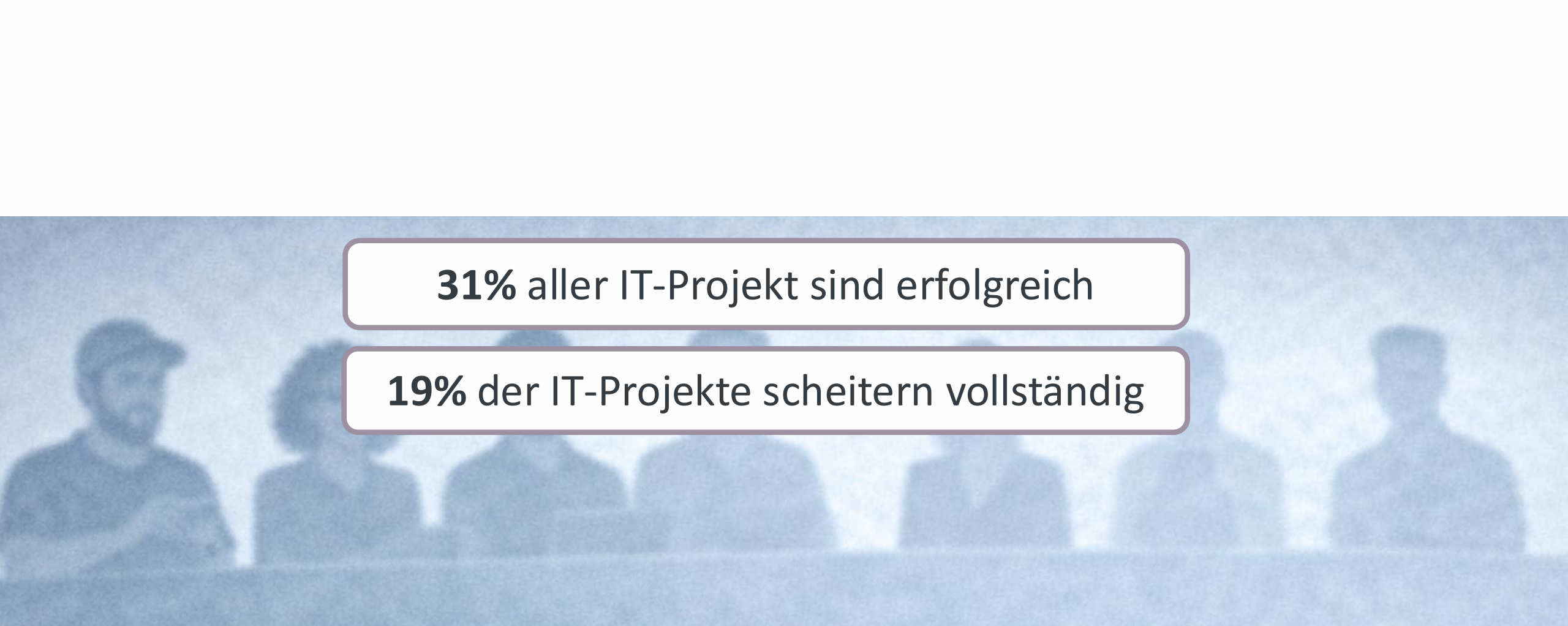


Standish Group Chaos Report 2020



31% aller IT-Projekt sind erfolgreich

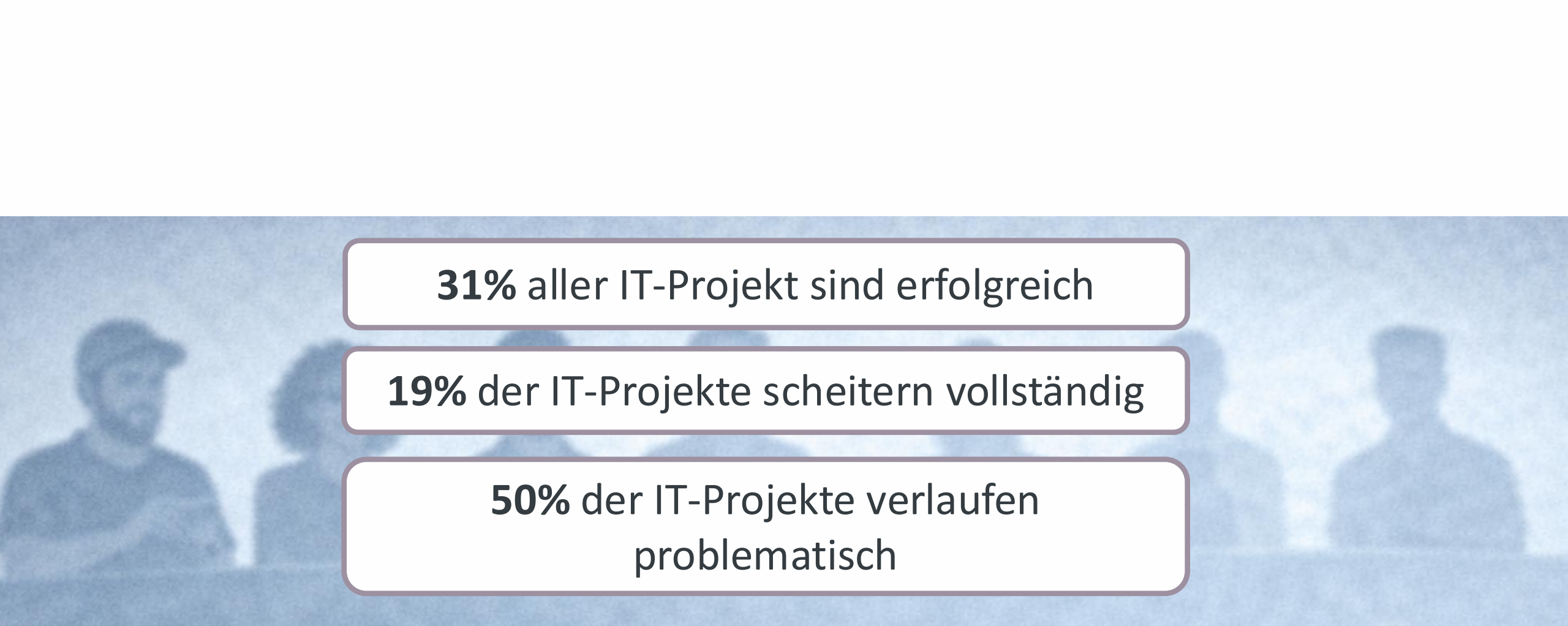
Standish Group Chaos Report 2020



31% aller IT-Projekt sind erfolgreich

19% der IT-Projekte scheitern vollständig

Standish Group Chaos Report 2020

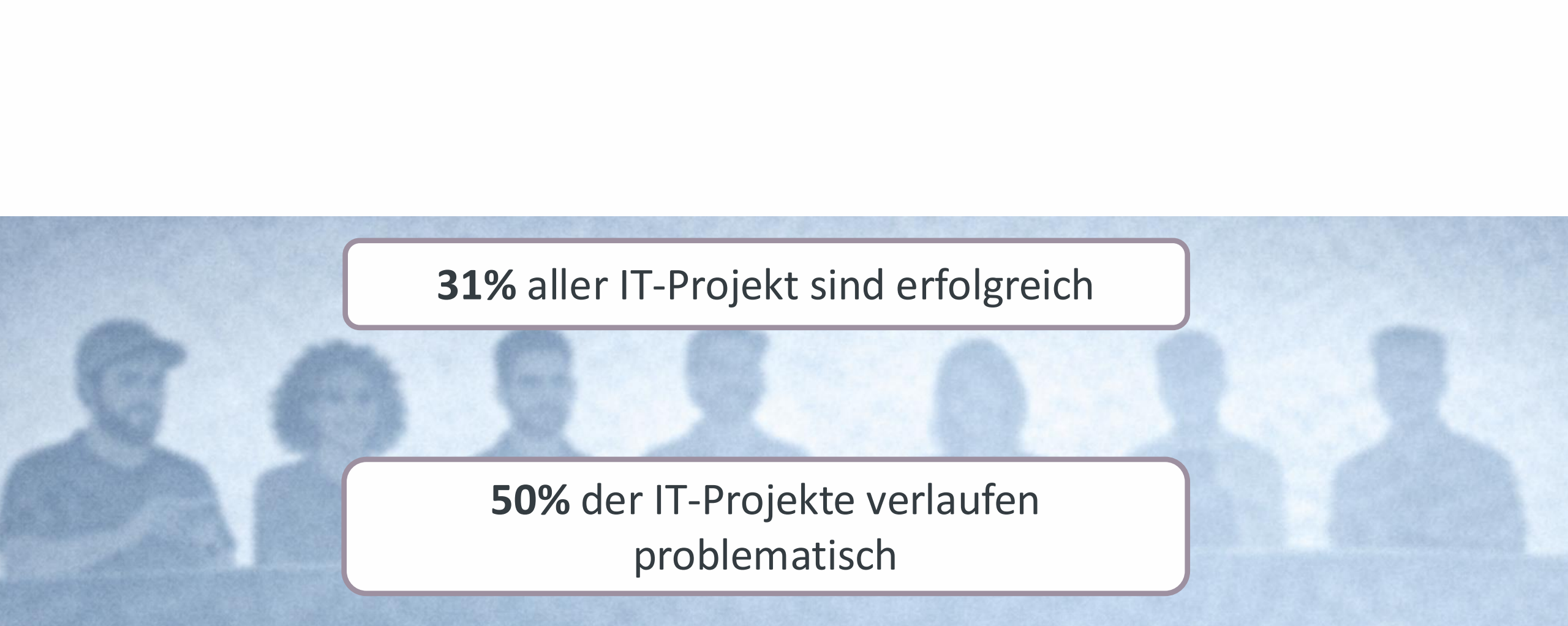


31% aller IT-Projekt sind erfolgreich

19% der IT-Projekte scheitern vollständig

50% der IT-Projekte verlaufen
problematisch

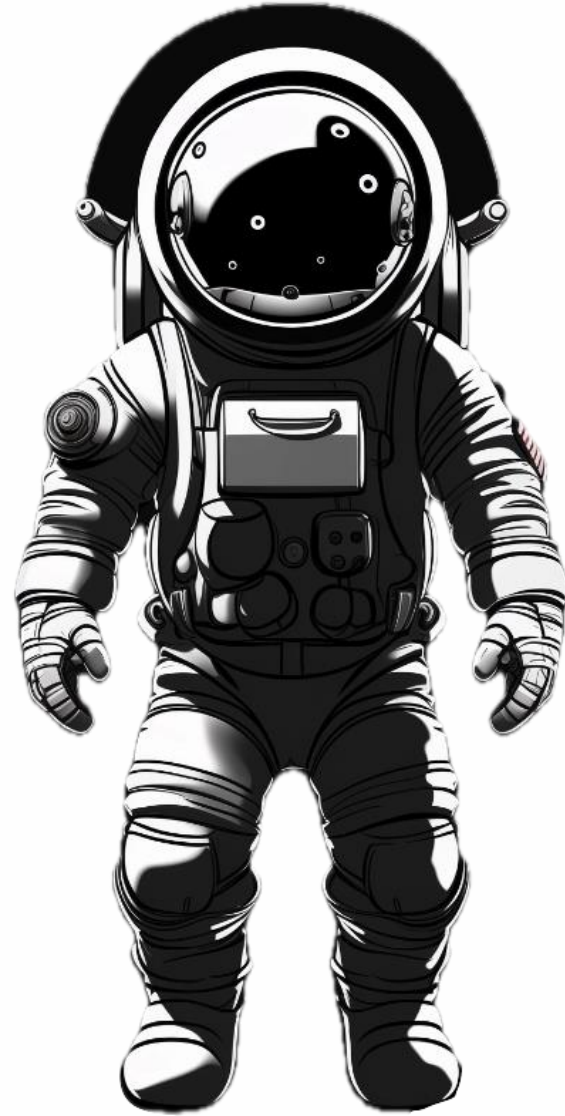
Standish Group Chaos Report 2020



31% aller IT-Projekt sind erfolgreich

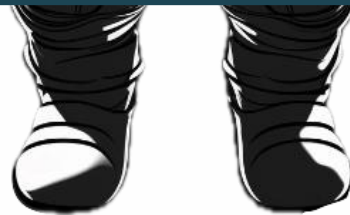
50% der IT-Projekte verlaufen
problematisch

Standish Group Chaos Report 2020



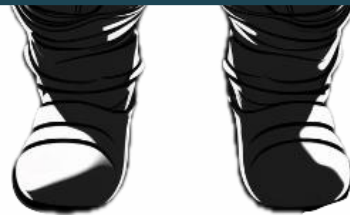


Zehn goldene Regeln





REGEL 1

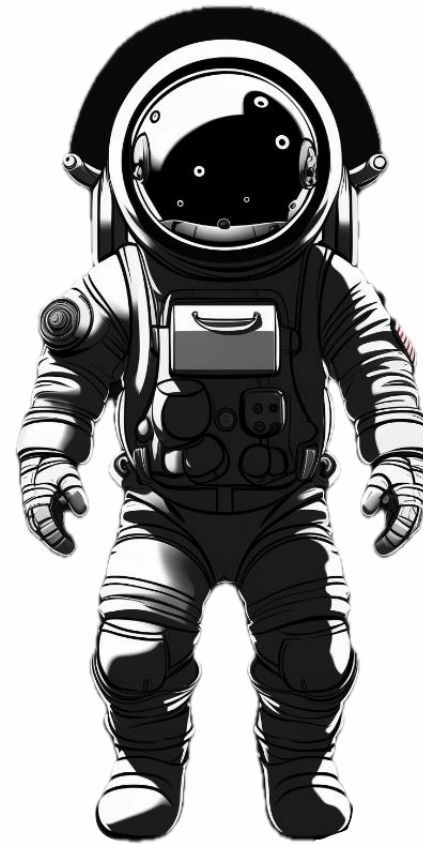


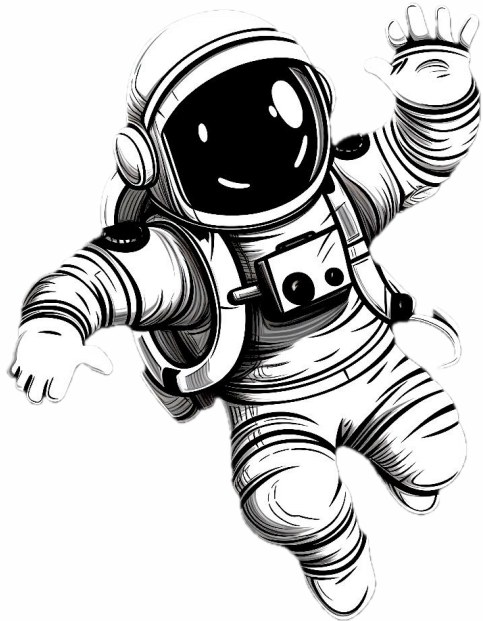


REGEL 1

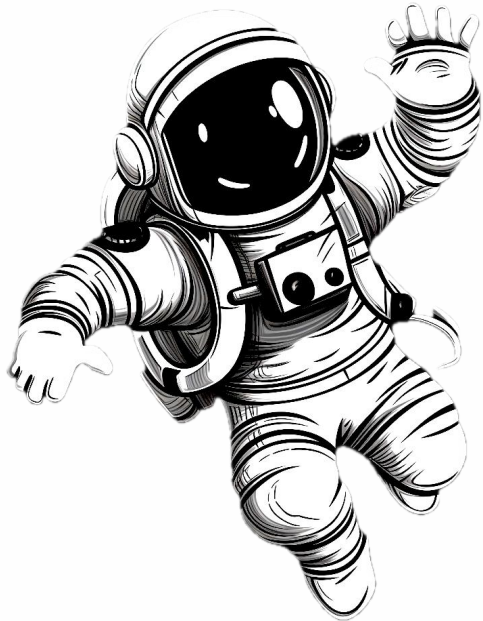


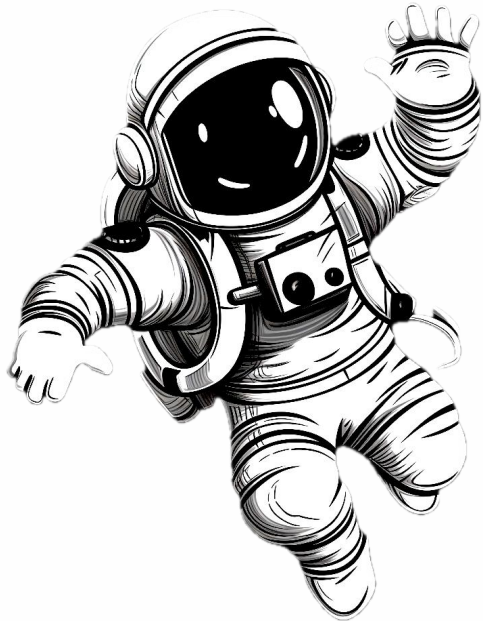
Für alle spannend!



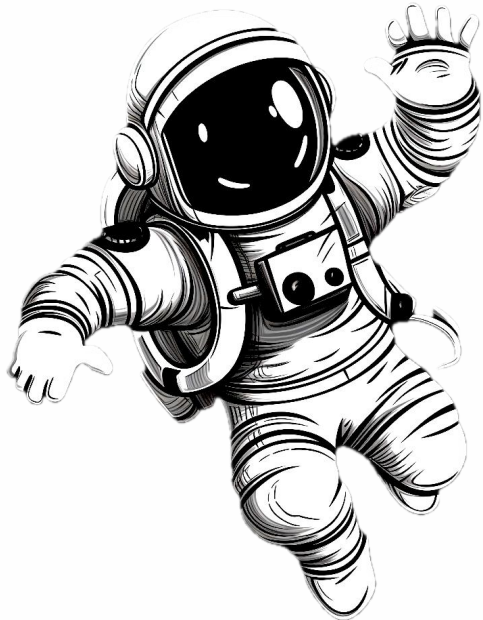


Projekteinstieg



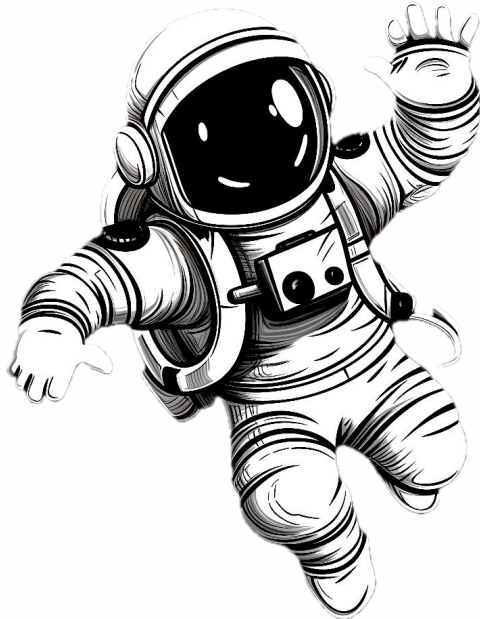


**Ein
unerfreuliches
Willkommen**



**Ein
unerfreuliches
Willkommen**

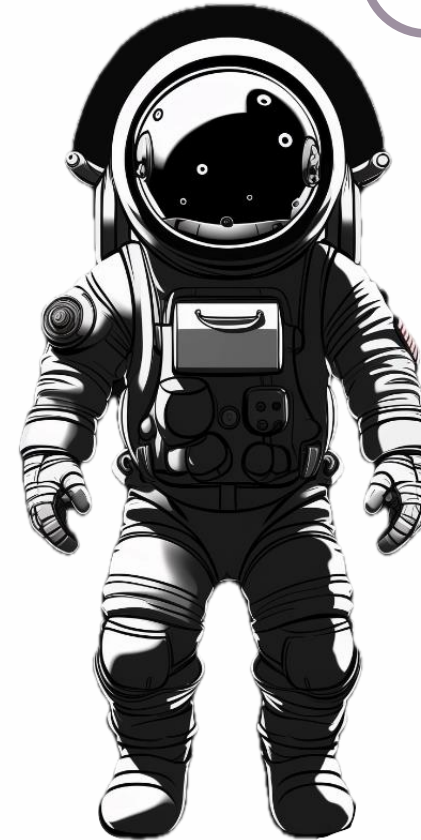
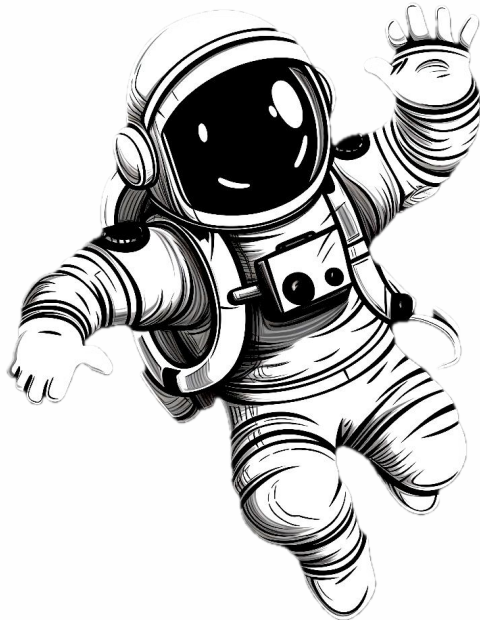
Unerreichbarkeit



**Ein
unerfreuliches
Willkommen**

Unerreichbarkeit

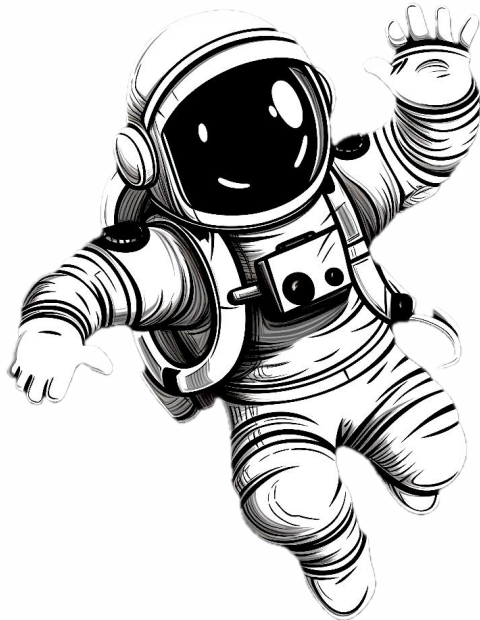
**Unsinnige,
widersprüchliche
oder gar keine
Aufgaben**



**Ein
unerfreuliches
Willkommen**

Unerreichbarkeit

**Unsinnige,
widersprüchliche
oder gar keine
Aufgaben**

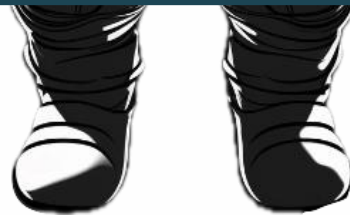


**Keine Fehler
verzeihen**



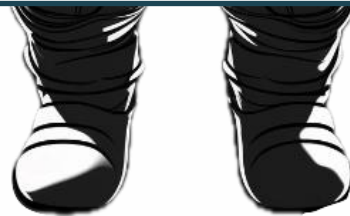


REGEL 1





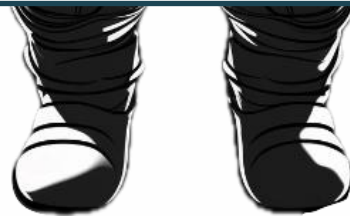
**Den Einstieg maximal
schlecht gestalten**





Den Einstieg maximal schlecht gestalten

Motivation wird
dadurch frühzeitig
verhindert





Den Einstieg maximal schlecht gestalten

Motivation wird
dadurch frühzeitig
verhindert

Das Projekt bekommt
keinen neuen Schwung



Den Einstieg maximal schlecht gestalten

**Motivation wird
dadurch frühzeitig
verhindert**

**Das Projekt bekommt
keinen neuen Schwung**

...



