



SEBRAS - Stripes over Sprints

Systems Engineering plus Agilität neu gedacht

ReConf 2025. Thaddäus Dorsch & Jürgen Rambo, München 05.-07. Mai 2025

GfSE →
Gesellschaft für
Systems Engineering e.V.
GERMAN CHAPTER OF INCOSE



1

Motivation

Motivation

Systems Engineering braucht mehr Agilität

Wir waren ...

traurig .. erschrocken .. enttäuscht ...

weil, ...

- immer noch so **viele Entwicklungsprojekte** ihre **Ziele verfehlen** (zu wenig Zeit / Qualität & zu hohe Kosten)
- **agile Transformationen oft gescheitert** sind
- **Agilität an Bedeutung verliert**
- **skalierten agile Frameworks oft nicht wirklich** Besserung bringen
- **hierarchische Strukturen, Silos und Matrix-Organisationen** vorherrschen
- **Agilität inzwischen als "verbrannt"** gesehen wird
- **(modellbasiertes) Systems Engineering** oft nur in **Nischen** zu finden ist
-



... aber dann kam das ...

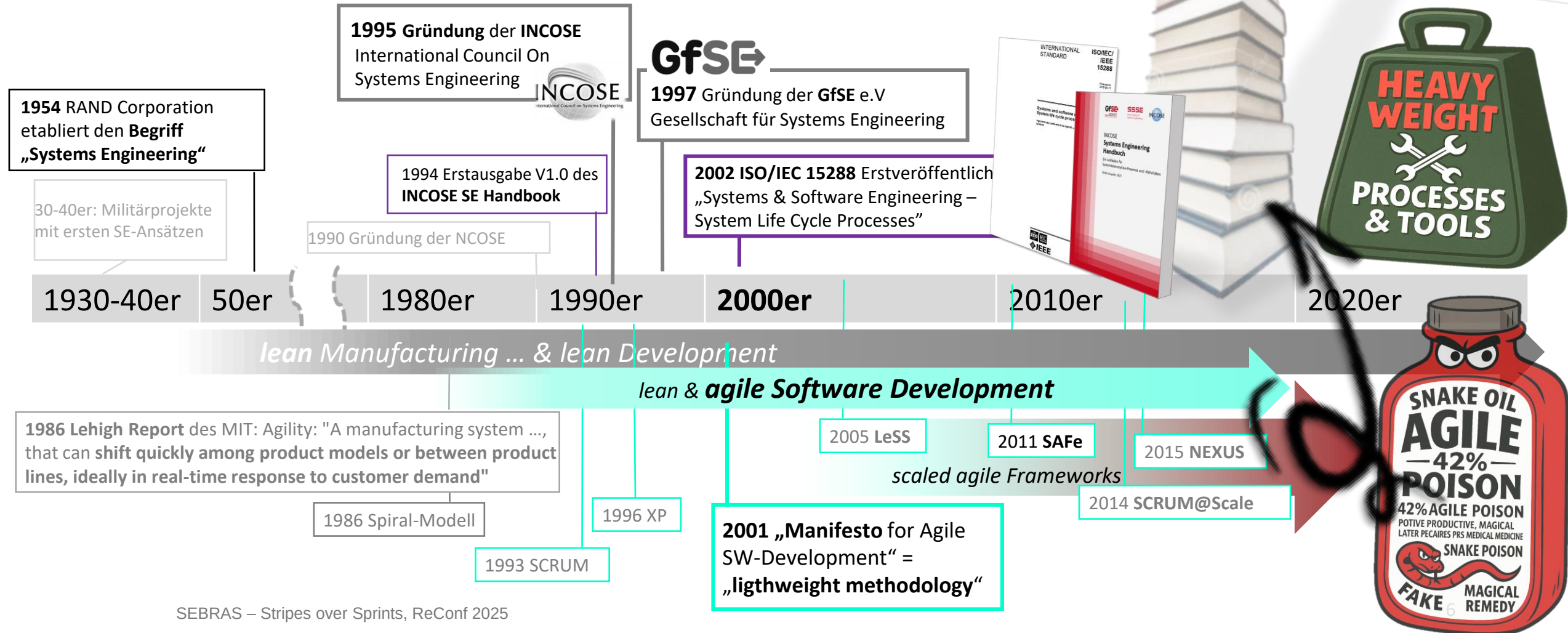


2

Das Problem

Die Entwicklung von technischer Entwicklung ...

Agile wird zu oft als Gegensatz zu Systems Engineering verstanden ...



Fake Agile, Dark Agile, Snake Oil Agile ...

Agilität ist kein Selbstzweck!

***"Das Wort 'agil' wurde bis zu einem Punkt ausgehöhlt,
an dem es praktisch bedeutungslos ist - ..."***

Dave Thomas, 2014

Mitverfasser des Agile Manifestos von 2001



Bild in Anlehnung an Andrè Baken



Agile & Systems Engineering

werden zu oft isoliert umgesetzt und falsch verstanden



■ Systems Engineering

- Fehlinterpretation: SE als immer passende „**Best**“ Practice und **Wasserfall** ohne Iteration verstanden
- zu viele Prozesse und **Modelle aus dem Elfenbeinturm** und **schwergewichtige Tools**
- (zu) starker **Fokus auf Technik** zu wenig "Organisationentwicklung & Kultur"

■ Agile Development

- **zwanghafter Glaube an Frameworks** führt zu „agilen Ansätzen“, die am Ende selbst wenig agil sind
- aus dem "lightweight development" werden **schwergewichtige Prozesse** mit vielen Rollen (z.B. SAFe)

ABER: unsere Entwicklung braucht beides!

Leichtgewichtiges Systems Engineering **UND** richtige Agilität!

Wir machen

Systems Engineering besser (mit) agil!



Struktur im Systems Engineering

& Kultur im Agilen Arbeiten!

Strukturen

als