Regel 1



Regel 6



Regel 2



Regel 7



Regel 3



Regel 8



Regel 4



Regel 9



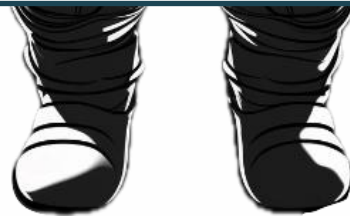
Regel 5



Regel 10



REGEL 2

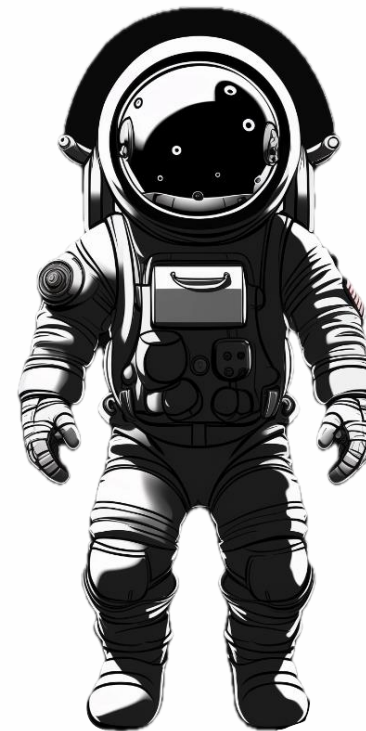
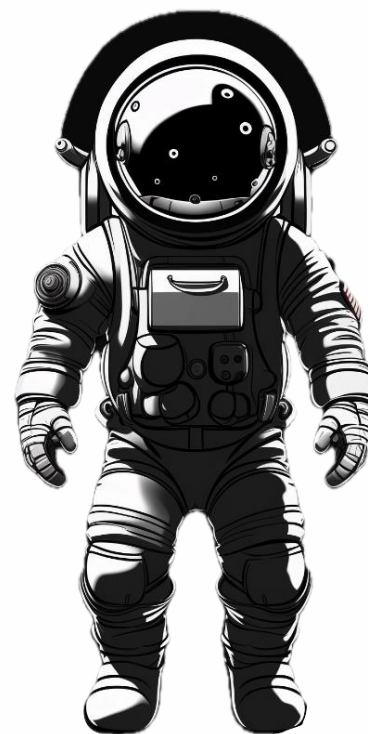
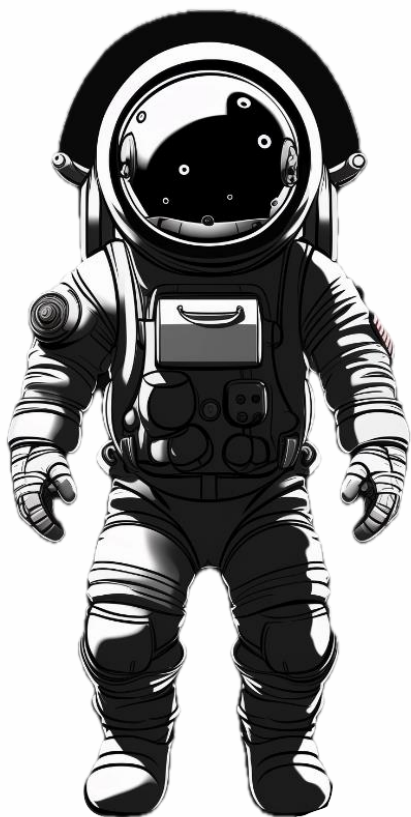




REGEL 2

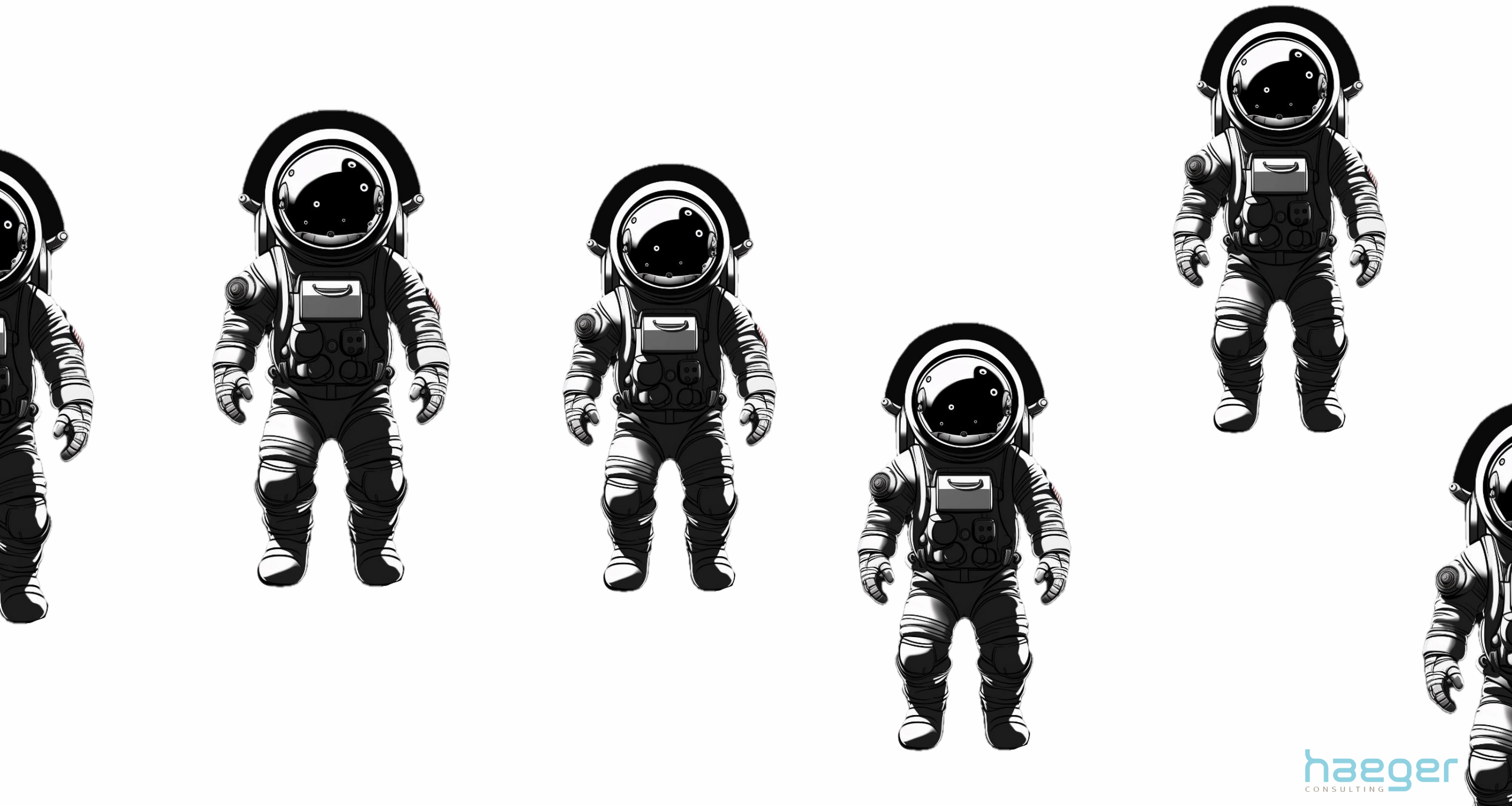


Für alle spannend, aber primär
für Devs





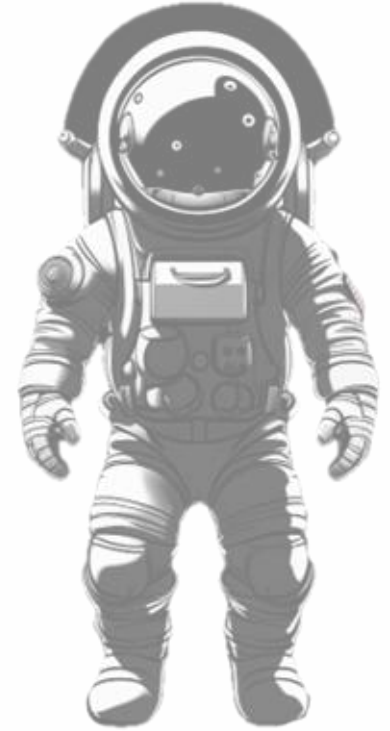
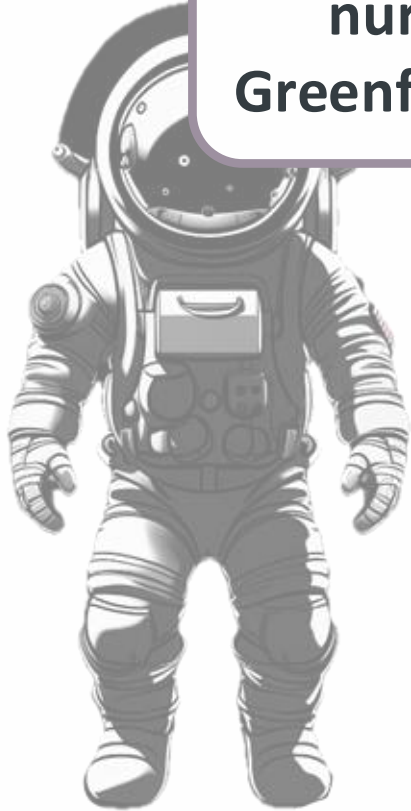
Hypothek auf die Zukunft





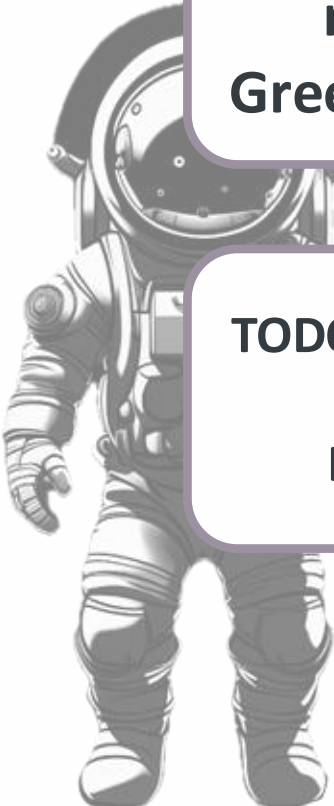
Grundsatzentscheidungen

**Sauberer Code ist
nur etwas für
Greenfield-Projekte!**

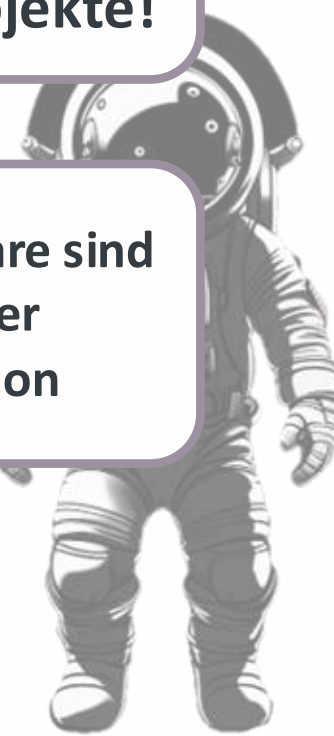


**Sauberer Code ist
nur etwas für
Greenfield-Projekte!**

**„Das fixen wird später“
ist eine valide
Architekturentscheidung**

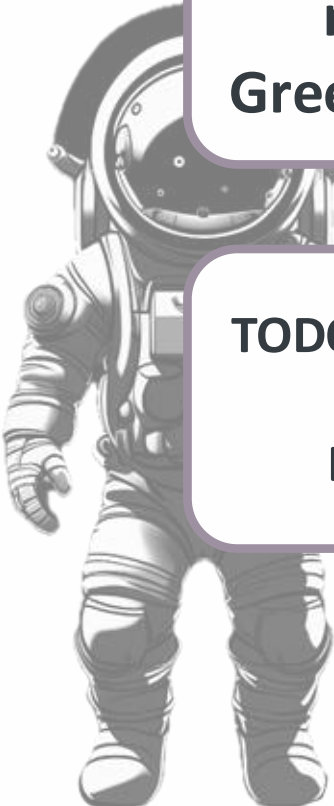


**Sauberer Code ist
nur etwas für
Greenfield-Projekte!**

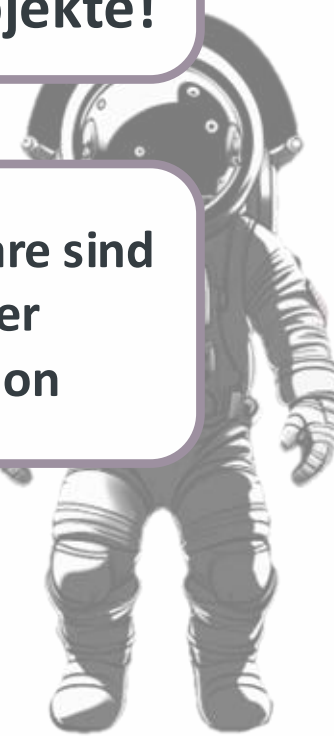


**TODO-Kommentare sind
eine Form der
Dokumentation**

**„Das fixen wir später“ ist
eine valide
Architekturentscheidung**



**Sauberer Code ist
nur etwas für
Greenfield-Projekte!**



**TODO-Kommentare sind
eine Form der
Dokumentation**

**„Das fixen wird später“
ist eine valide
Architekturentscheidung**



**Copy & Paste spart
Denkzeit**



**Sauberer Code ist
nur etwas für
Greenfield-Projekte!**

**„Das fixen wird später“
ist eine valide
Architekturentscheidung**

**TODO-Kommentare sind
eine Form der
Dokumentation**

**Copy & Paste spart
Denkzeit**

**Alte Workarounds
könnten noch mal
wichtig werden**

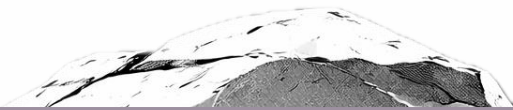


Führt zu...



Technische Schulden





Technische Schulden



Technische Schulden

```
public void processOrder(Order order) {  
    validate(order);  
    save(order);  
    notifyCustomer(order);  
}
```

Technische Schulden

```
public void processOrder(Order order) {  
    if(order == null) {  
        return;  
    }  
    if(!featureToggleService.isEnabled("NEW_ORDER_FLOW")) {  
        legacyProcess(order);  
        return;  
    }  
    if(order.getCustomer() != null &&  
        order.getCustomer().getType() == CustomerType.SPECIAL) {  
        applySpecialDiscount(order);  
    }  
    try {  
        validate(order);  
        enrichFromExternalSystem(order);  
        save(order);  
    } catch(Exception e) {  
        log.warn("Order processing failed", e);  
    }  
    if(order.getAmount().doubleValue() > 10000) {  
        notifySales(order);  
    }  
    notifyCustomer(order);  
}
```

Technische Schulden

```
public void processOrder(Order order) {  
    if (order == null) {  
        return;  
    }  
    if (!featureToggleService.isEnabled("NEW_ORDER_FLOW")) {  
        legacyProcess(order);  
        return;  
    }  
    if (order.getCustomer() != null &&  
        order.getCustomer().getType() == CustomerType.SPECIAL) {  
        applySpecialDiscount(order);  
    }  
    try {  
        validate(order);  
        enrichFromExternalSystem(order);  
        save(order);  
    } catch (Exception e) {  
        log.warn("Order processing failed", e);  
    }  
    if (order.getAmount().doubleValue() > 10000) {  
        notifySales(order);  
    }  
    notifyCustomer(order);  
}
```

Businesslogik

Fehlerbehandlung

Feature-Toggling

Technische Schulden



**Jedes Refactoring
wird zum Risiko**

```
public void processOrder(Order order) {  
    if (order.isNew()) {  
        registerNewCustomer(order);  
    }  
    if (order.isExisting()) {  
        validateExistingCustomer(order);  
    }  
    if (order.isNew() || order.isExisting()) {  
        order.applyDiscount();  
    }  
    try {  
        validate(order);  
        enrichFromExternalSystem(order);  
        save(order);  
    } catch (Exception e) {  
        log.warn("Order processing failed", e);  
    }  
    if (order.getAmount().doubleValue() > 10000) {  
        notifySales(order);  
    }  
    notifyCustomer(order);  
}
```

Technische Schulden



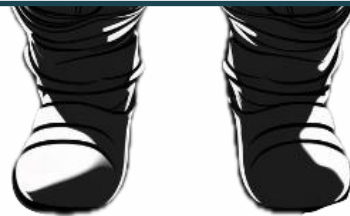
**Jedes Refactoring
wird zum Risiko**



**Nebenwirkungen
werden unkalkulierbar**



REGEL 2





Feiert technische Schulden



Feiert technische Schulden

**Joberhaltende
Maßnahme**



Feiert technische Schulden

**Joberhaltende
Maßnahme**

**Einarbeitung wird
nahezu unmöglich**



Feiert technische Schulden

**Joberhaltende
Maßnahme**

**Einarbeitung wird
nahezu unmöglich**

...



Regel 1



Regel 2



Regel 3



Regel 4



Regel 5



Regel 6



Regel 7



Regel 8



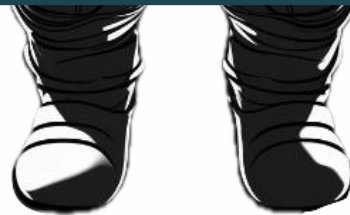
Regel 9



Regel 10



REGEL 3





REGEL 3



Für alle spannend



Dokumentation!





Grundsatzentscheidungen



**Dokumentation ist
über mehrere
Systeme verteilt**



**Dokumentation ist
über mehrere
Systeme verteilt**

**Dokumentation ist
veraltet**



**Dokumentation ist
über mehrere
Systeme verteilt**

**Dokumentation ist
veraltet**

**Das Wissen existiert
nur in den Köpfen**



**Dokumentation ist
über mehrere
Systeme verteilt**

**Dokumentation ist
veraltet**

**Das Wissen existiert
nur in den Köpfen**

**Dokumentation
planen, aber nicht
realisieren**





Führt zu...



Schlechte Dokumentation



Schlechte Dokumentation



Schlechte Dokumentation

Anforderungen



Erstellt von Andreas Monschau

Letzte Aktualisierung: vor Kurzem •  Analysen



TODO

+ Stichwort hinzufügen



Schlechte Dokumentation

```
/**
 *
 * todo
 *
 * @return
 */
@Override
public ArrayList<Client> importDataFromDB() {
    return datenbank.readClientList();
}
```



Schlechte Dokumentation

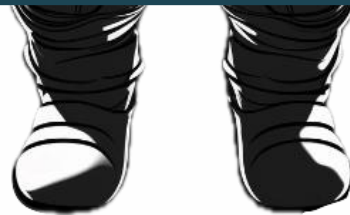


<https://arc42.org/images/arc42-site-logo.webp>





REGEL 3





**Dokumentiert
schlecht!**



Dokumentiert schlecht!

Vermasselt die
Einarbeitung



Dokumentiert schlecht!

**Vermasselt die
Einarbeitung**

**Schere zwischen Doku
und Realität wird
immer größer**



Dokumentiert schlecht!

Vermasselt die
Einarbeitung

Schere zwischen Doku
und Realität wird
immer größer

...



Regel 1



Regel 2



Regel 3



Regel 4



Regel 5



Regel 6



Regel 7



Regel 8



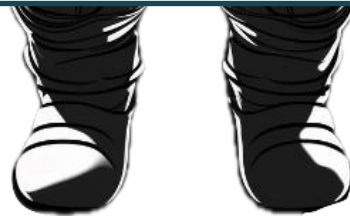
Regel 9



Regel 10



REGEL 4

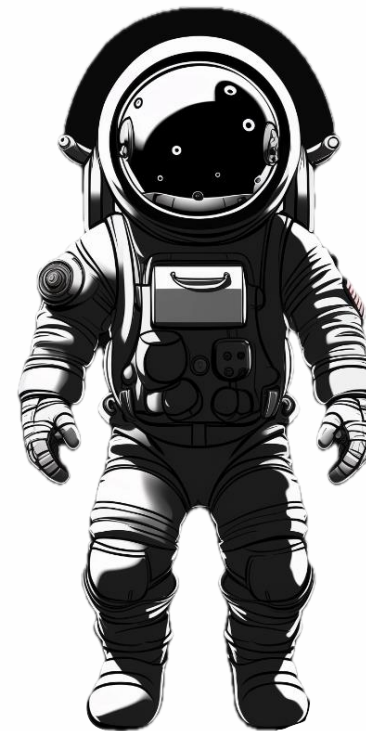
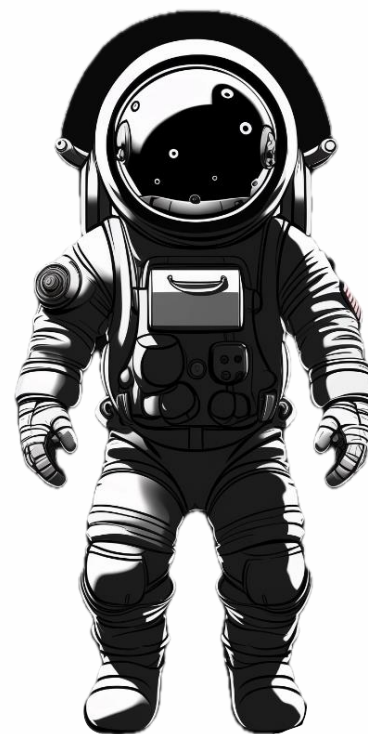
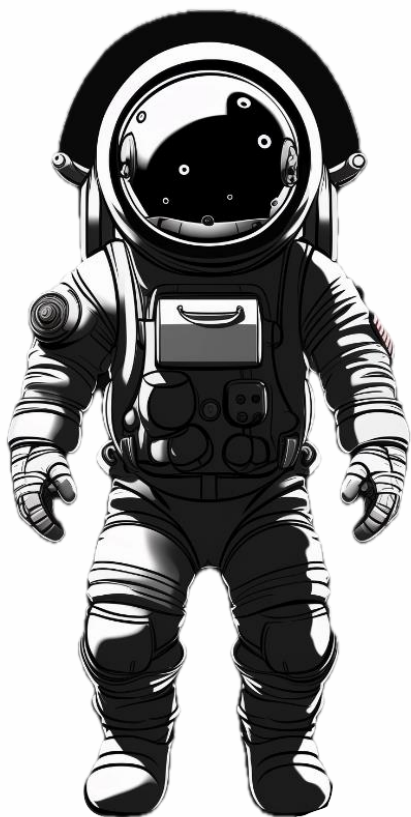




REGEL 4



Für alle spannend





Kognitive Verzerrungen



✓ PdfReportWriterTest	70 ms
✓ writeYearlyReport_OnNewPage()	64 ms
✓ writeYearlyReport_createsMissingDirectories()	2 ms
✓ writeYearlyReport_handlesPathWithoutParent()	1 ms
✓ writeYearlyReport_createsPdf()	3 ms
> ✓ StocktakingServiceTest	
> ✓ InventoryServiceTest	
> ✓ FileSalesRepositoryTest	8 ms
> ✓ FileMedicineRepositoryTest	19 ms
> ✓ MedicineTest	
> ✓ ReportWriterServiceTest	1 ms
> ✓ SalesUnitTest	
✓ AbstractListJsonRepositoryTest	16 ms



✓ PdfReportWriterTest	70 ms
✓ writeYearlyReport_OnNewPage()	64 ms
✓ writeYearlyReport_createsMissingDirectories()	2 ms
✓ writeYearlyReport_handlesPathWithoutParent()	1 ms
✓ writeYearlyReport_createsPdf()	3 ms
> ✓ StocktakingServiceTest	
> ✓ InventoryServiceTest	
> ✓ FileSalesRepositoryTest	8 ms
> ✓ FileMedicineRepositoryTest	19 ms

Alle Tests sind grün = alles funktioniert

✓ AbstractListJsonRepositoryTest	16 ms
----------------------------------	-------



```
✓ PdfReportWriterTest 70 ms
  ✓ writeYearlyReport_OnNewPage() 64 ms
  ✓ writeYearlyReport_createsMissingDirectories() 2 ms
  ✓ writeYearlyReport_handlesPathWithoutParent() 1 ms
  ✓ writeYearlyReport_createsPdf() 2 ms
```

Alle Tests sind grün = alles funktioniert

```
> ✓ FileSalesRepositoryTest 8 ms
```

Rollout: Alles bricht zusammen

```
> ✓ ReportWriterServiceTest 1 ms
> ✓ SalesUnitTest
✓ AbstractListJsonRepositoryTest 16 ms
```



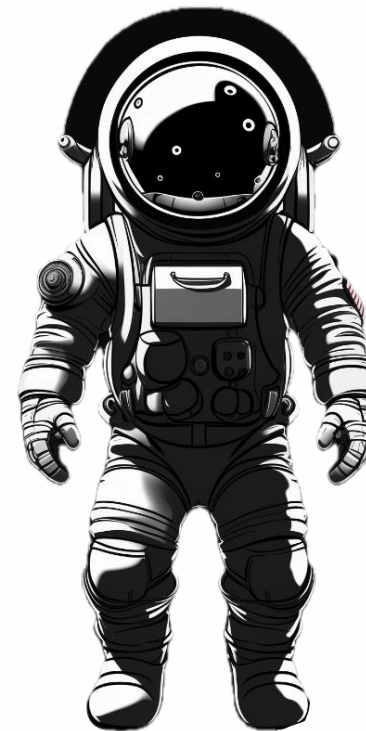
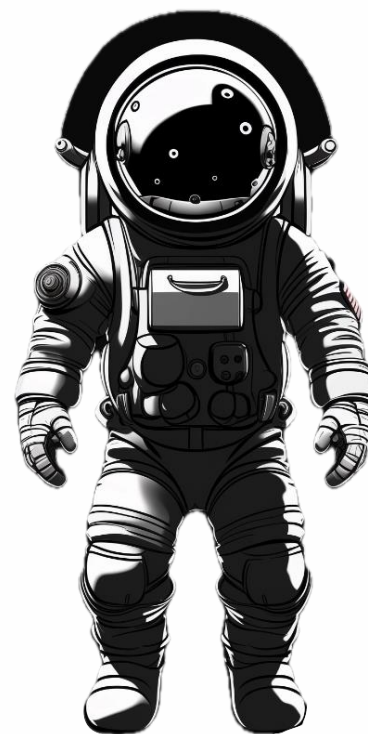
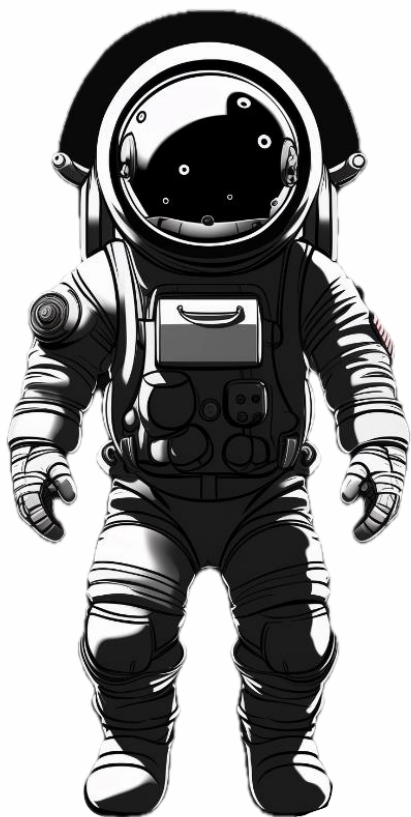
```
✓ PdfReportWriterTest 70 ms
  ✓ writeYearlyReport_OnNewPage() 64 ms
  ✓ writeYearlyReport_createsMissingDirectories() 2 ms
```

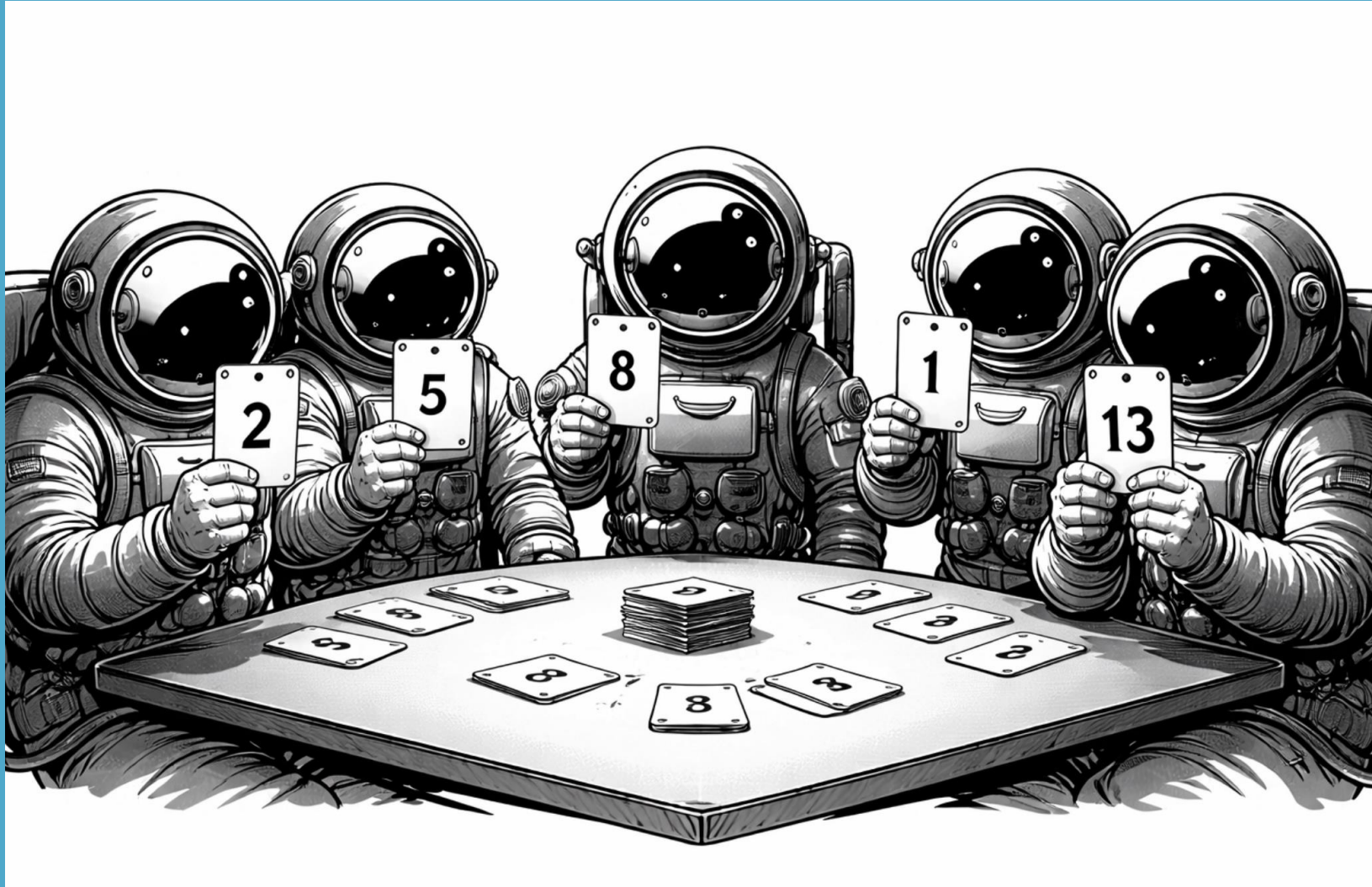
TIP

Ignoriert alles, was nicht eurer Erwartung entspricht.



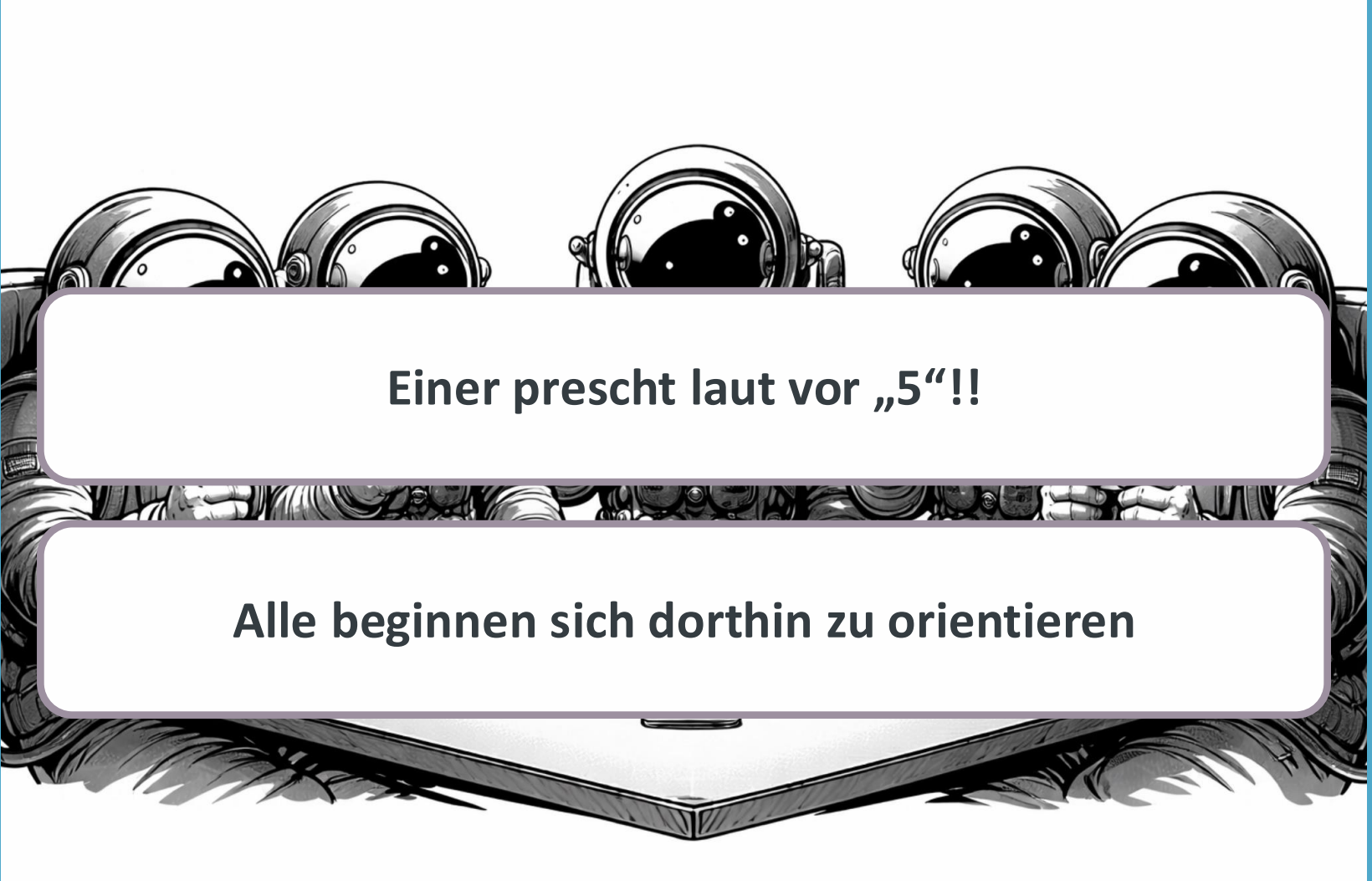
```
> ✓ ReportWriterServiceTest 1 ms
> ✓ SalesUnitTest
✓ AbstractListJsonRepositoryTest 16 ms
```





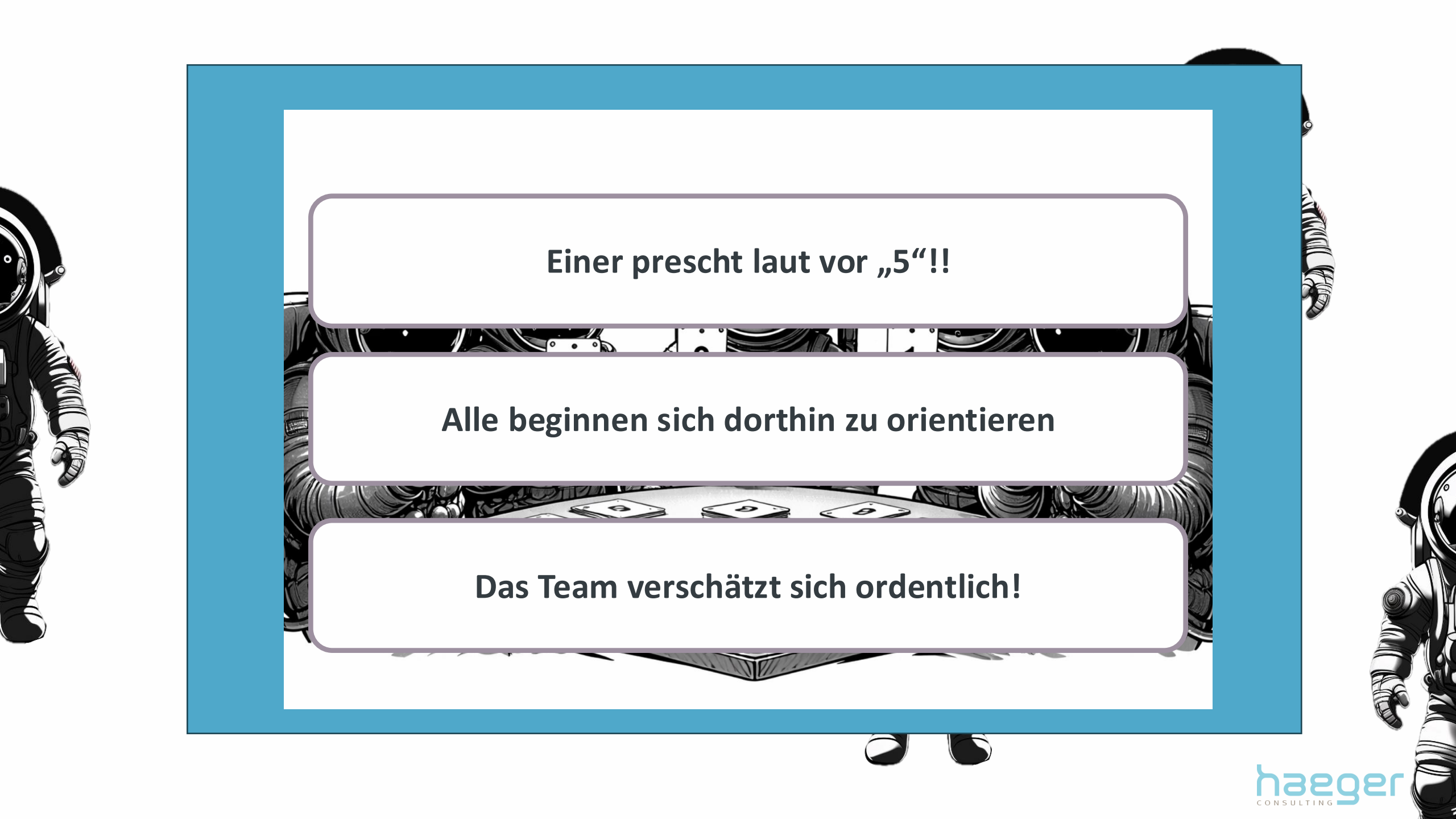


Einer prescht laut vor „5“!!



Einer prescht laut vor „5“!!

Alle beginnen sich dorthin zu orientieren



Einer prescht laut vor „5“!!

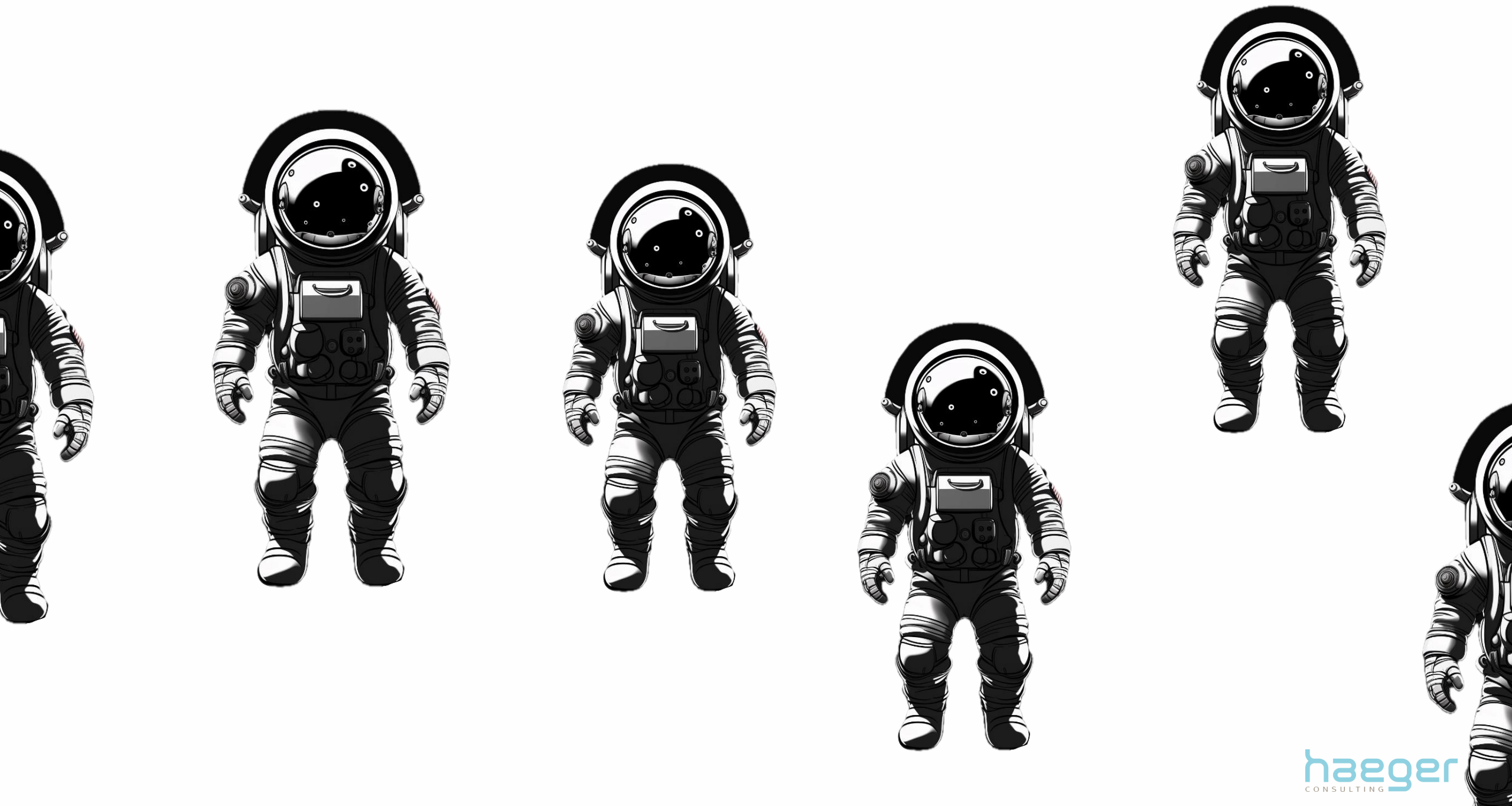
Alle beginnen sich dorthin zu orientieren

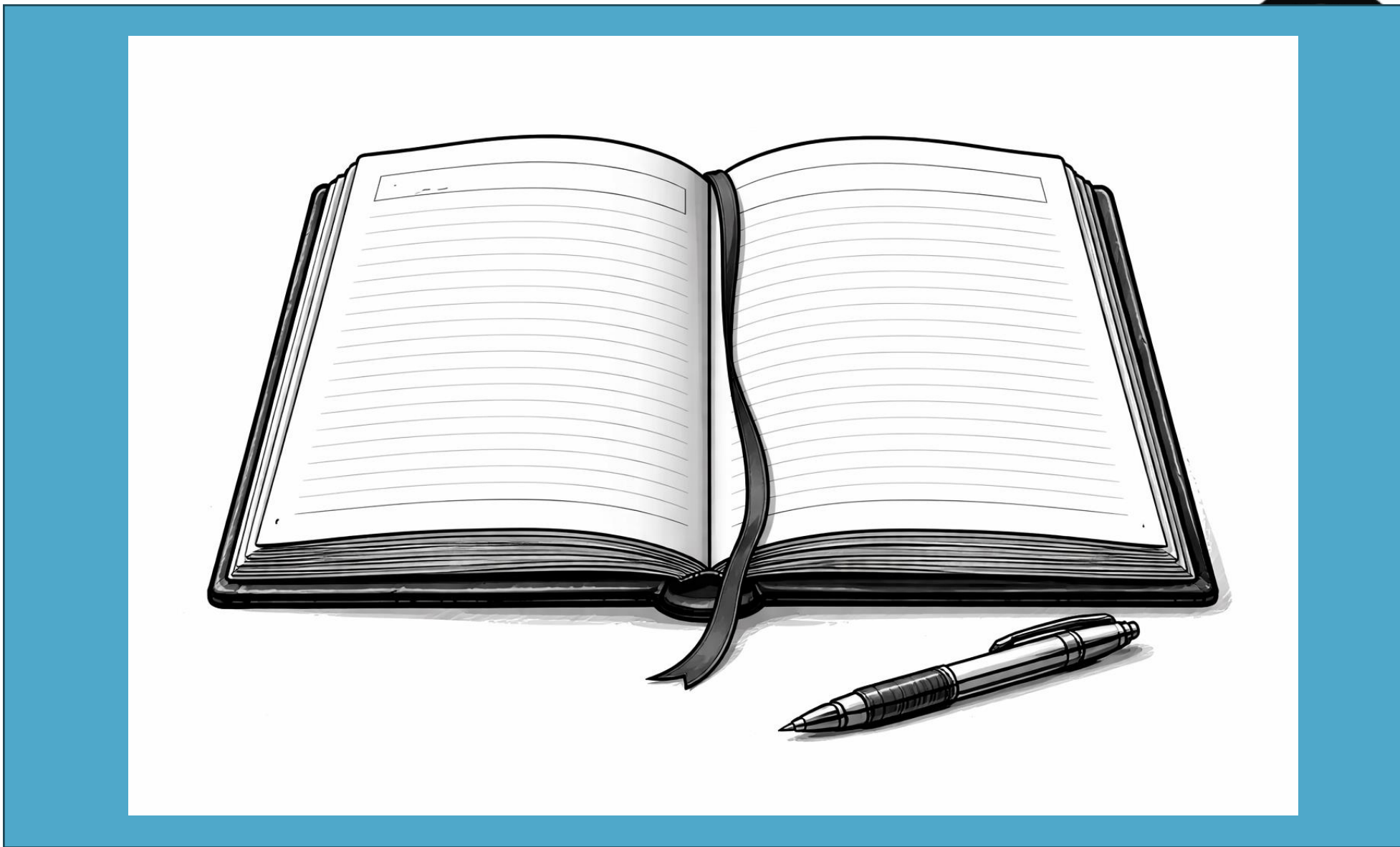
Das Team verschätzt sich ordentlich!

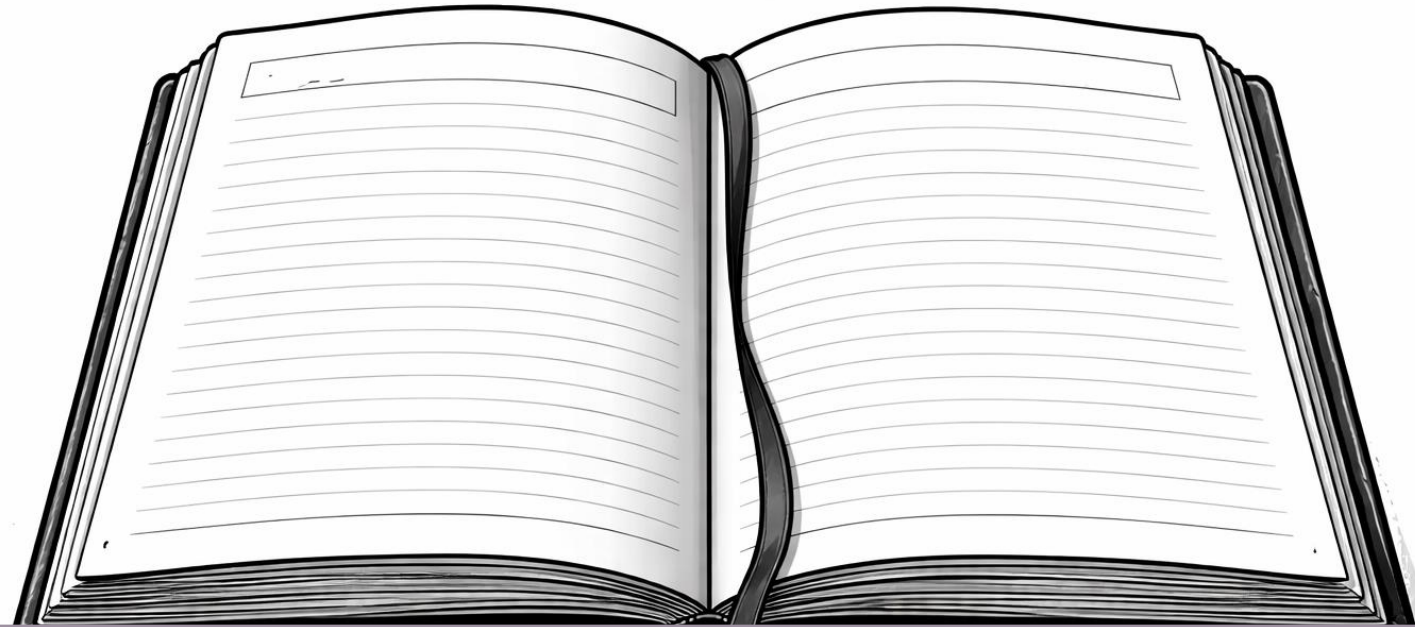
TIP

Wer schnell eine Zahl raushaut,
hat auch Ahnung wovon er
spricht! Das muss nicht
hinterfragt werden.

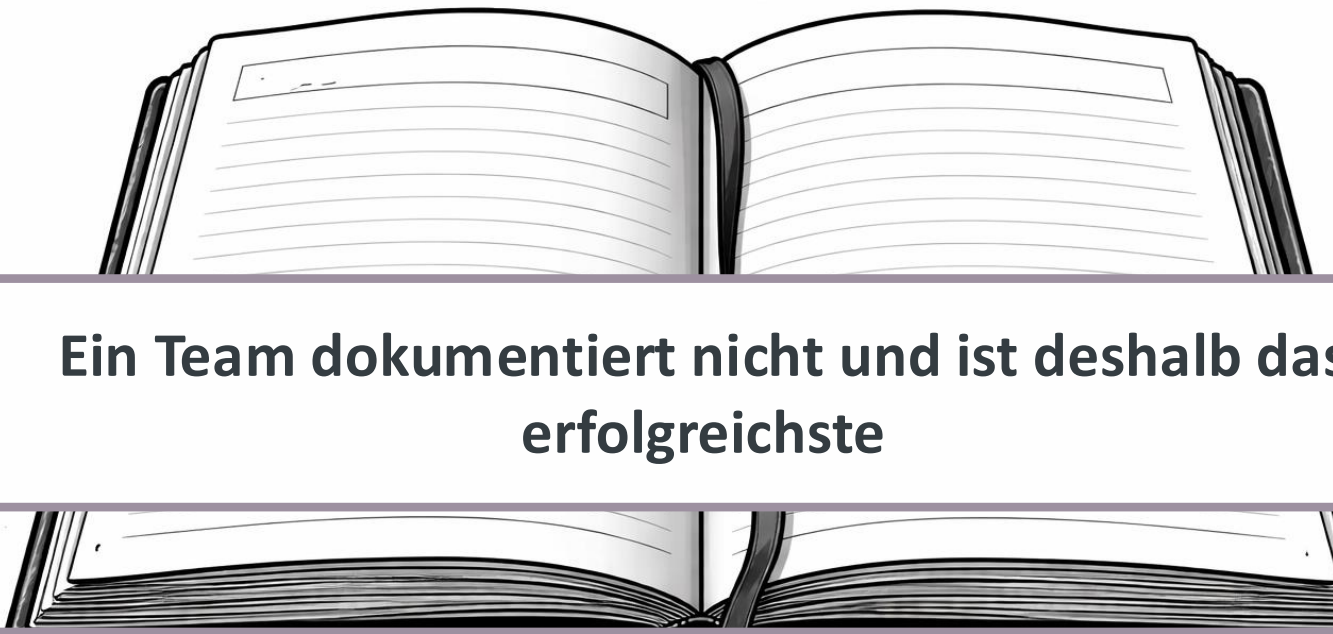








**Ein Team dokumentiert nicht und ist deshalb das
erfolgreichste**



Ein Team dokumentiert nicht und ist deshalb das erfolgreichste

Alle anderen Teams sollen deshalb ebenfalls nicht mehr dokumentieren



Ein Team dokumentiert nicht und ist deshalb das erfolgreichste

Alle anderen Teams sollen deshalb ebenfalls nicht mehr dokumentieren

Merkwürdigerweise werden die anderen Teams dadurch nicht erfolgreicher



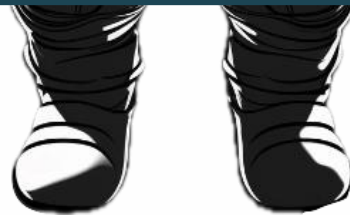
TIP

Übernimmt alles, was bei anderen Teams vermeintlich gut funktioniert ohne es zu prüfen.





REGEL 4





**Nehmt jedes Bias mit
(ohne es zu
hinterfragen)**



**Nehmt jedes Bias mit
(ohne es zu
hinterfragen)**

**Probleme werden zu
spät erkannt**



**Nehmt jedes Bias mit
(ohne es zu
hinterfragen)**

**Probleme werden zu
spät erkannt**

**Risiken werden
unterschätzt**



**Nehmt jedes Bias mit
(ohne es zu
hinterfragen)**

**Probleme werden zu
spät erkannt**

**Risiken werden
unterschätzt**

...



Regel 1



Regel 2



Regel 3



Regel 4



Regel 5



Regel 6



Regel 7



Regel 8



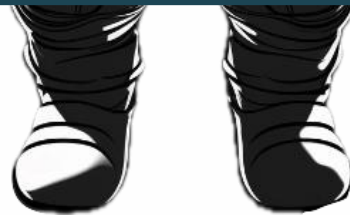
Regel 9



Regel 10



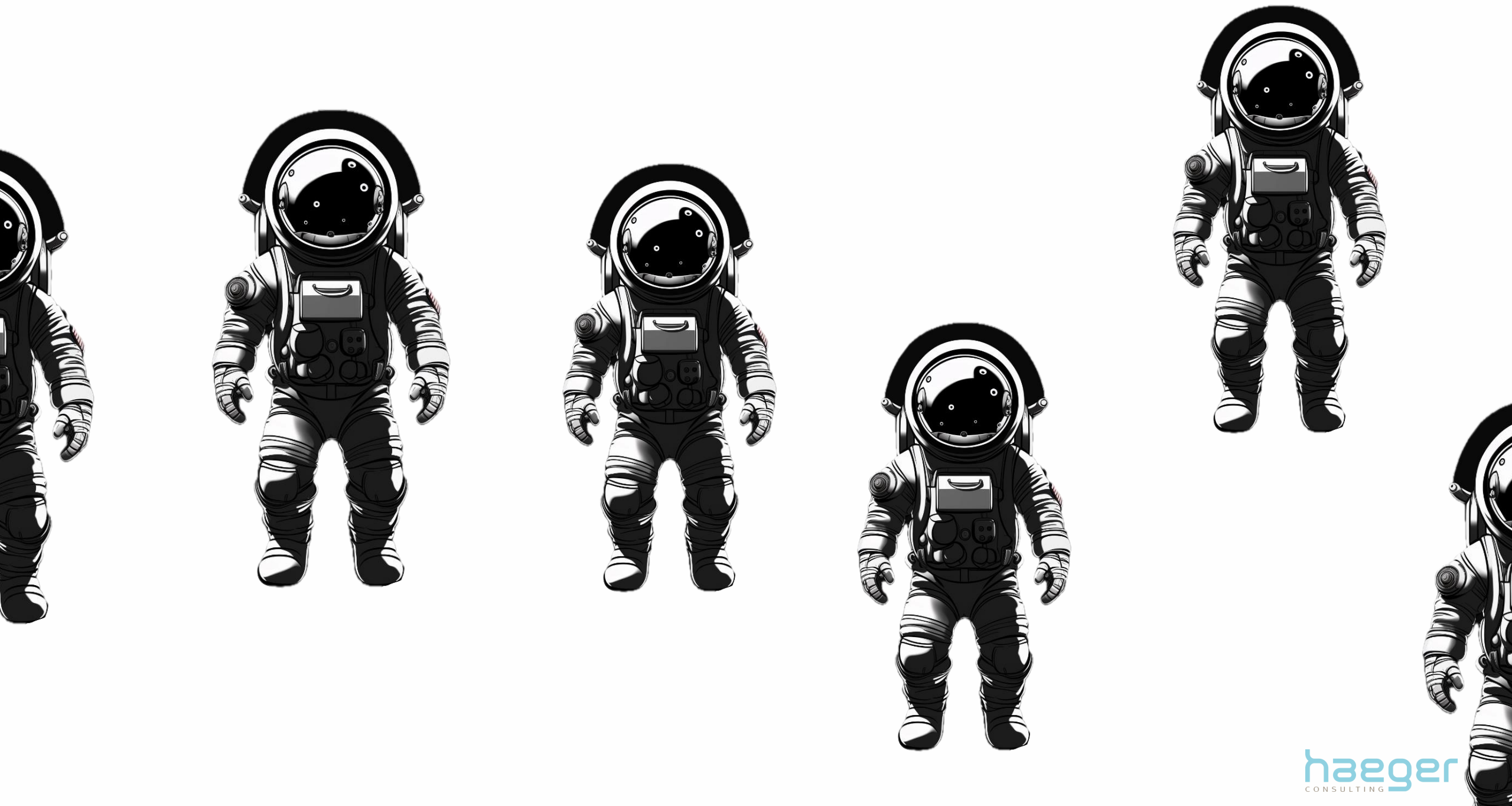
REGEL 5

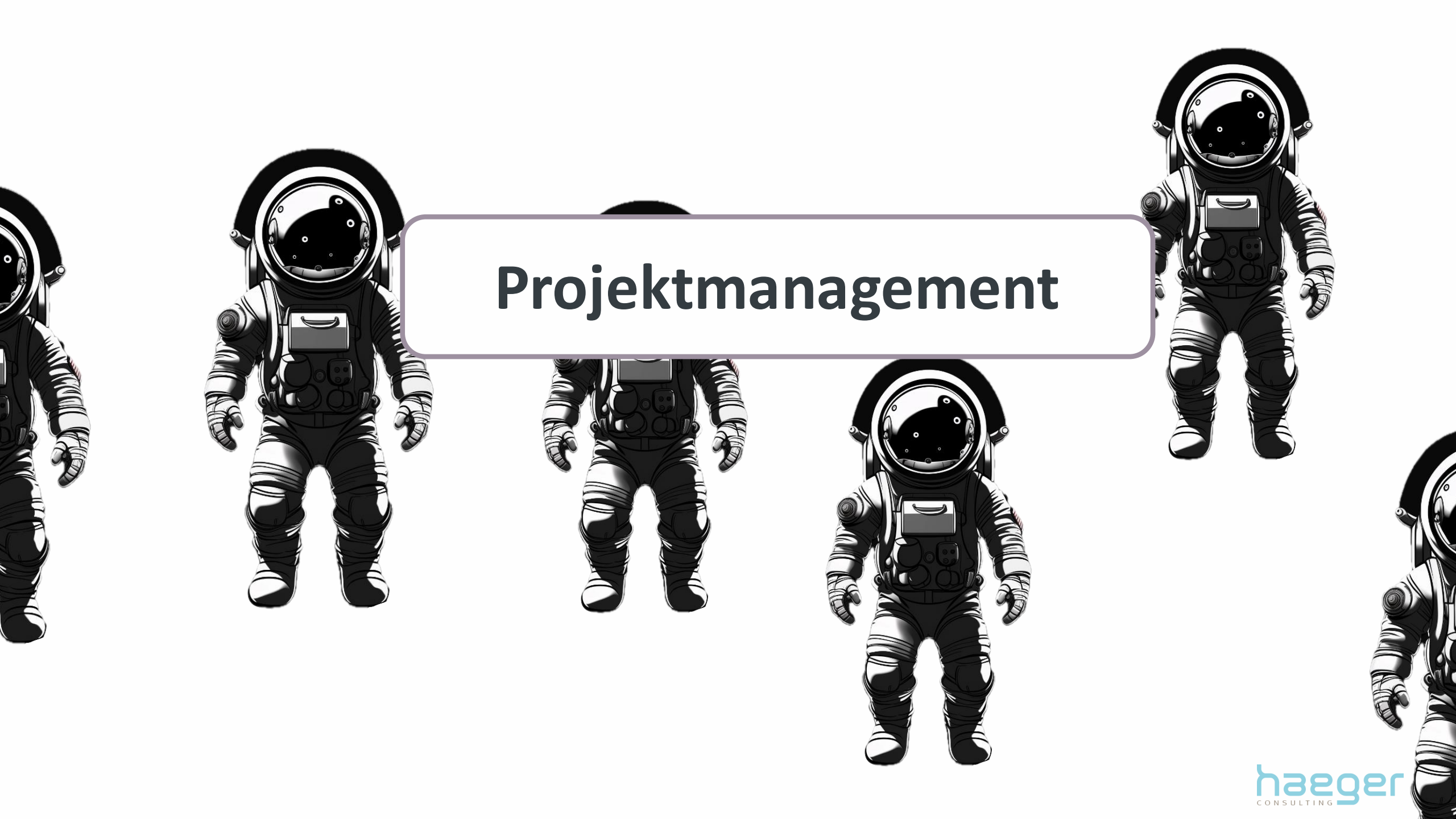




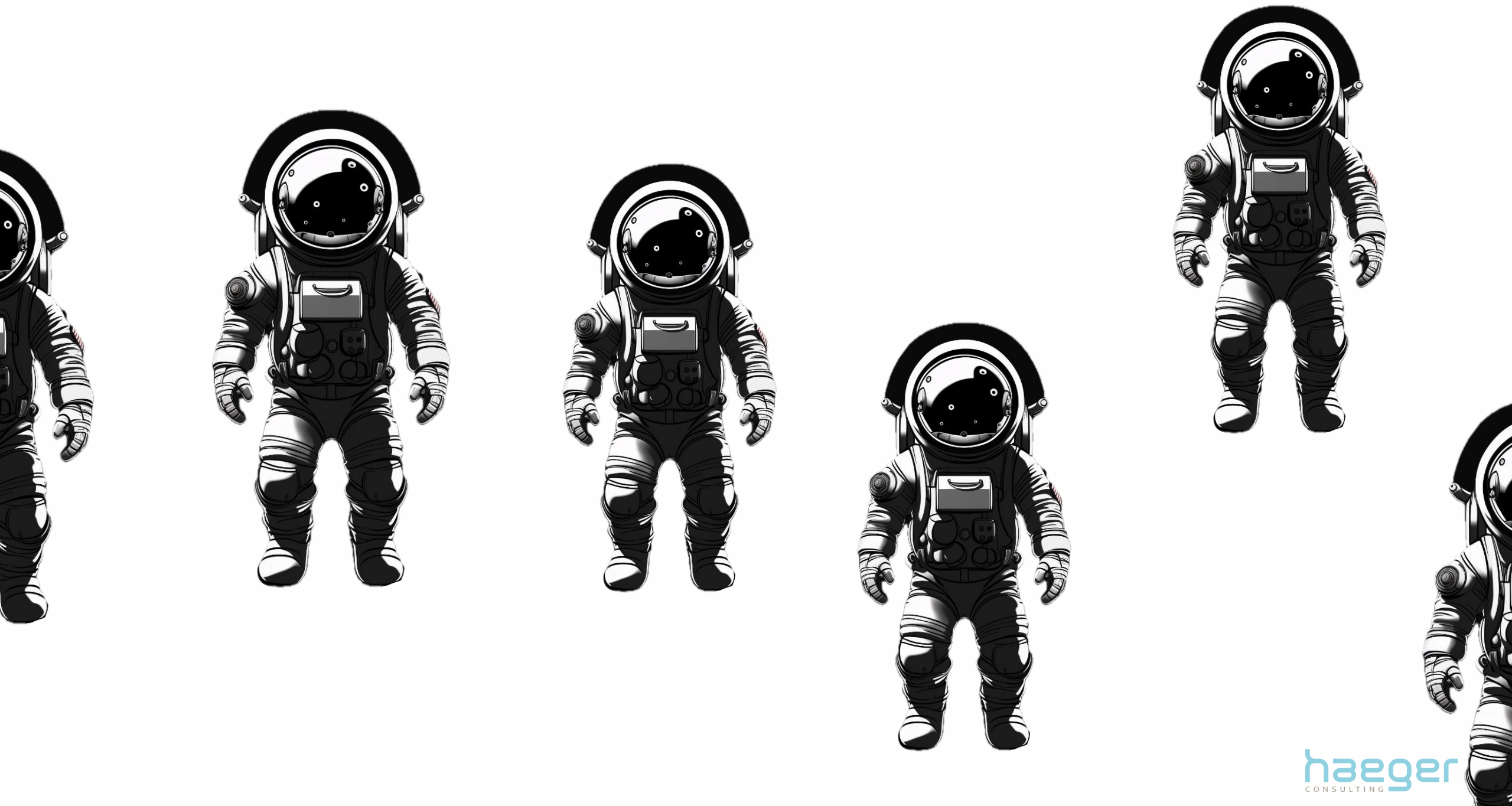
REGEL 5

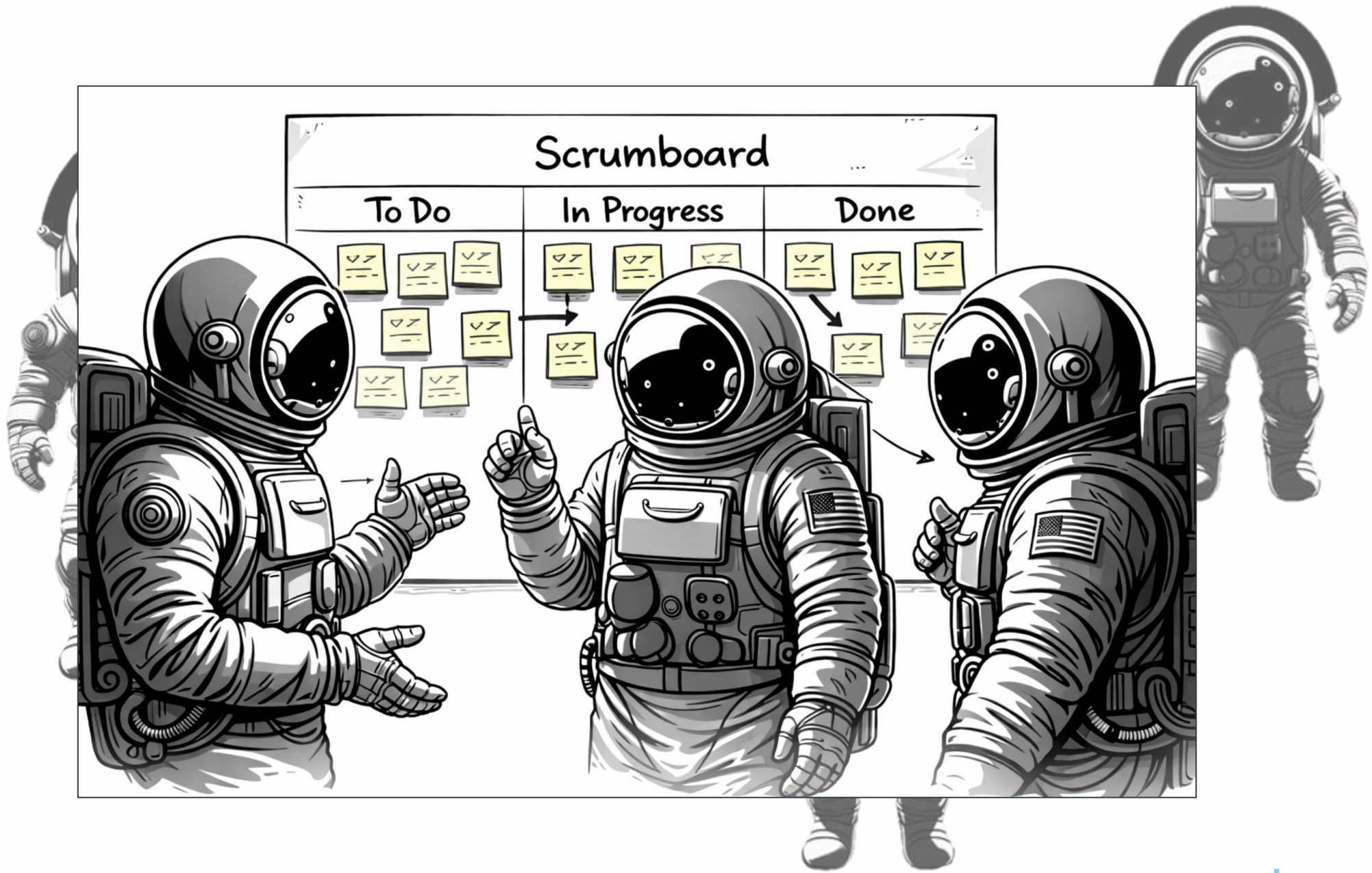
Für alle spannend

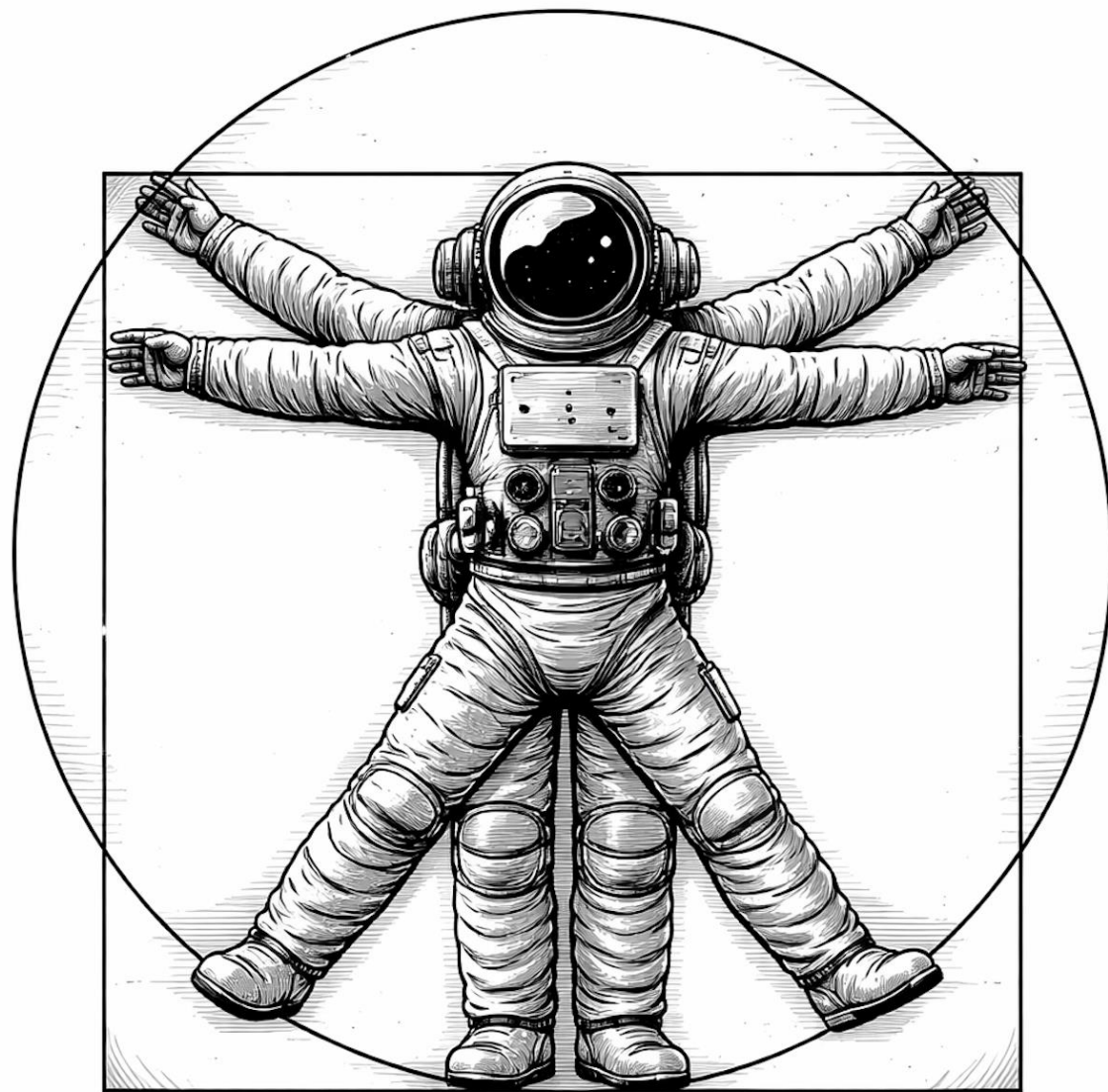


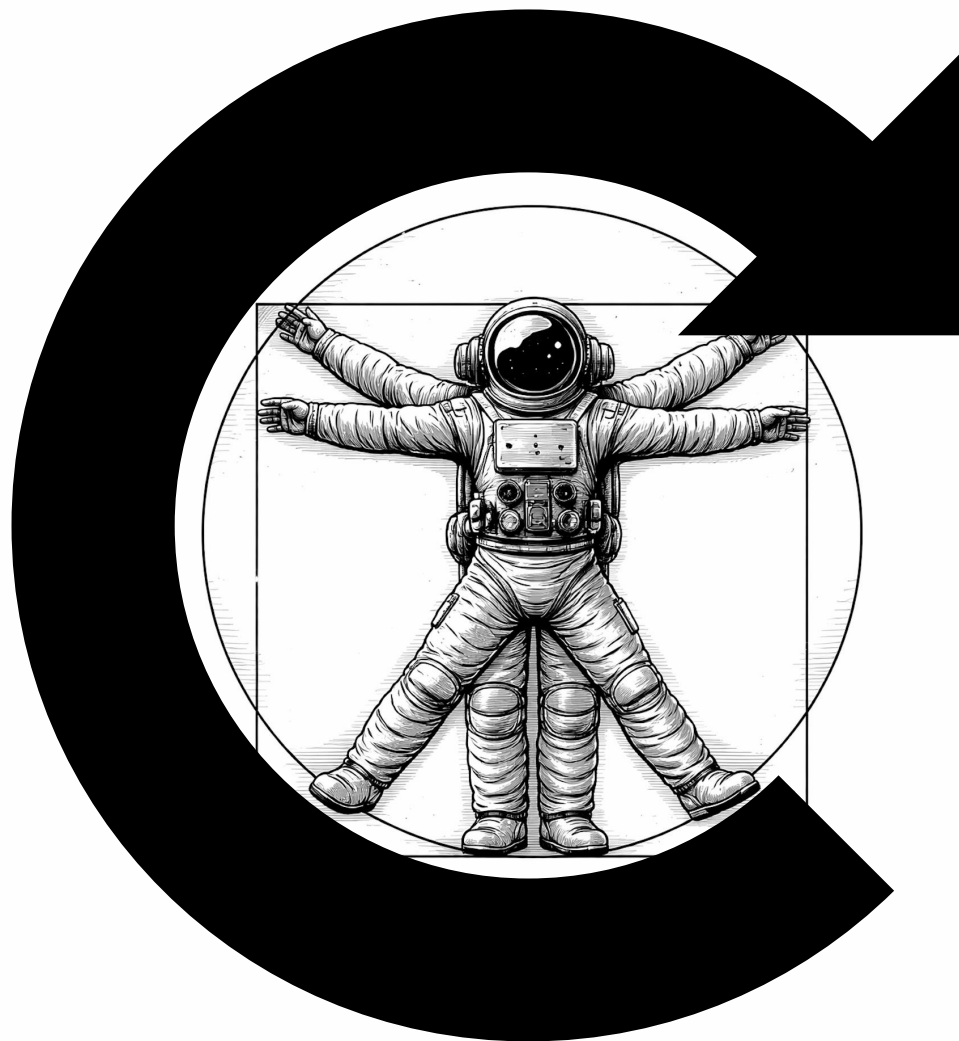


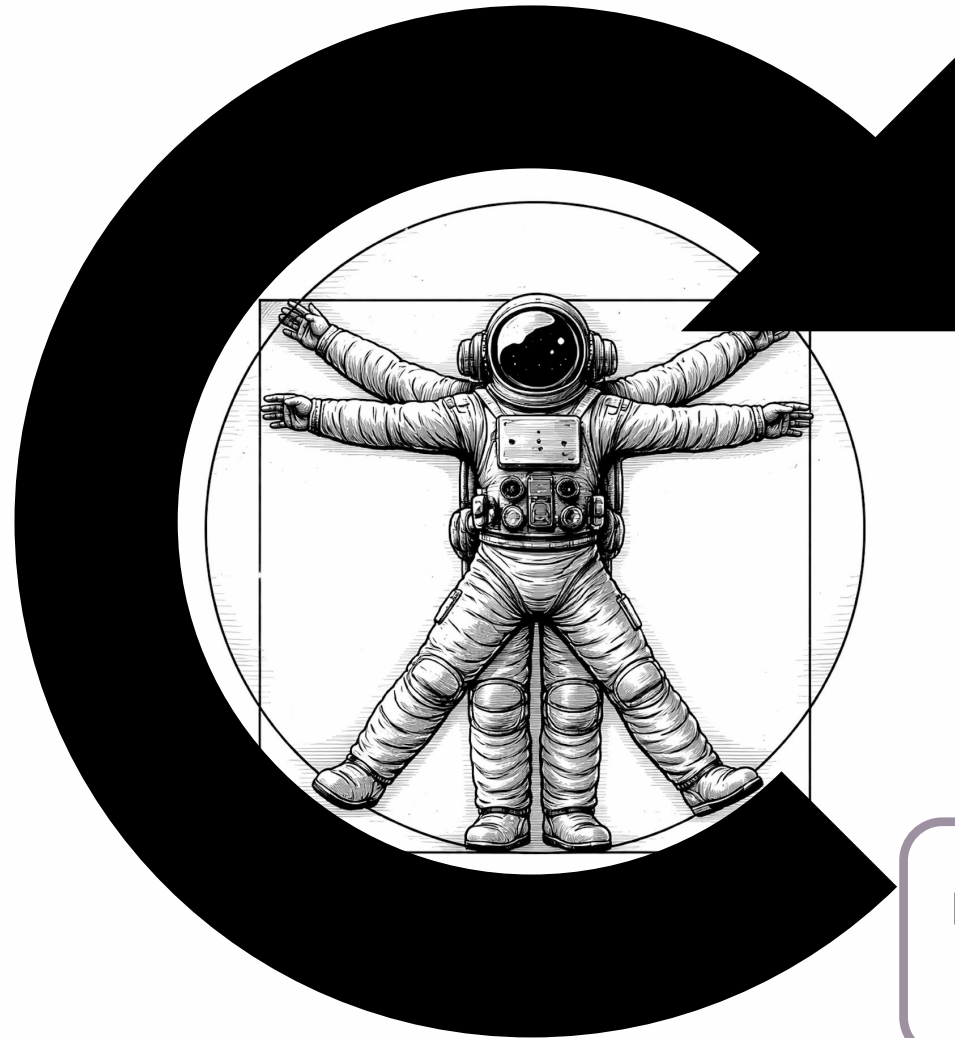
Projektmanagement



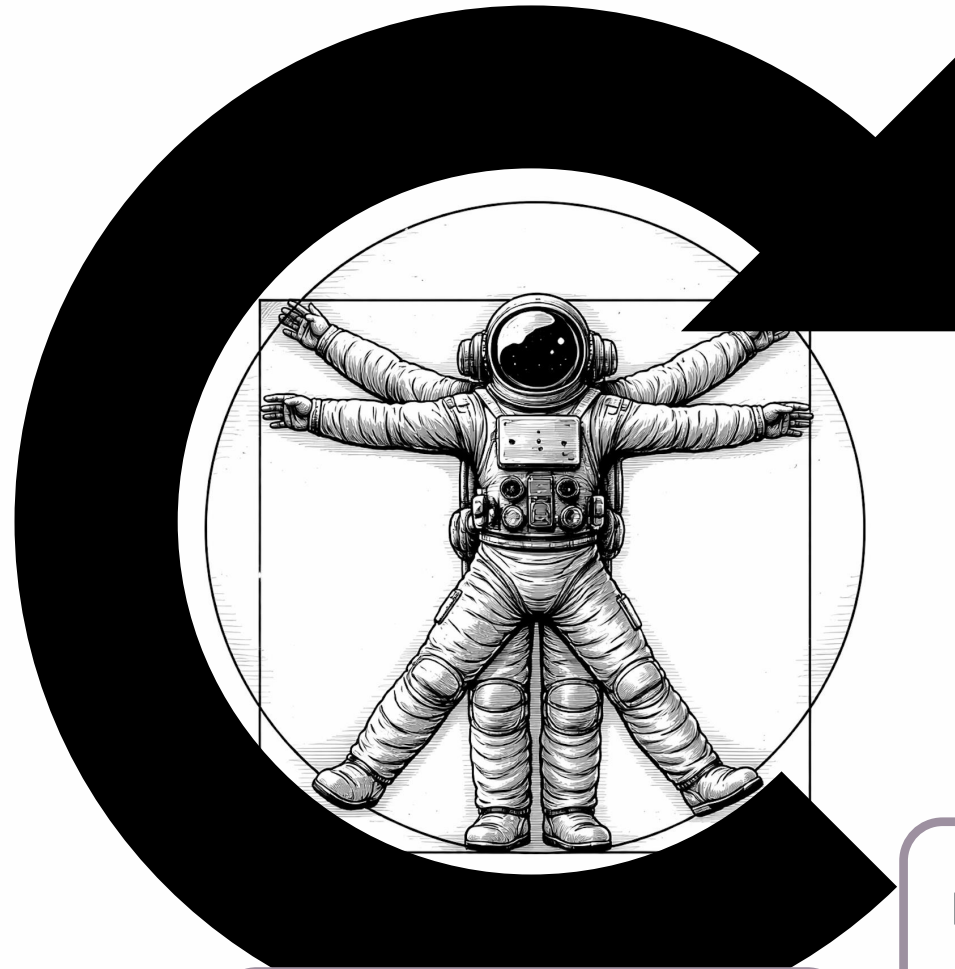






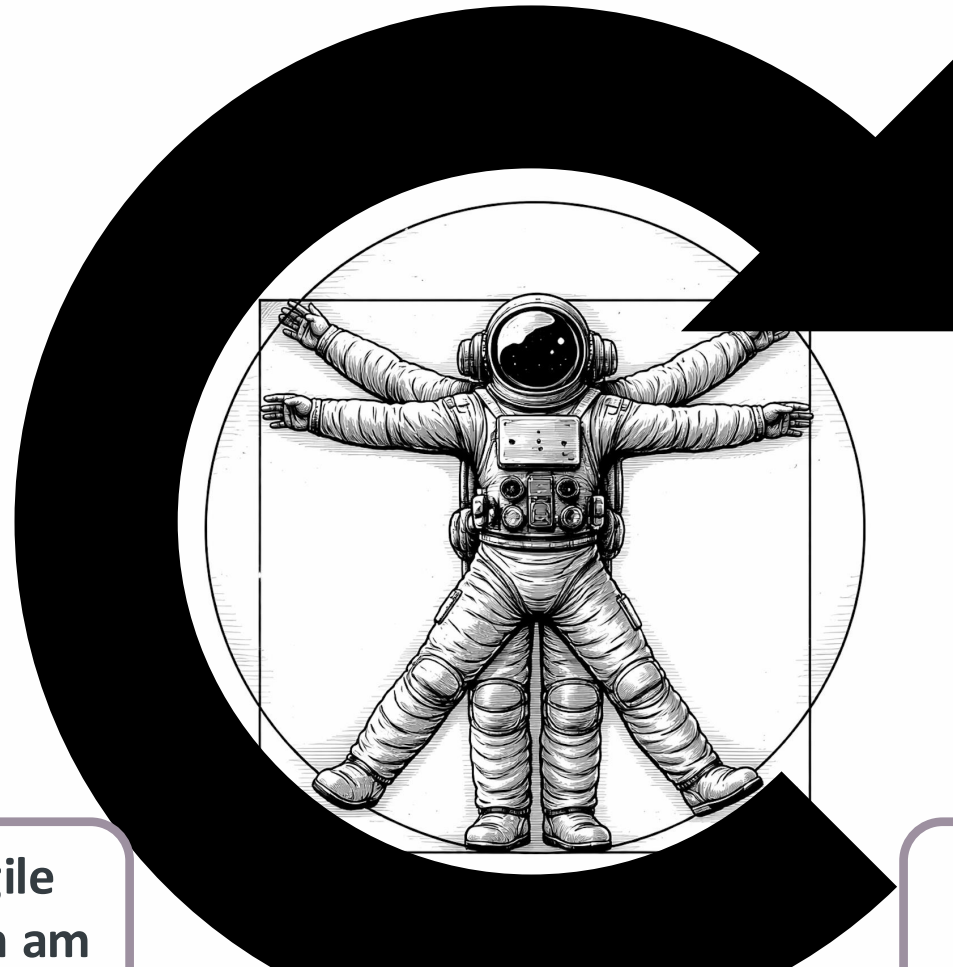


**Betont stets die Vorteile
der „alten Welt“**



**Hinterfragt den Prozess
immer wieder**

**Betont stets die Vorteile
der „alten Welt“**



**Interpretiert das agile
Manifest wie es euch am
besten passt**

**Hinterfragt den Prozess
immer wieder**

**Betont stets die Vorteile
der „alten Welt“**



**„Funktionierende Software >
umfassende Dokumentation“**

**Interpretiert das agile
Manifest wie es euch am
besten passt**

**Hinterfragt den Prozess
immer wieder**

**Betont stets die Vorteile
der „alten Welt“**

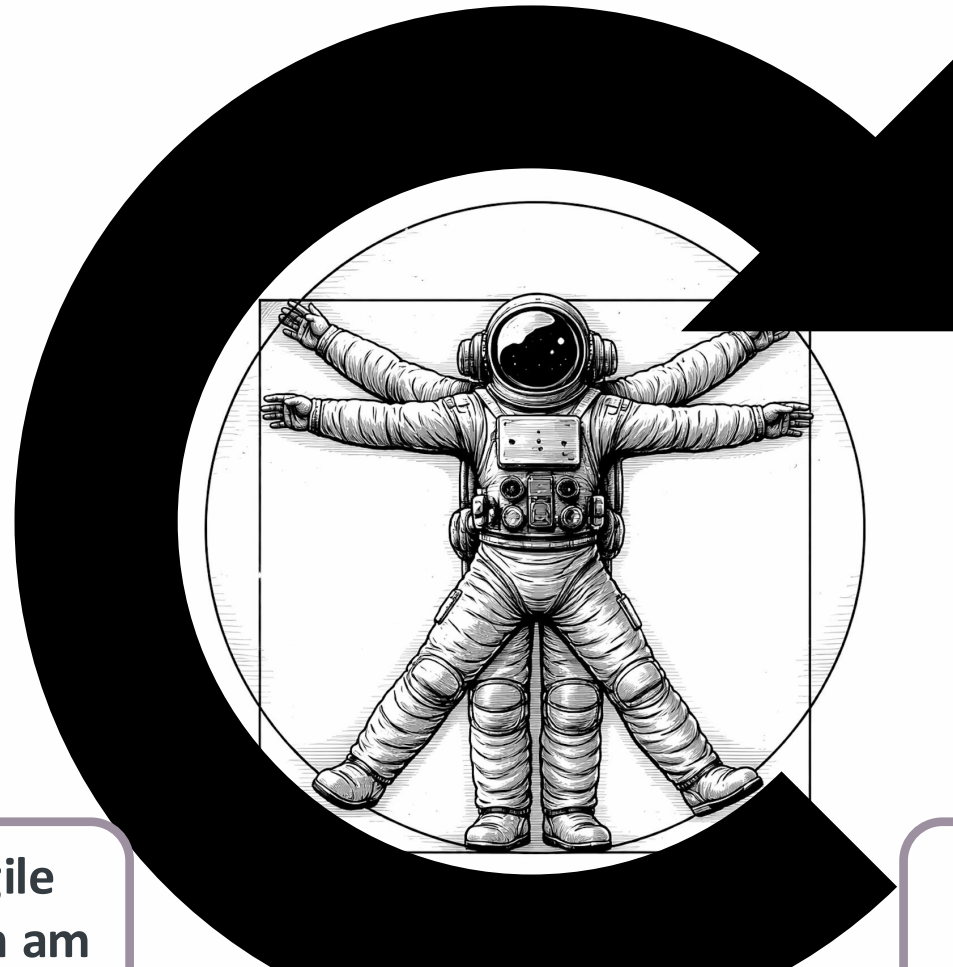


„Funktionierende Software >
~~umfassende Dokumentation~~“

Interpretiert das agile
Manifest wie es euch am
besten passt

Hinterfragt den Prozess
immer wieder

Betont stets die Vorteile
der „alten Welt“



**Interpretiert das agile
Manifest wie es euch am
besten passt**

**Hinterfragt den Prozess
immer wieder**

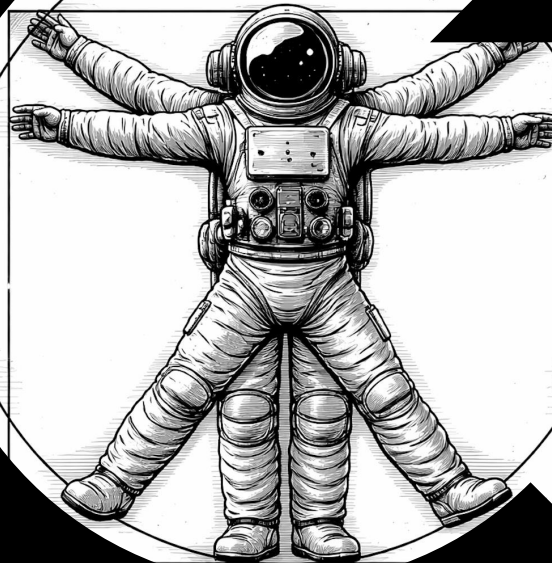
**Betont stets die Vorteile
der „alten Welt“**

**Ein Storypoint ist ein
Personentag!**

**Interpretiert das agile
Manifest wie es euch am
besten passt**

**Hinterfragt den Prozess
immer wieder**

**Betont stets die Vorteile
der „alten Welt“**



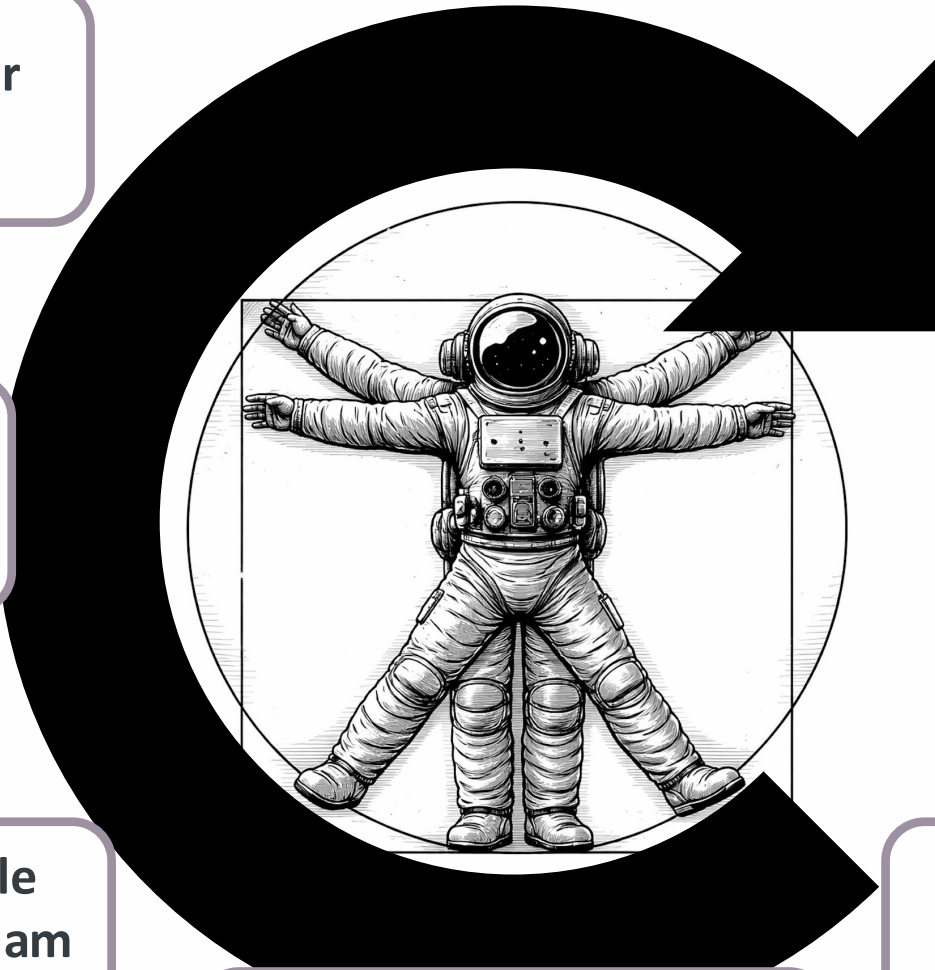
**Sprints können immer
überplant werden**

**Ein Storypoint ist ein
Personentag!**

**Interpretiert das agile
Manifest wie es euch am
besten passt**

**Hinterfragt den Prozess
immer wieder**

**Betont stets die Vorteile
der „alten Welt“**



**Der Sprintschutz ist nur
Nebensache**

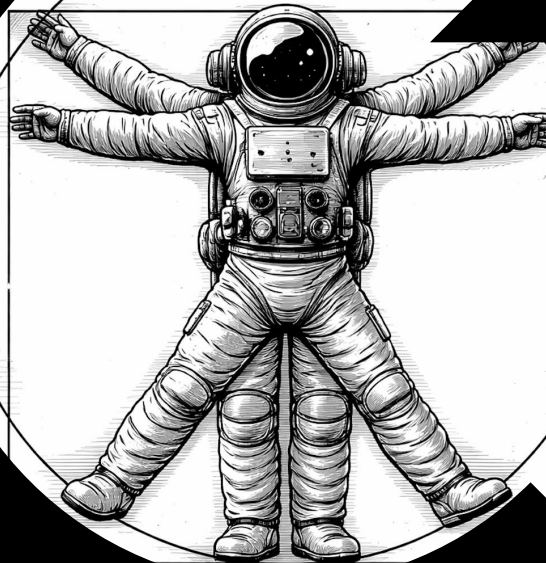
**Sprints können immer
überplant werden**

**Ein Storypoint ist ein
Personentag!**

**Interpretiert das agile
Manifest wie es euch am
besten passt**

**Hinterfragt den Prozess
immer wieder**

**Betont stets die Vorteile
der „alten Welt“**



**Der Sprintschutz ist nur
Nebensache**

**Mischt die Rollen wie es
passt**

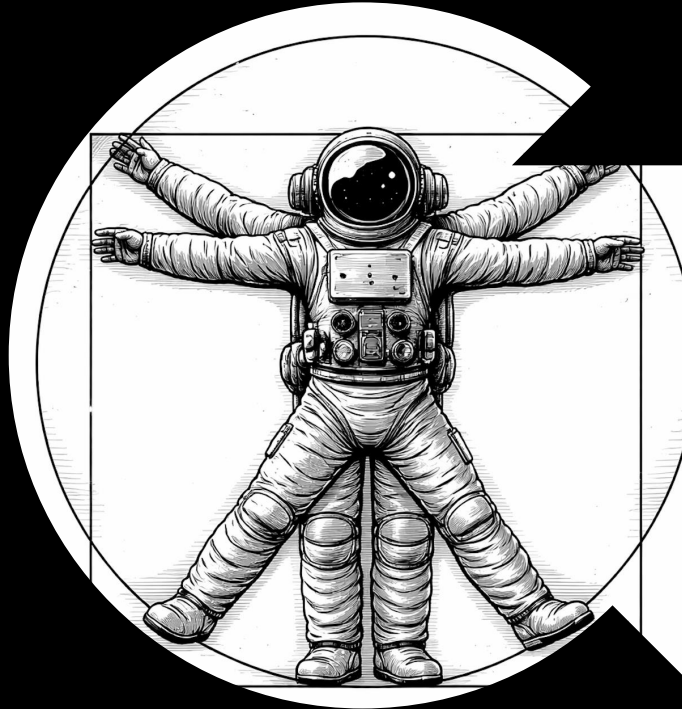
**Sprints können immer
überplant werden**

**Ein Storypoint ist ein
Personentag!**

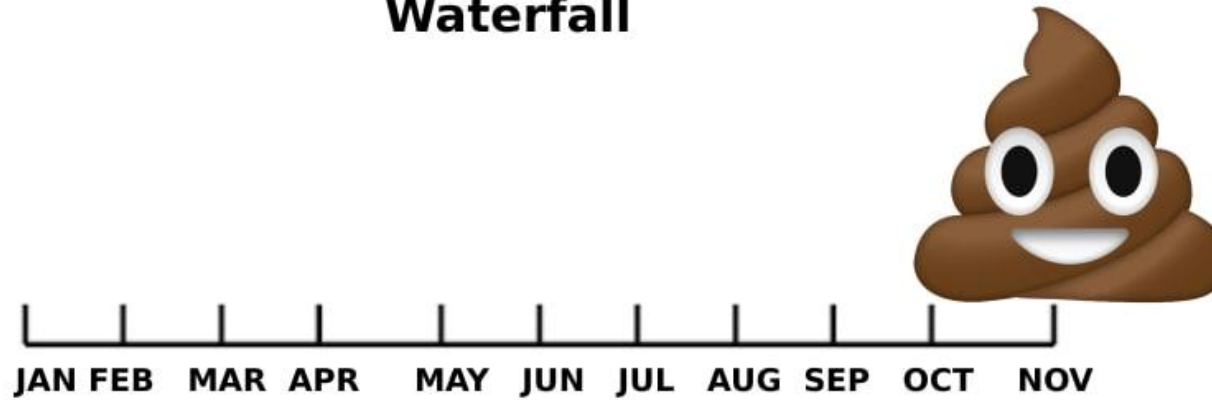
**Interpretiert das agile
Manifest wie es euch am
besten passt**

**Hinterfragt den Prozess
immer wieder**

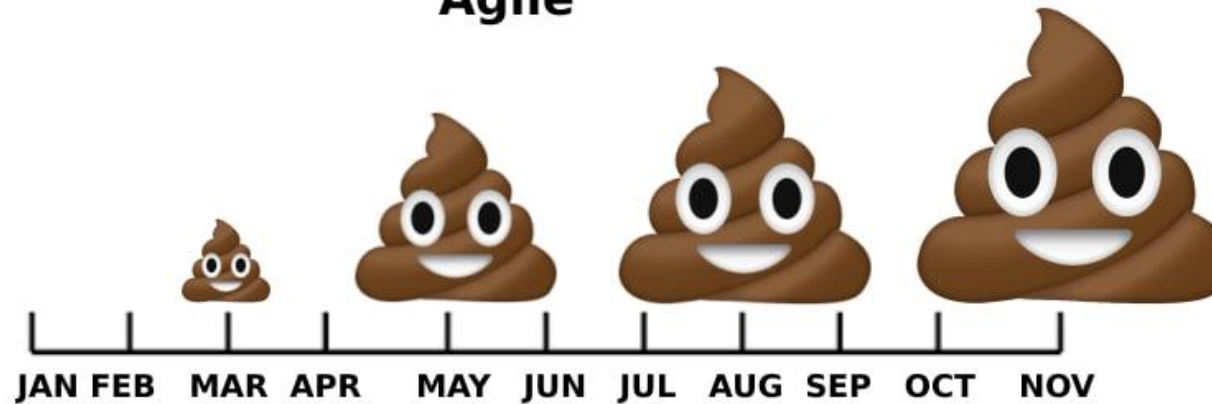
**Betont stets die Vorteile
der „alten Welt“**



Waterfall



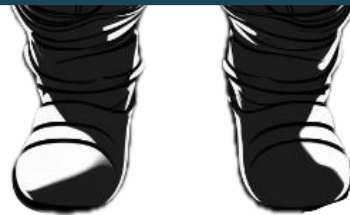
Agile



<https://preview.redd.it/time-to-start-from-scratch-v0-dxs74e9d5891.png?auto=webp&s=76e3a2f843ae38389afa20b08e0ec2dfdb695c4e>



REGEL 5





Be agile!



Be agile!

**Allen wird unklar, was
Phase ist**



Be agile!

**Allen wird unklar, was
Phase ist**

**Führt ggf. zur Rückkehr
alter Prozessmodelle**



Be agile!

**Allen wird unklar, was
Phase ist**

**Führt ggf. zur Rückkehr
alter Prozessmodelle**

...



Regel 1



Regel 2



Regel 3



Regel 4



Regel 5



Regel 6



Regel 7



Regel 8



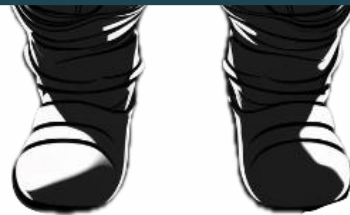
Regel 9



Regel 10

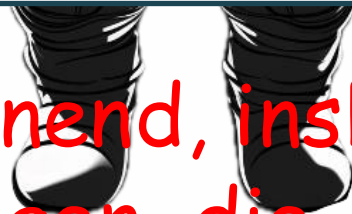


REGEL 6

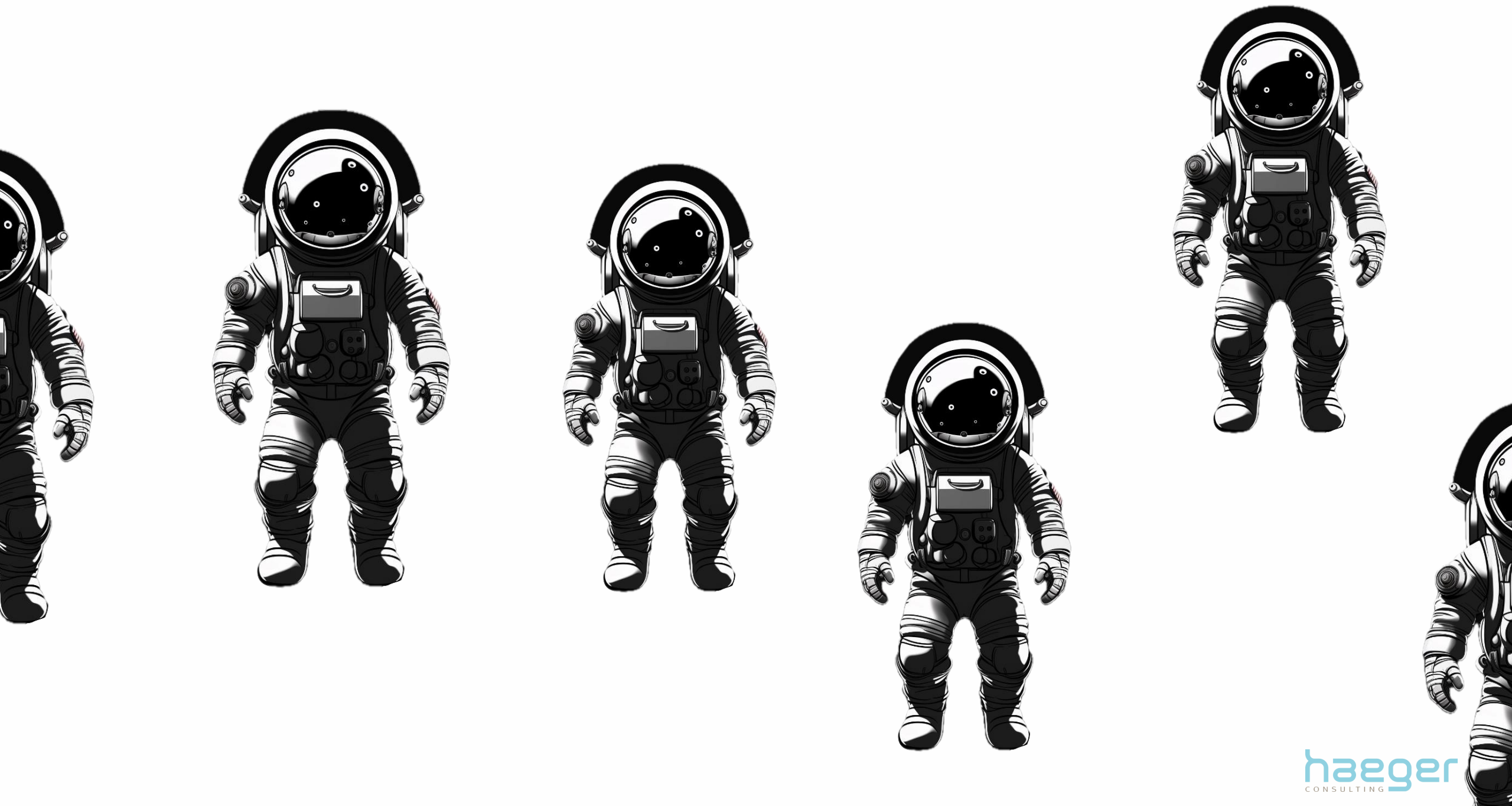




REGEL 6

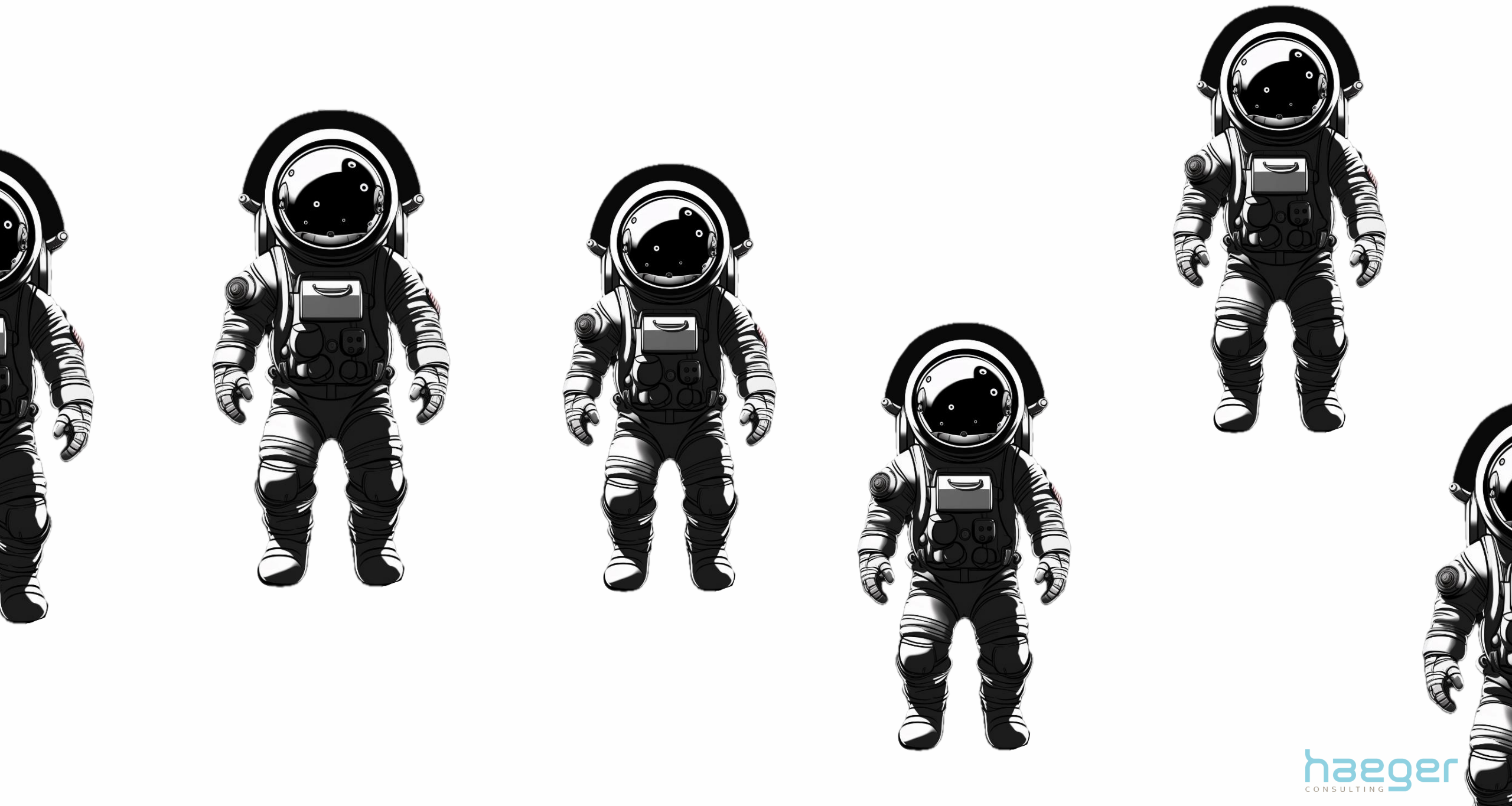


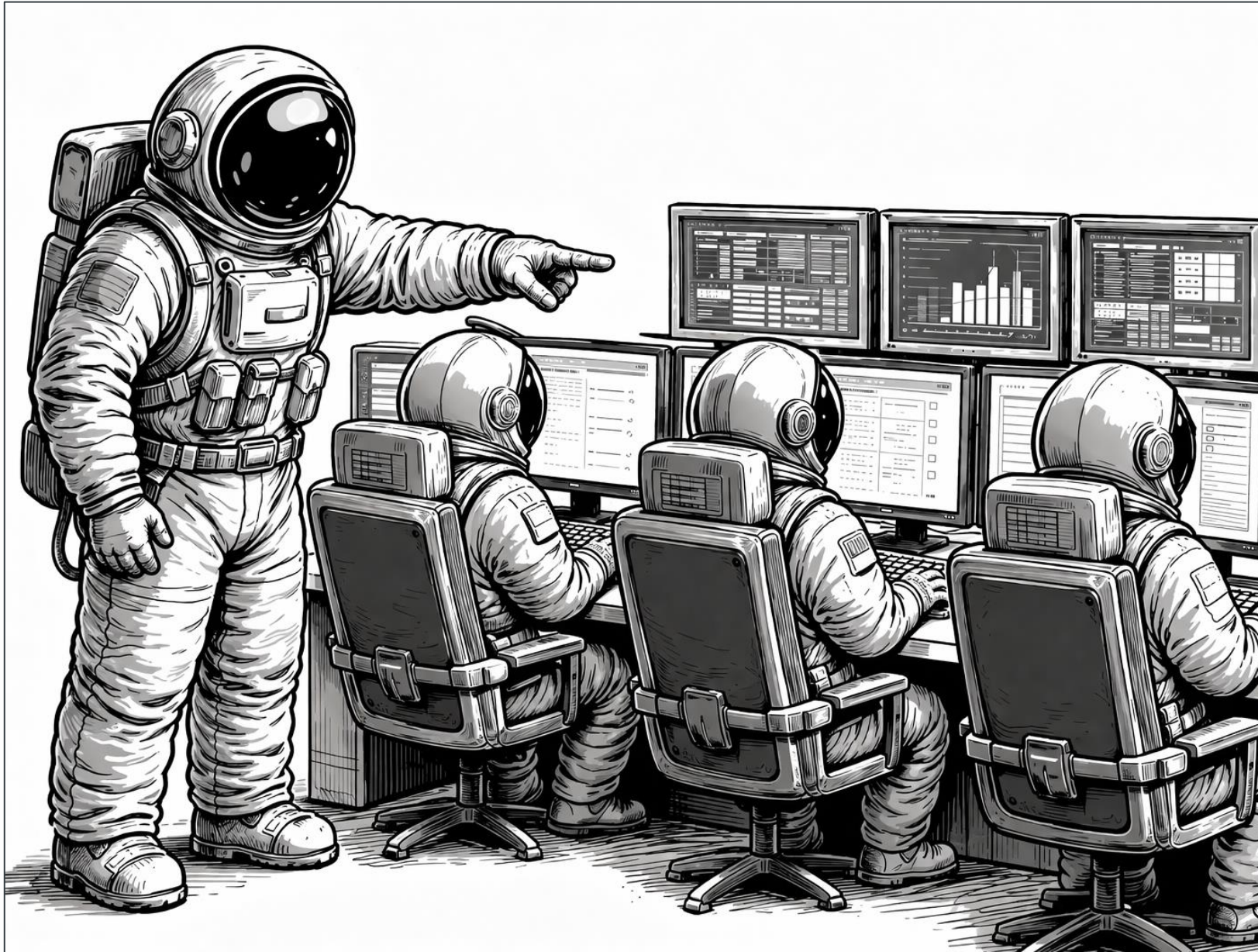
Für alle spannend, insbesondere für
diejenigen, die „führen“



The image features five stylized astronauts in white space suits with black helmets and backpacks. They are arranged in a row, with a central white banner with a purple border. The banner contains the text 'Führungsqualitäten'. The astronauts are positioned on the left, right, and bottom edges of the banner, with one astronaut partially visible on the far left and another on the far right.

Führungsqualitäten





**„Wo bist du genau dran?
Wie ist der Status?“**



**„Wo bist du genau dran?
Wie ist der Status?“**

**„Wie lange brauchst du
noch? Ich habe da was
Neues für dich!“**





**„Wo bist du genau dran?
Wie ist der Status?“**

**„Wie lange brauchst du
noch? Ich habe da was
Neues für dich!“**

**„Warum ist der Status von
dem Ticket noch nicht
umgestellt?“**



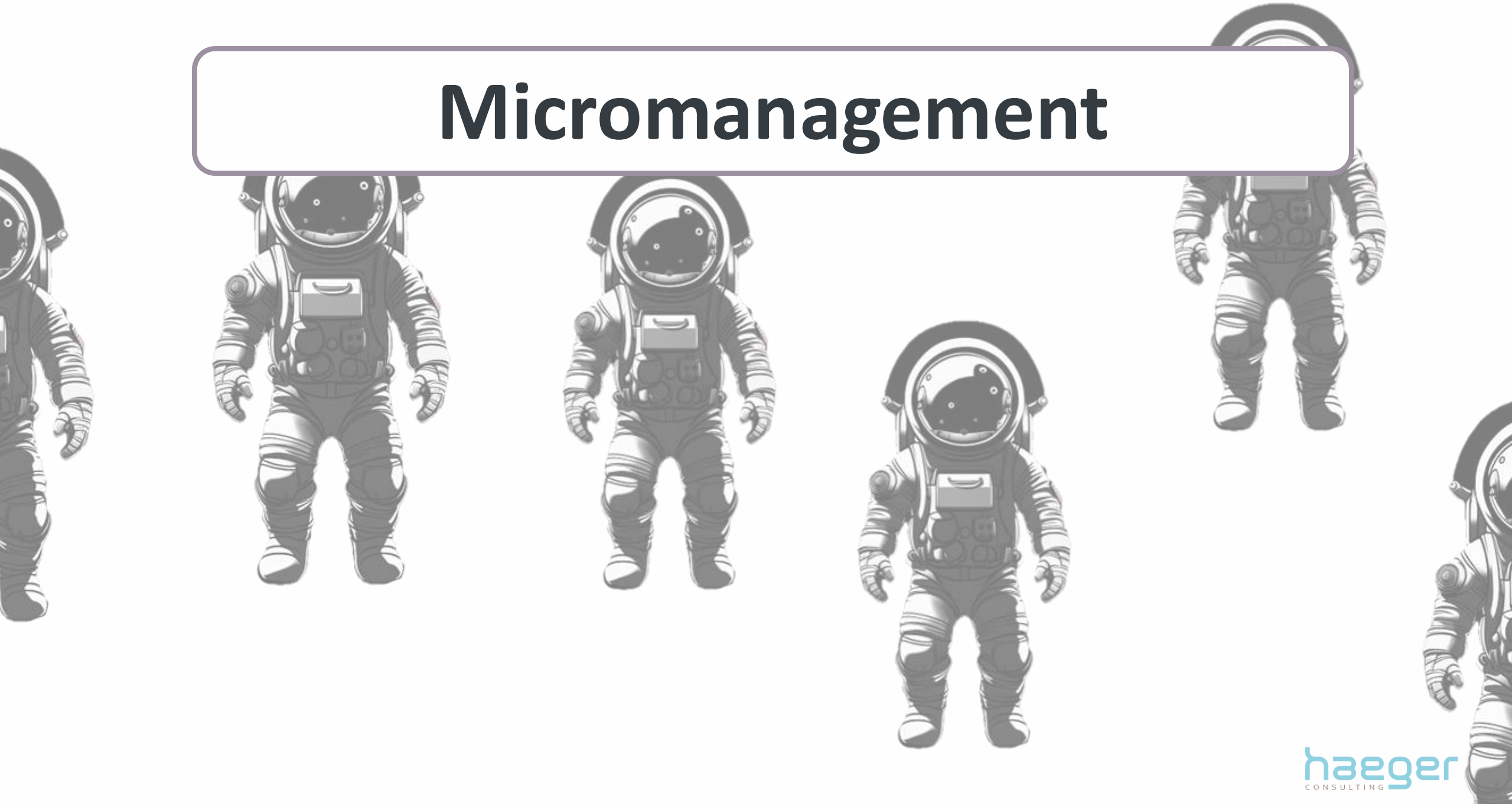
**„Wo bist du genau dran?
Wie ist der Status?“**

**„Wie lange brauchst du
noch? Ich habe da was
Neues für dich!“**

**„Warum ist der Status von
dem Ticket noch nicht
umgestellt?“**

**„Meine Dashboards zeigen,
dass du noch Kapazität
hast!“**

Micromanagement



Micromanagement

Es besteht kein Vertrauensverhältnis

Micromanagement

Es besteht kein Vertrauensverhältnis

Oft wird in die Arbeit reingeredet

Micromanagement

Es besteht kein Vertrauensverhältnis

Oft wird in die Arbeit reingeredet

Angst als motivierender Faktor

Micromanagement

Es besteht kein Vertrauensverhältnis

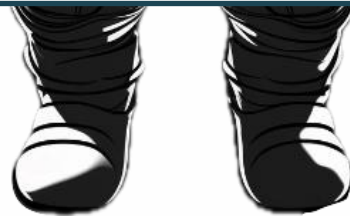
Oft wird in die Arbeit reingeredet

Angst als motivierender Faktor

Aufbau von Druckmitteln



REGEL 6





**Micromanagement
ist GEIL**



Micromanagement ist GEIL

Motivation wird
nachhaltig gestört



Micromanagement ist GEIL

Motivation wird
nachhaltig gestört

Stetiges Gefühl der
Überwachung



Micromanagement ist GEIL

Motivation wird
nachhaltig gestört

Stetiges Gefühl der
Überwachung

...



Regel 1



Regel 2



Regel 3



Regel 4



Regel 5



Regel 6



Regel 7



Regel 8



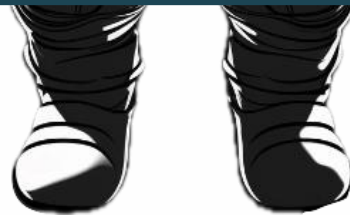
Regel 9



Regel 10



REGEL 7





REGEL 7

Für alle spannend, insbesondere für
POs, BAs, Anforderungsmanager



Requirements







Folgende Situation





“Ich weiß nicht, was ich da machen soll.“

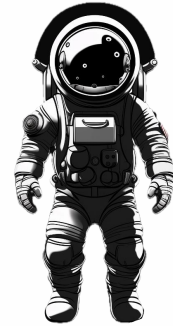




“Ich weiß nicht, was ich da machen soll.“

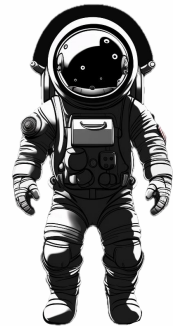
“Aber das steht doch da.“





“Ich weiß nicht, was ich da machen soll.“

“Aber das steht doch da.“



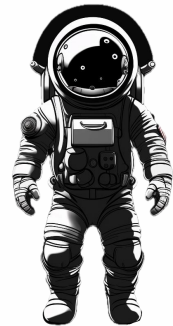
“Ich verstehe es aber nicht!“





“Ich weiß nicht, was ich da machen soll.“

“Aber das steht doch da.“



“Ich verstehe es aber nicht!“

“Dann bist du dumm!“



Schlechte Anforderungslage

Schlechte Anforderungslage

Verstrickt die Stakeholder/Anwender in Widersprüche

Schlechte Anforderungslage

Verstrickt die Stakeholder/Anwender in Widersprüche

Spielt die Stakeholder gegeneinander aus

Schlechte Anforderungslage

Verstrickt die Stakeholder/Anwender in Widersprüche

Spielt die Stakeholder gegeneinander aus

Lasst Teile der Anforderungen unter den Tisch fallen

Schlechte Anforderungslage

Verstrickt die Stakeholder/Anwender in Widersprüche

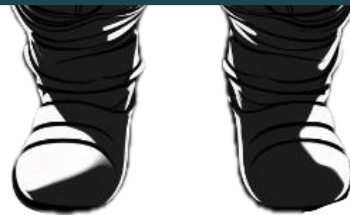
Spielt die Stakeholder gegeneinander aus

Lasst Teile der Anforderungen unter den Tisch fallen

Bringt Anforderungen in eine nicht-lesbare Form



REGEL 7





**Sorgt für eine
unklare
Anforderungslage**



Sorgt für eine unklare Anforderungslage

Unklares Ziel = viel
Interpretation



Sorgt für eine unklare Anforderungslage

Unklares Ziel = viel
Interpretation

Niemand kann wirklich
was falsch machen



Sorgt für eine unklare Anforderungslage

Unklares Ziel = viel
Interpretation

Niemand kann wirklich
was falsch machen

...



Regel 1



Regel 2



Regel 3



Regel 4



Regel 5



Regel 6



Regel 7



Regel 8



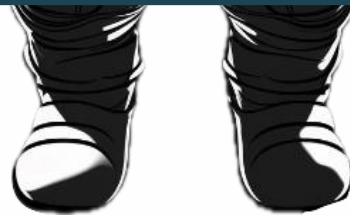
Regel 9



Regel 10

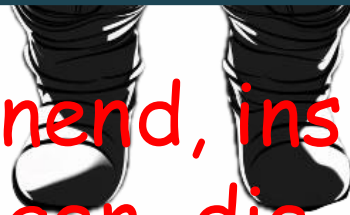


REGEL 8

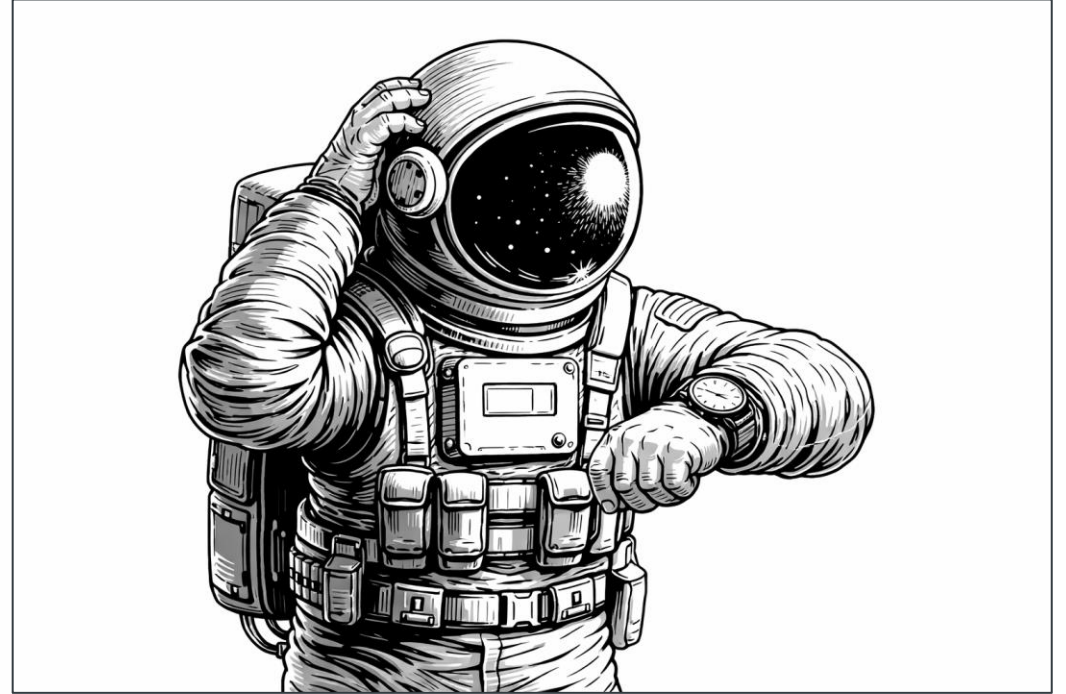




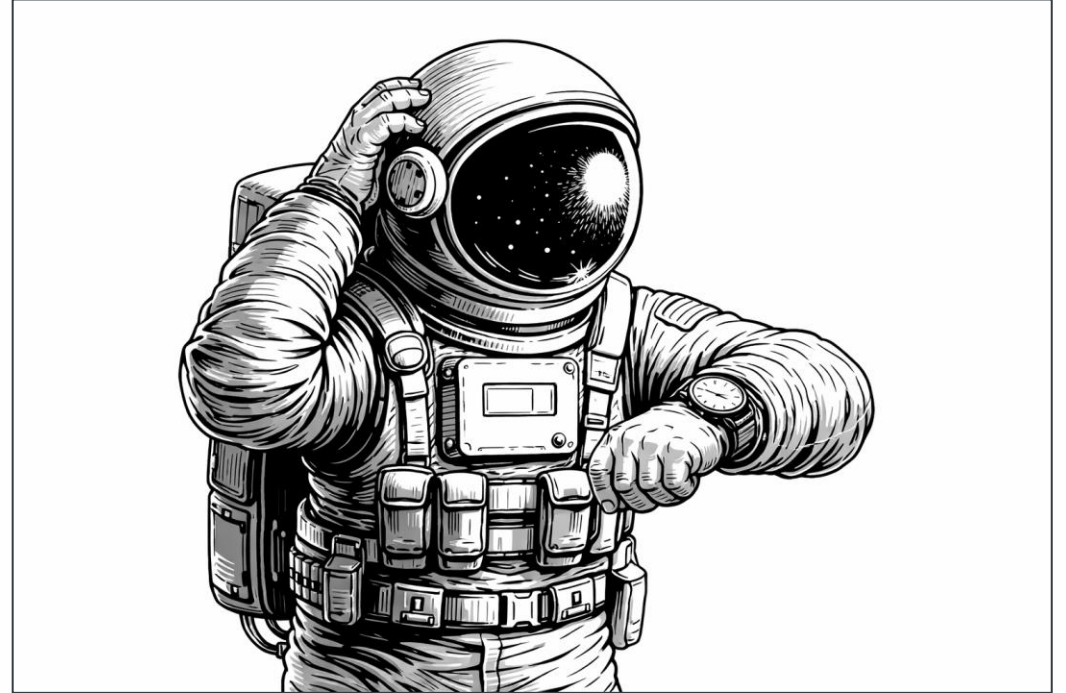
REGEL 8

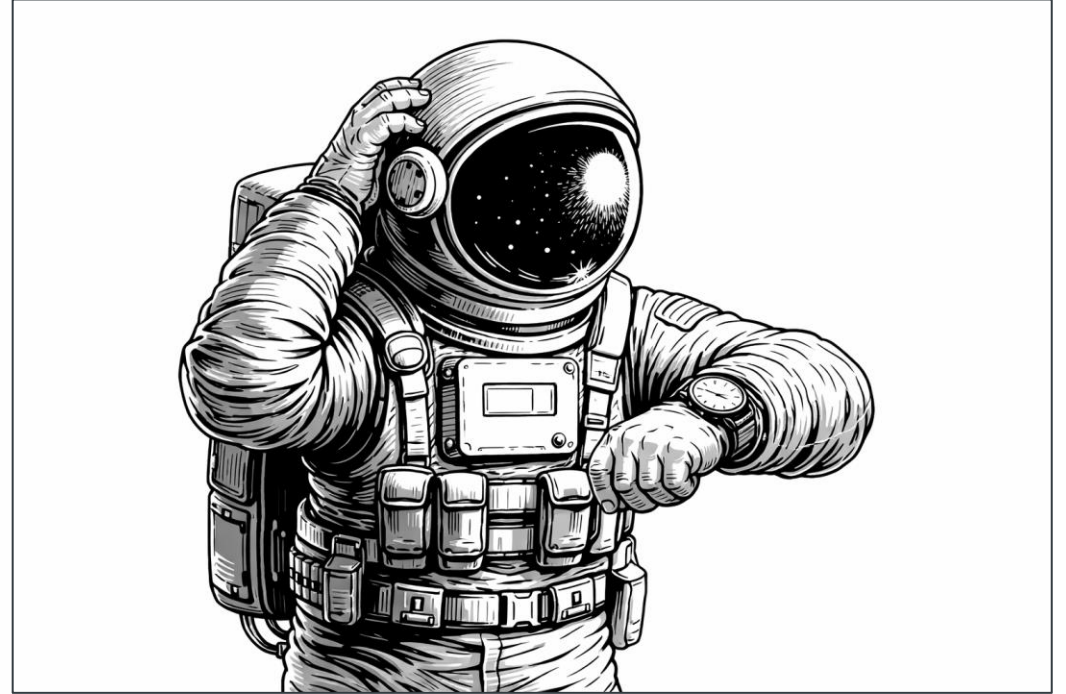


Für alle spannend, insbesondere für
diejenigen, die „führen“



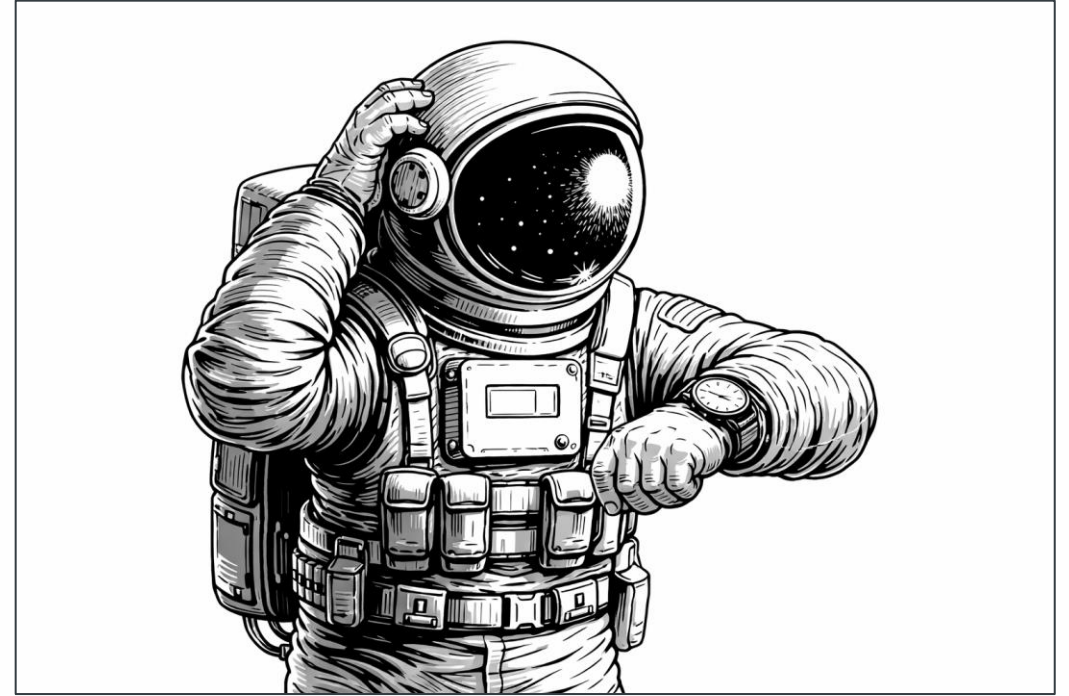
Personae







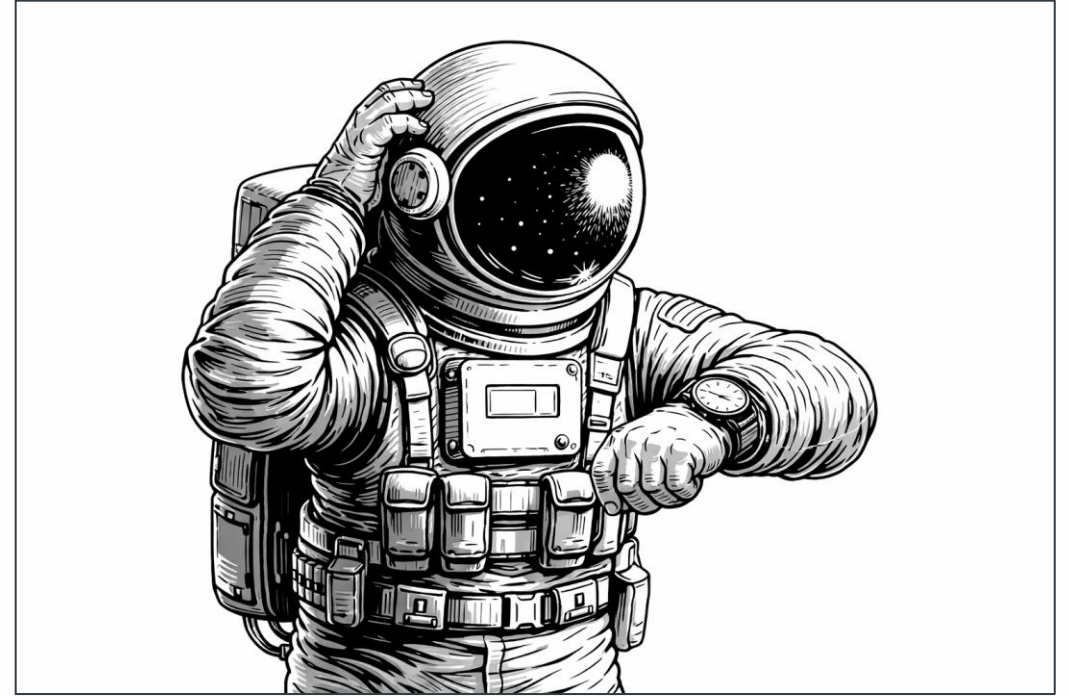
**Übermüdetes Deployment um drei Uhr
morgens**





**Übermüdetes Deployment um drei Uhr
morgens**

**Schlafentzug und immer ein Monster am
Start**

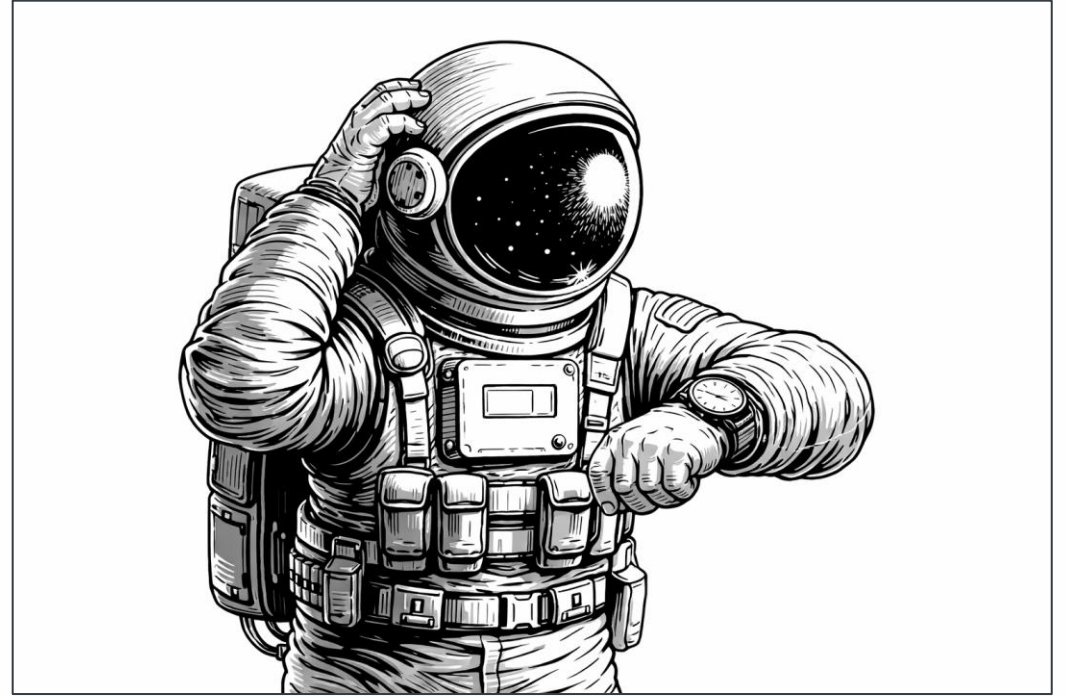




**Übermüdetes Deployment um drei Uhr
morgens**

**Schlafentzug und immer ein Monster am
Start**

Kurz vorm Burnout

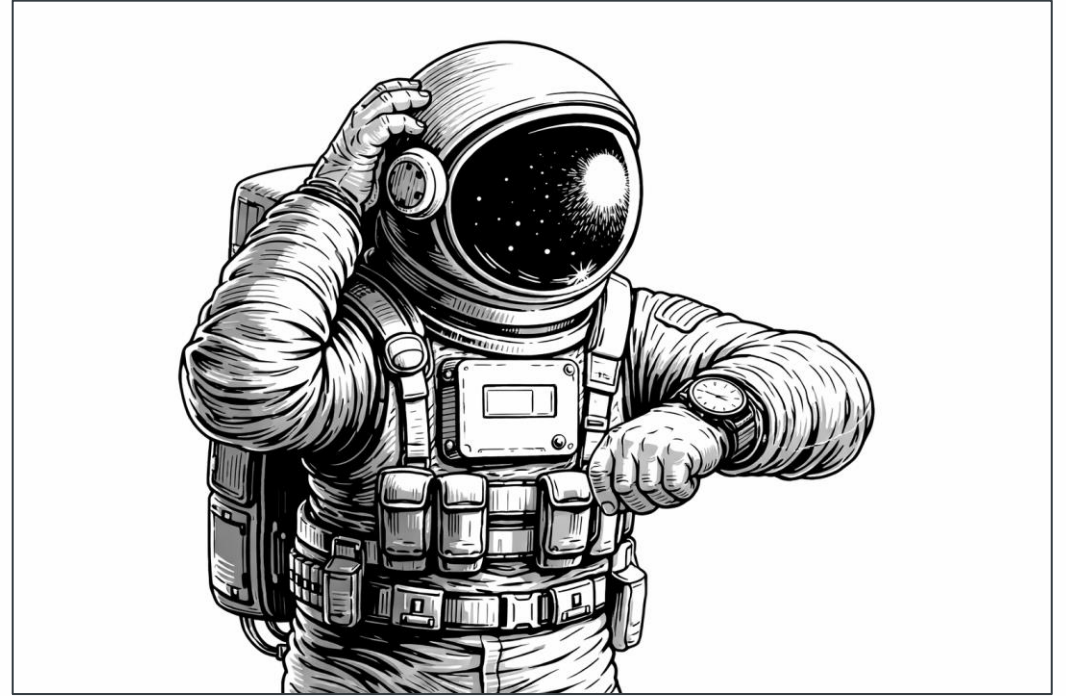




**Übermüdetes Deployment um drei Uhr
morgens**

**Schlafentzug und immer ein Monster am
Start**

Kurz vorm Burnout



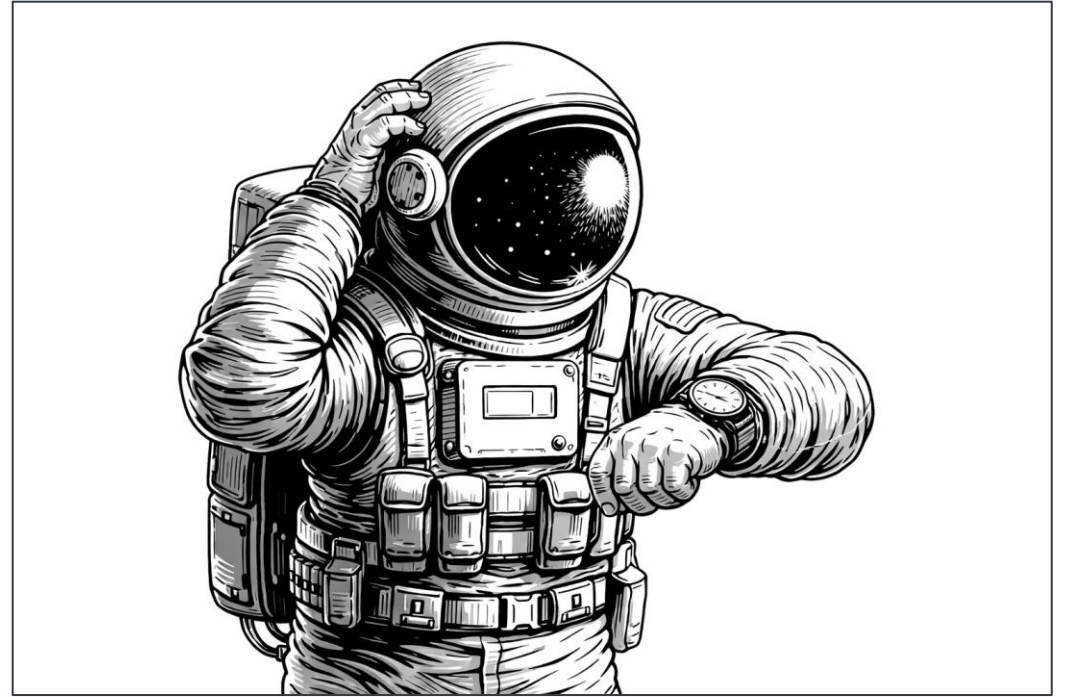
Hinweis auf nicht optimale Projektplanung



**Übermüdetes Deployment um drei Uhr
morgens**

**Schlafentzug und immer ein Monster am
Start**

Kurz vorm Burnout



Hinweis auf nicht optimale Projektplanung

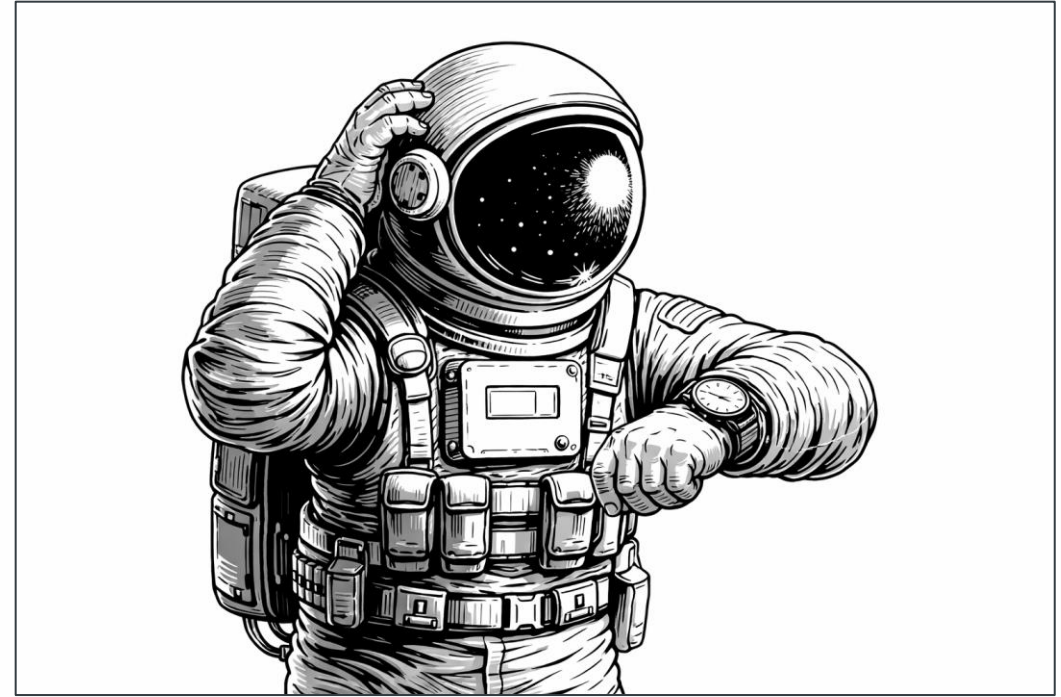
Pochen auf Prozesse



**Übermüdetes Deployment um drei Uhr
morgens**

**Schlafentzug und immer ein Monster am
Start**

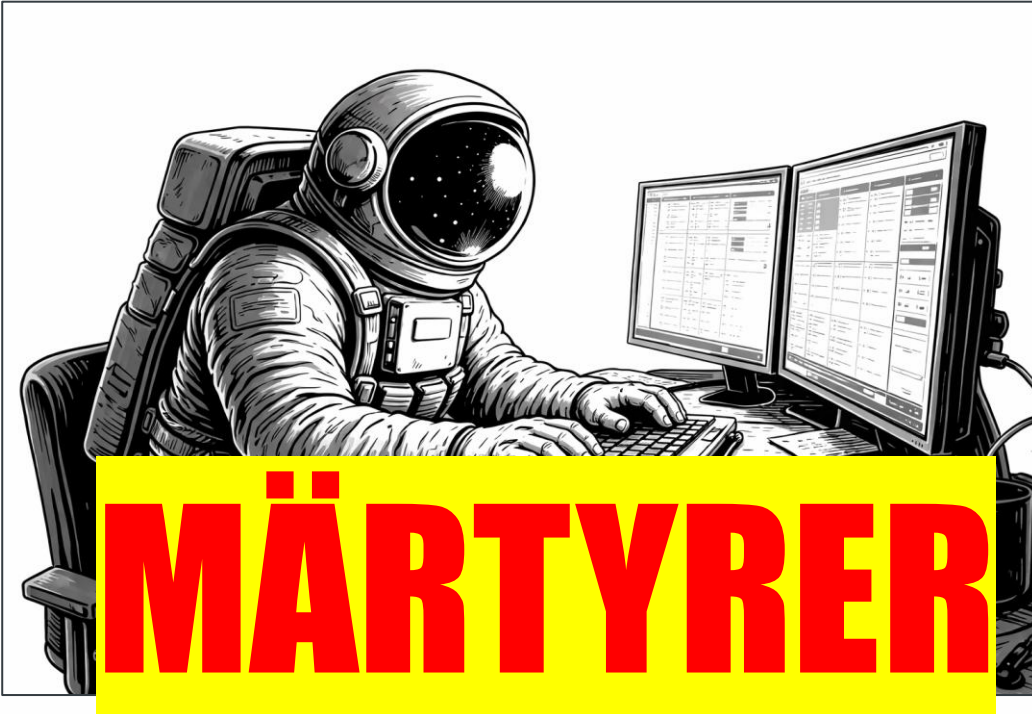
Kurz vorm Burnout



Hinweis auf nicht optimale Projektplanung

Pochen auf Prozesse

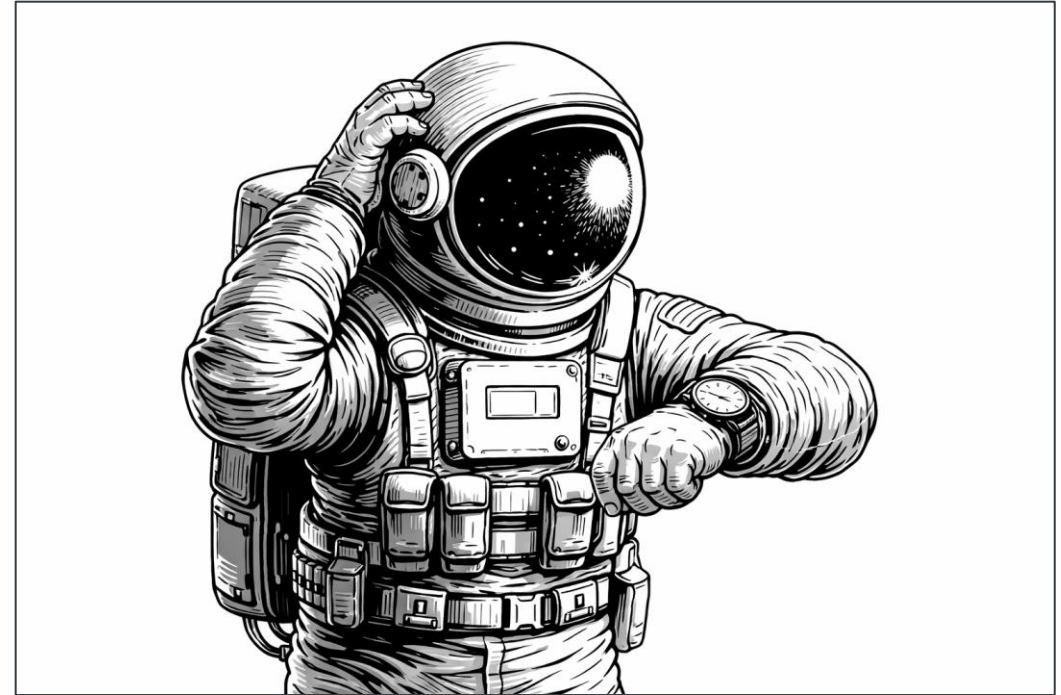
**Aufzeigen, wenn es für das Team zu stressig
wird**



Übermüdetes Deployment um drei Uhr
morgens

Schlafentzug und immer ein Monster am
Start

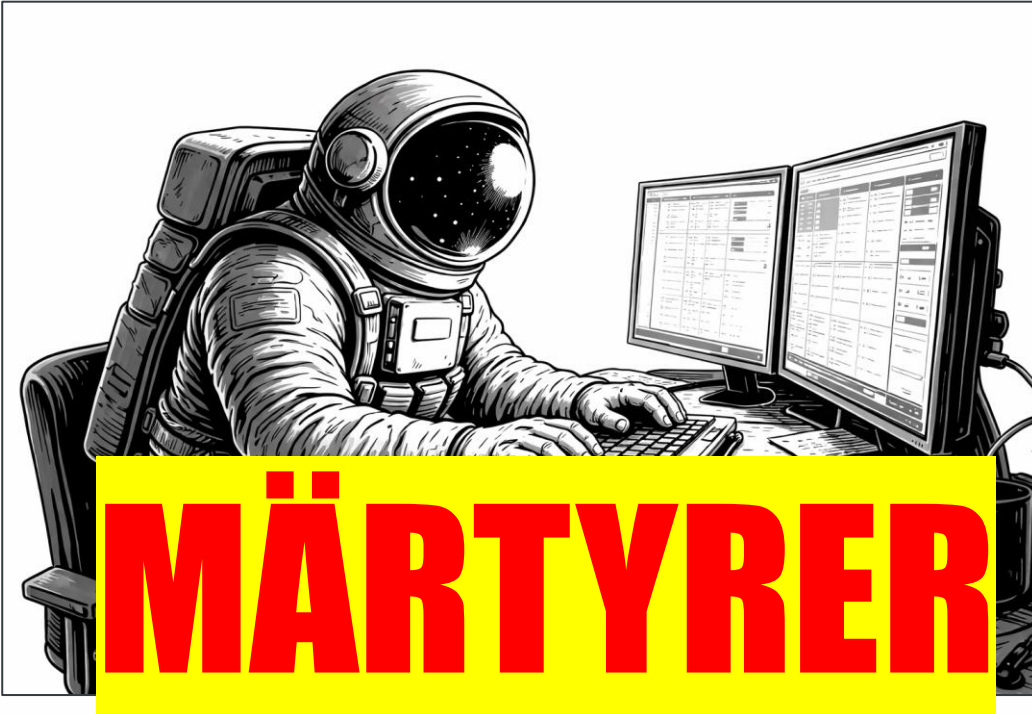
Kurz vorm Burnout



Hinweis auf nicht optimale Projektplanung

Pochen auf Prozesse

Aufzeigen, wenn es für das Team zu stressig
wird



Übermüdetes Deployment um drei Uhr
morgens

Schlafentzug und immer ein Monster am
Start

Kurz vorm Burnout



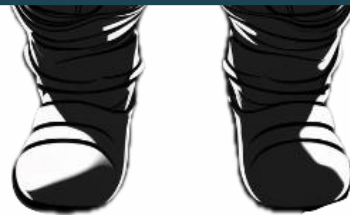
Hinweis auf nicht optimale Projektplanung

Pochen auf Prozesse

Aufzeigen, wenn es für das Team zu stressig
wird



REGEL 8





Belohnt Märtyrertum



Belohnt Märtyrertum

Alle verlassen sich auf
eine Person



Belohnt Märtyrertum

**Alle verlassen sich auf
eine Person**

Jeder ist ersetzbar



Belohnt Märtyrertum

Alle verlassen sich auf
eine Person

Jeder ist ersetzbar

...



Regel 1



Regel 2



Regel 3



Regel 4



Regel 5



Regel 6



Regel 7



Regel 8



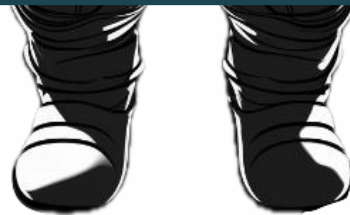
Regel 9



Regel 10



REGEL 9

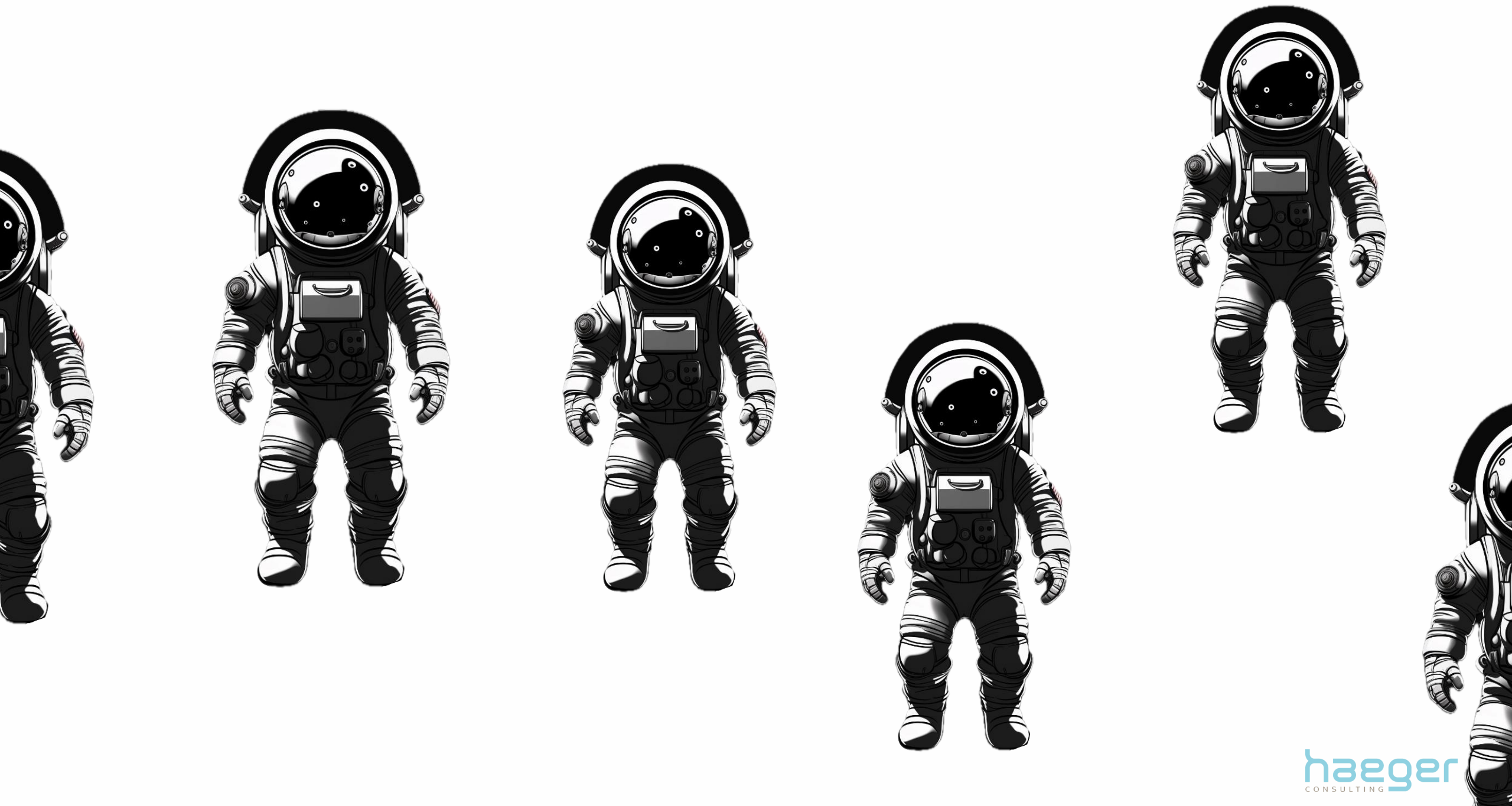


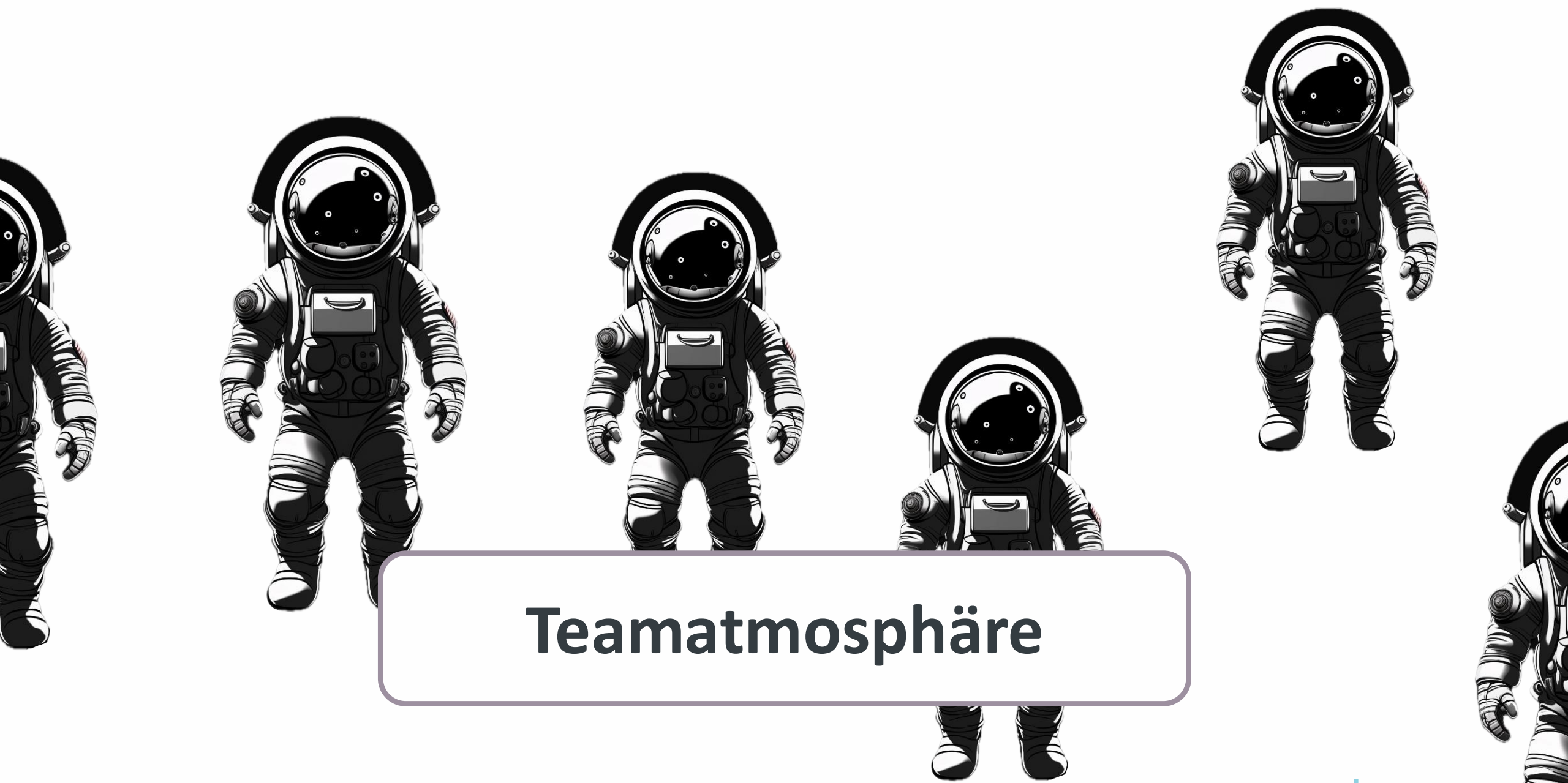


REGEL 9

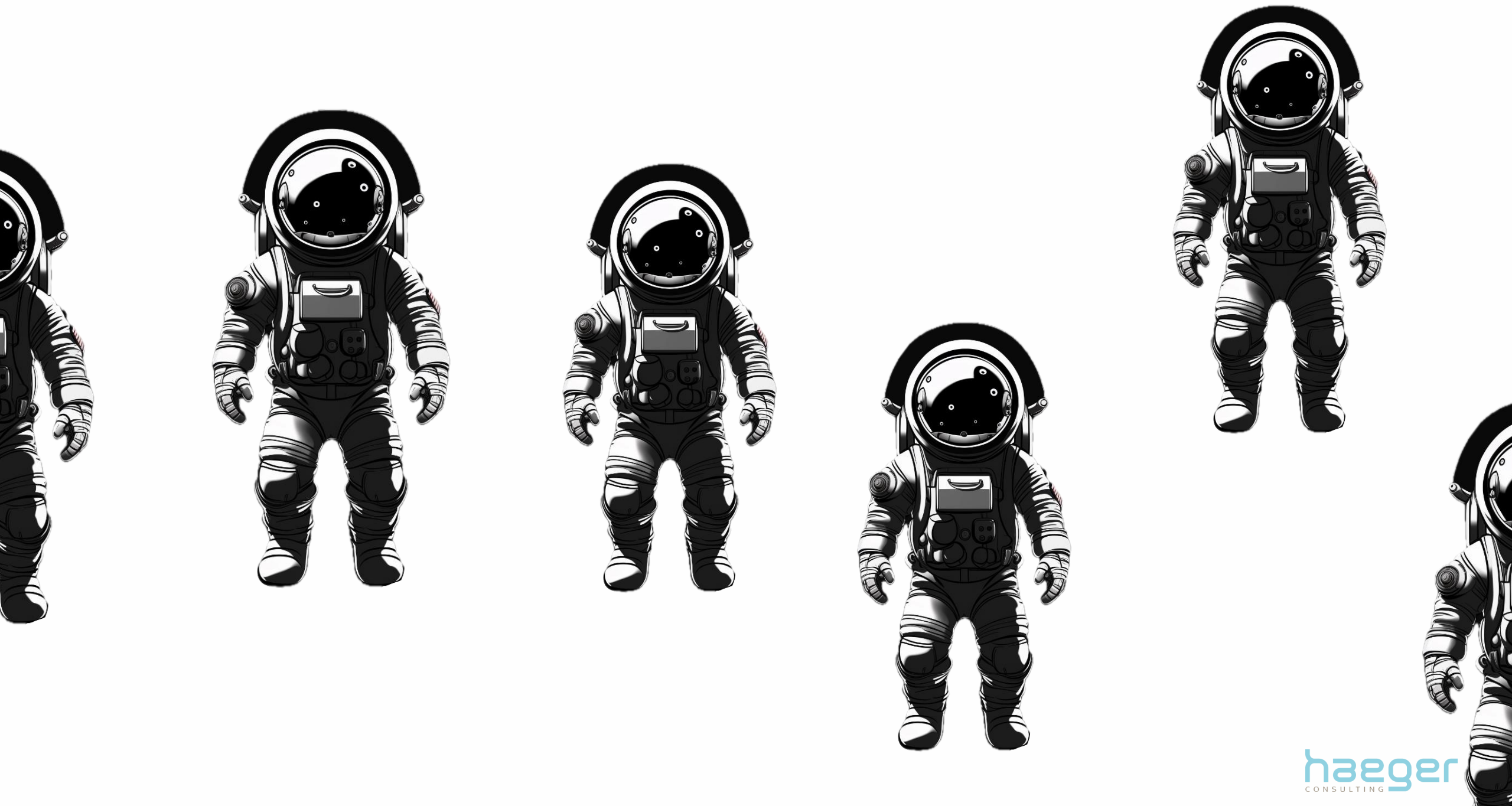
Für alle spannend







Teamatmosphäre







Folgende Situation

**„Ich will nicht mit
Albert
zusammenarbeiten!“**



**„Ich will nicht mit
Albert
zusammenarbeiten!“**

**„Der ist nur Tester, also
ein gescheiterter
Entwickler!“**



„Ich will nicht mit
Albert
zusammenarbeiten!“

„Von Tom winkt er alle
PRs durch, aber meine
werden immer
zurückgewiesen!“

„Der ist nur Tester, also
ein gescheiterter
Entwickler!“



„Ich will nicht mit
Albert
zusammenarbeiten!“

„Von Tom winkt er alle
PRs durch, aber meine
werden immer
zurückgewiesen!“

„Der ist nur Tester, also
ein gescheiterter
Entwickler!“

„Du hast doch keine
Ahnung!“



„Ich will nicht mit
Albert
zusammenarbeiten!“

„Von Tom winkt er alle
PRs durch, aber meine
werden immer
zurückgewiesen!“

„Ein Storypoint ist ein
Personentag! Was soll
der
Komplexitätsquatsch?“

„Der ist nur Tester, also
ein gescheiterter
Entwickler!“

„Du hast doch keine
Ahnung!“



„Ich will nicht mit
Albert
zusammenarbeiten!“

„Von Tom winkt er alle
PRs durch, aber meine
werden immer
zurückgewiesen!“

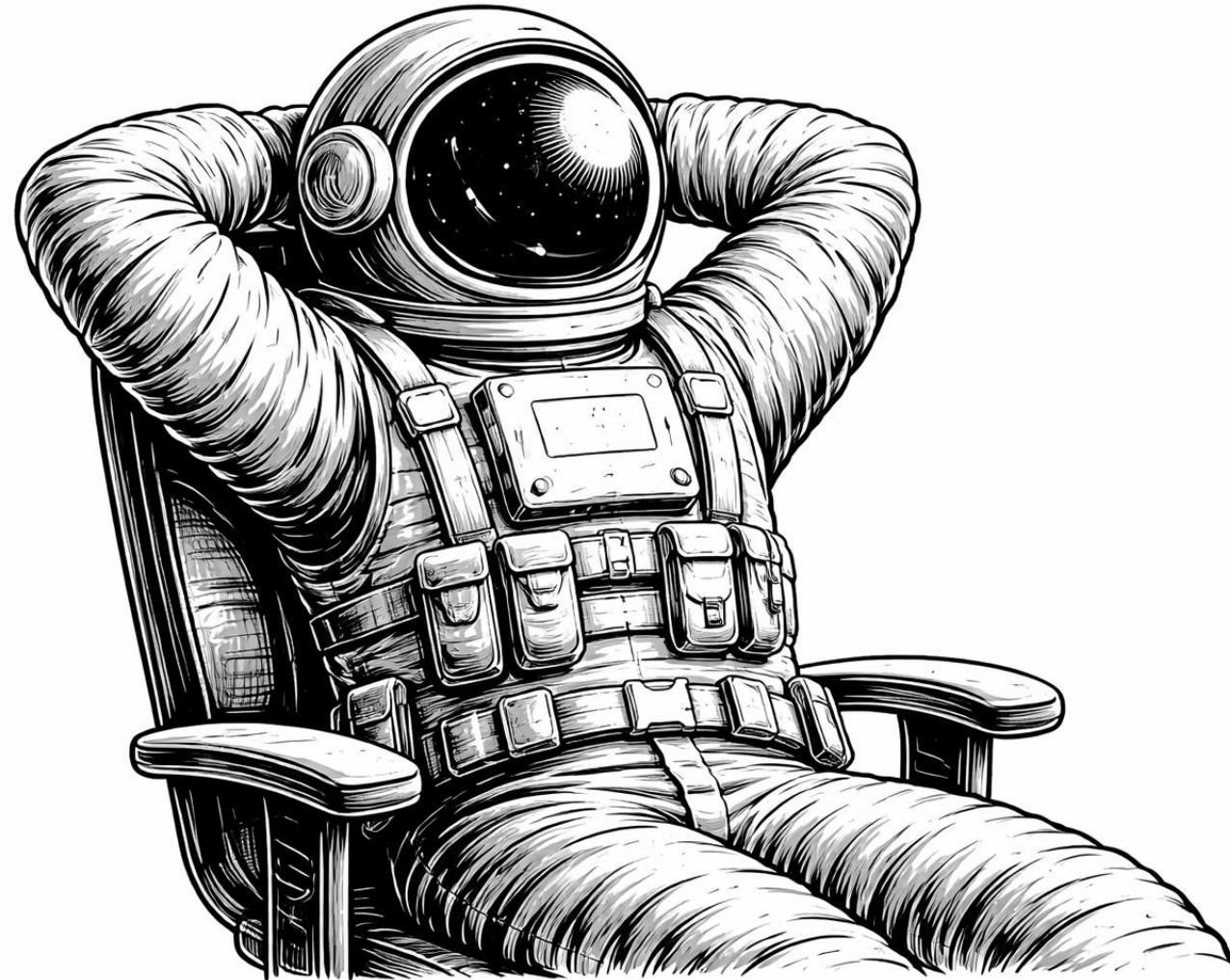
„Ein Storypoint ist ein
Personentag! Was soll
der
Komplexitätsquatsch?“

„Der ist nur Tester, also
ein gescheiterter
Entwickler!“

„Du hast doch keine
Ahnung!“

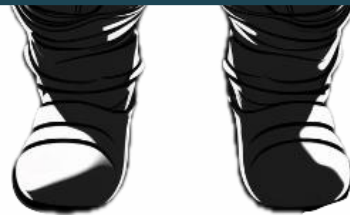
„Ich erzähle euch mal
was über die
Pharmaindustrie...“







REGEL 9





**Sorgt für eine
ungesunde
Teamatmosphäre**



Sorgt für eine ungesunde Teamatmosphäre

Motivation im
allgemeinen sinkt



Sorgt für eine ungesunde Teamatmosphäre

Motivation im
allgemeinen sinkt

Produktivität nimmt ab



Sorgt für eine ungesunde Teamatmosphäre

Motivation im
allgemeinen sinkt

Produktivität nimmt ab

...



Regel 1



Regel 2



Regel 3



Regel 4



Regel 5



Regel 6



Regel 7



Regel 8



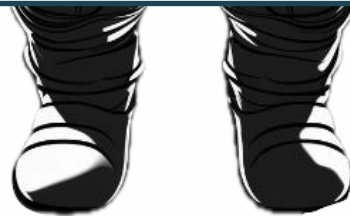
Regel 9



Regel 10



REGEL 10

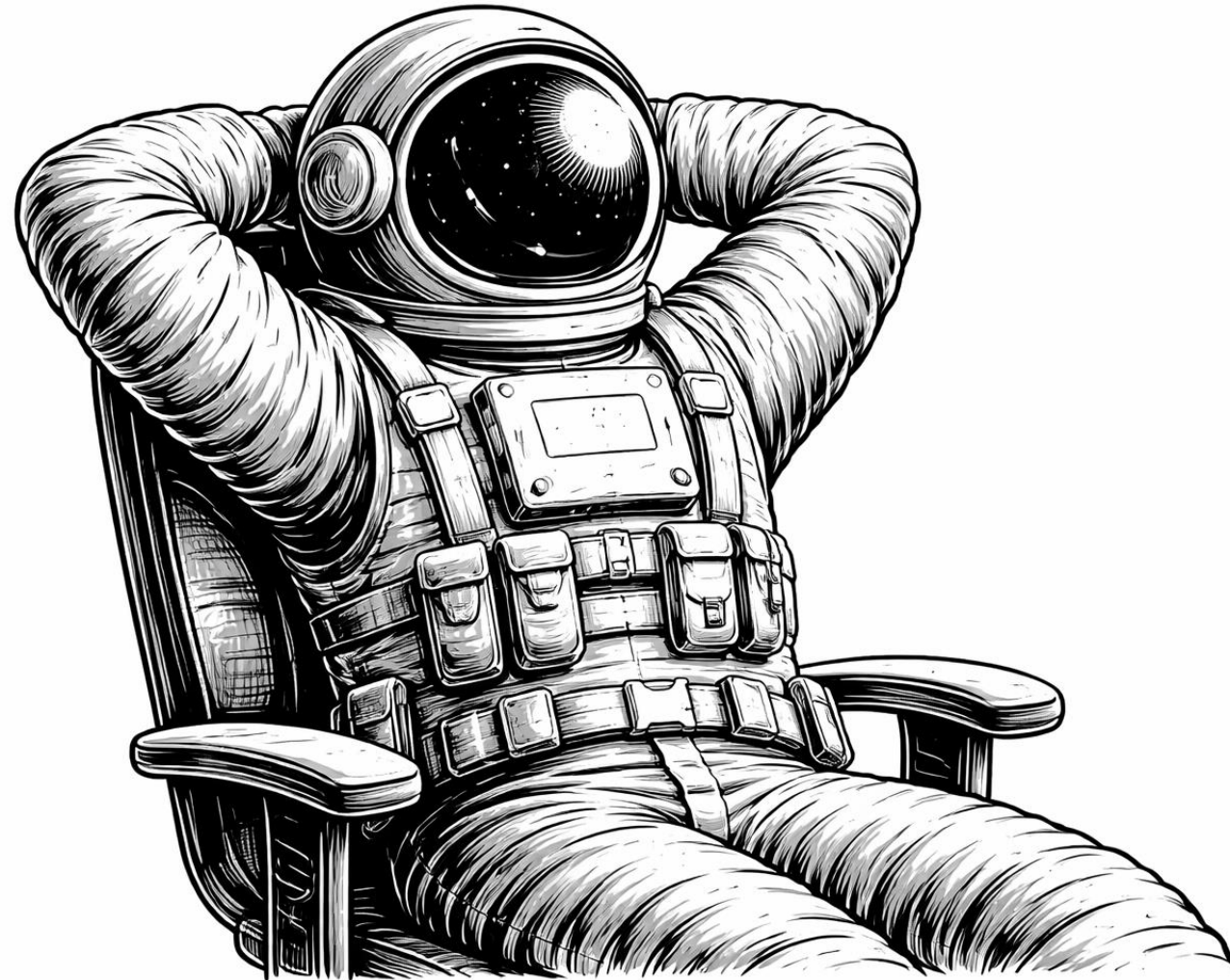




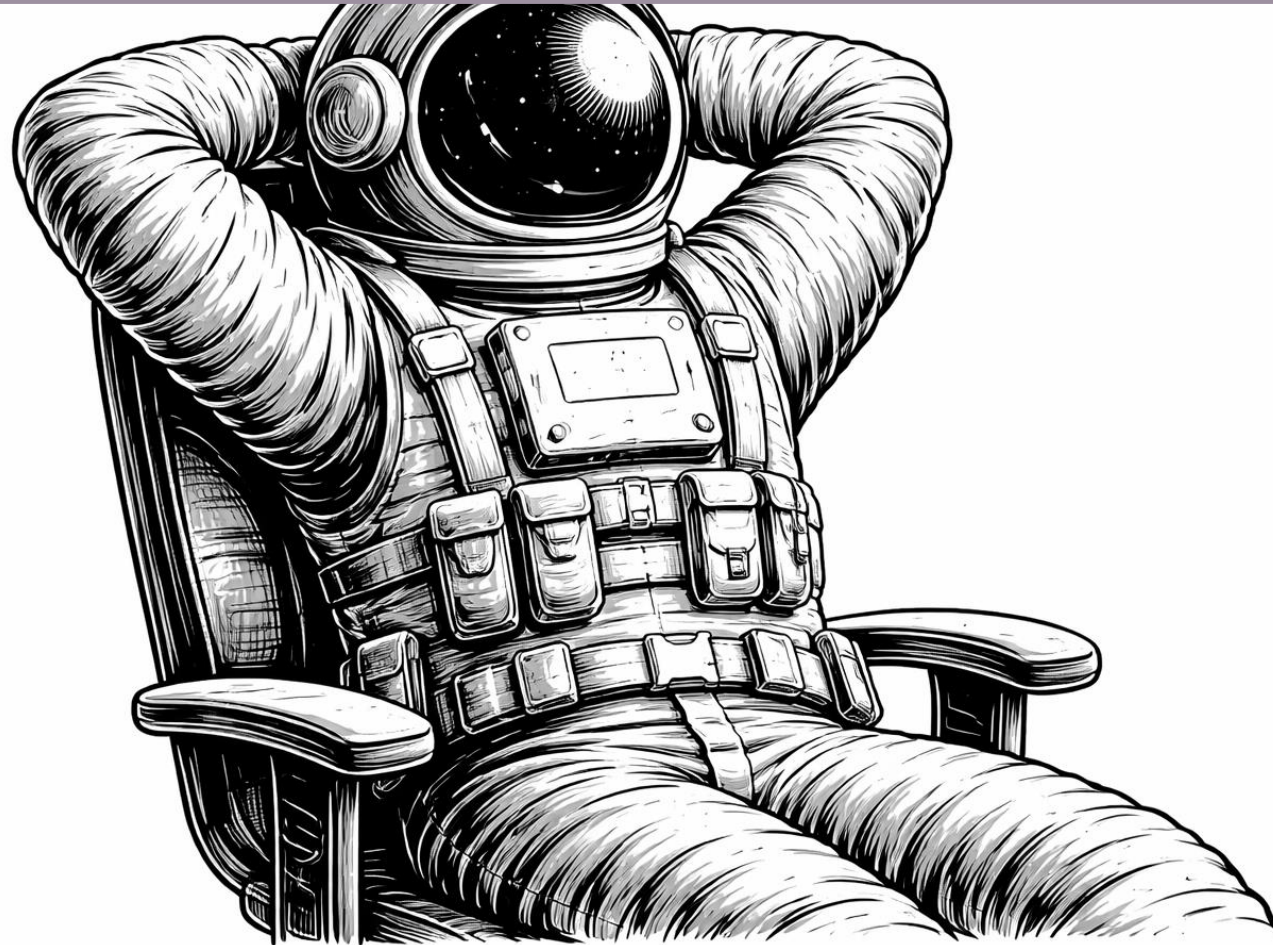
REGEL 10



Für alle spannend



Wendet alle Regeln konsequent an!



Wendet alle Regeln konsequent an!

Gefährdet euren Projekterfolg



Wendet alle Regeln konsequent an!

Gefährdet euren Projekterfolg

Demotiviert eure Teammitglieder



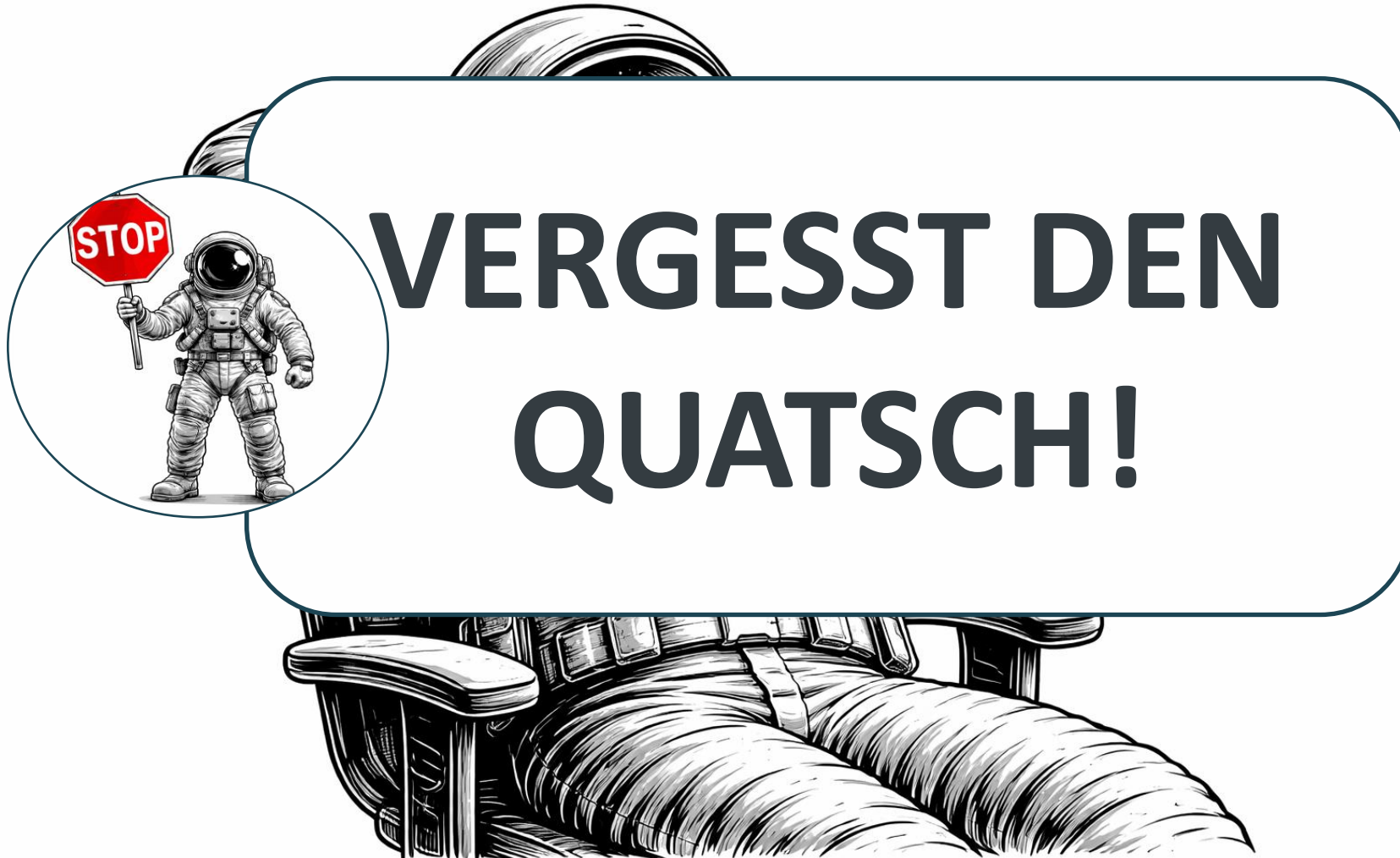
Wendet alle Regeln konsequent an!

Gefährdet euren Projekterfolg

Demotiviert eure Teammitglieder

Prägt sie für den Rest ihres (Berufs-)Lebens





Key Takeaways



Key Takeaways

Benennt den Elefanten im Raum



Key Takeaways

Benennt den Elefanten im Raum



Sucht euch Verbündete

Key Takeaways

Benennt den Elefanten im Raum

Beobachtet, Analysiert, Bespricht



Sucht euch Verbündete

Key Takeaways

Benennt den Elefanten im Raum

Beobachtet, Analysiert, Bespricht



Sucht euch Verbündete

Nicht alles lässt ändern

Key Takeaways

Benennt den Elefanten im Raum

Beobachtet, Analysiert, Bespricht

Sucht euch Verbündete

Nicht alles lässt ändern

Inspect & adapt!





Menti.com **7241 4215**



Den Einstieg maximal
schlecht gestalten



Feiert technische
Schulden



Dokumentiert schlecht



Nimmt jedes Bias mit



Be Agile



Micromanagement ist
GEIL



Sorgt für eine schlechte
Anforderungslage



Belohnt Märtyrertum



Sorgt für eine
ungesunde
Teamatmosphäre



Wendet alle Regeln an

Ausgabe 02/2026

Java aktuell

Praxis, Wissen, Networking. Das Magazin für Entwicklerinnen und Entwickler

1,99 CHF | A: 9,00 EUR | Benelux: 8,50 EUR

Java aktuell

JUG
member
java-aktuell.eu

Updates im Detail

JUnit 6, Maven 4

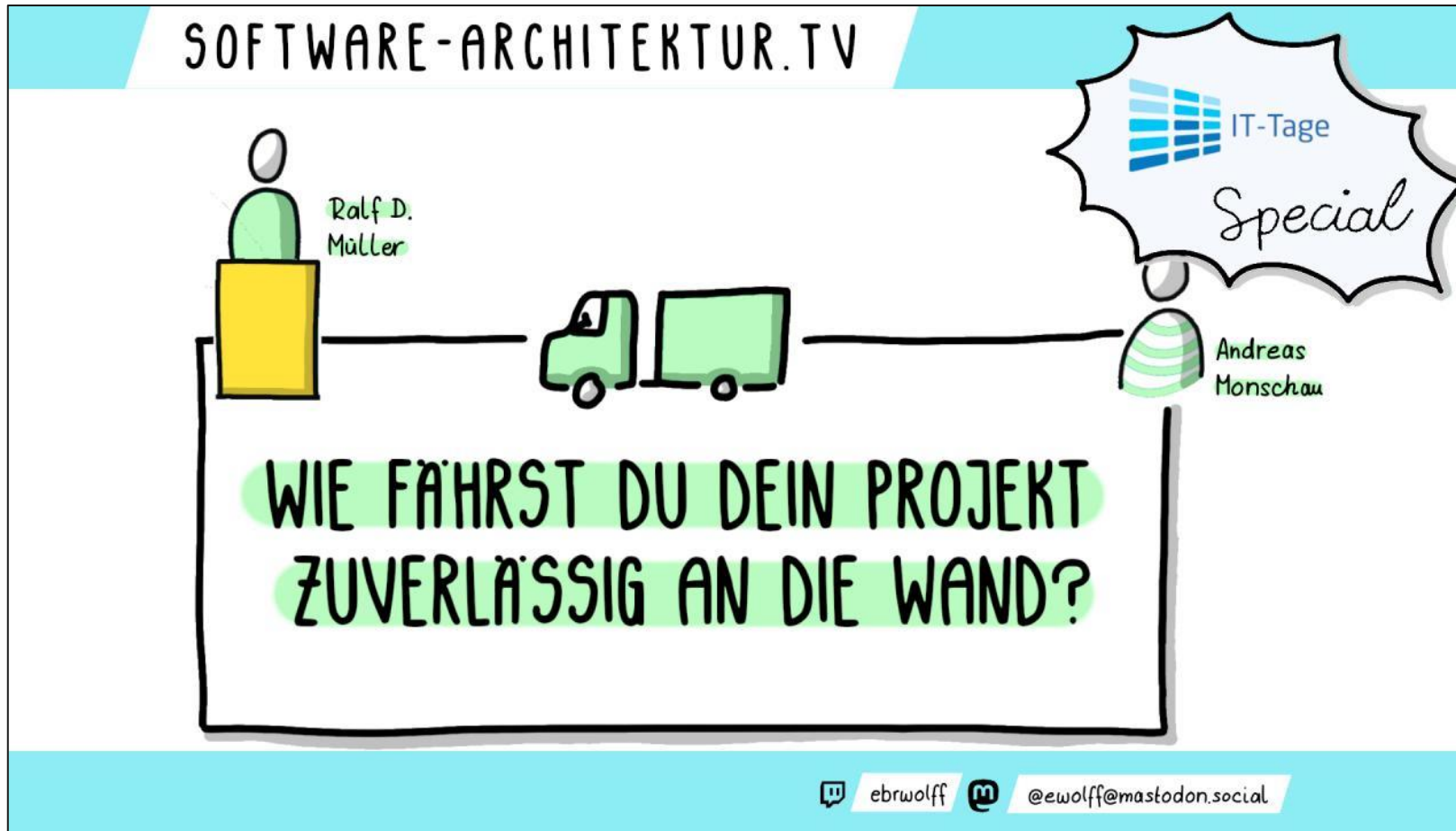
Fediverse

Die Alternative zu gängigen sozialen Netzwerken

Women in Tech

Status quo und Perspektiven





<https://www.youtube.com/watch?v=uirY-dxT8uQ>

Held oder Märtyrer – Willkommen im Projektchaos

Oft scheitern IT-Projekte an unklaren Anforderungen und unrealistischen Deadlines. Strategien zur Rettung können entscheidende Verbesserungen bringen.



(Bild: A.Basler/Shutterstock.com)

11.11.2025, 08:31 Uhr Lesezeit: 16 Min. Developer

Von Andreas Monschau

INHALTSVERZEICHNIS

Es könnte so schön sein: Als neues Teammitglied wendet man sich einem Softwareprojekt zu, alle wissen, was sie zu tun haben, die Anforderungen sind klar und verständlich und das Team beherrscht die Technik (und nicht umgekehrt). Gibt es doch mal Probleme, dann kümmern sich eigens dafür vorgesehene Rollen darum, und räumen die Schwierigkeiten beiseite. Die Organisation sorgt dafür, dass alle in Ruhe arbeiten können. Am Ende hat man dann eine Sache ganz gewiss erreicht: Anwender beziehungsweise Kunden erhalten ein Produkt mit echtem Mehrwert. Das klingt doch zu schön, um wahr zu sein!

Held oder Märtyrer – Willkommen im Projektchaos





Alles auf grün im Projekt?
VIER Gründe, warum nicht!

Java Magazin 5.26
windows .developer 5.26



<https://www.linkedin.com/in/andreas-monschau-544aa141/>

<https://www.andreasmonschau.de>

amonschau@haeger-consulting.de

VIELEN DANK!



Haeger Consulting GmbH
Brunnenallee 6
53177 Bonn

haeger
CONSULTING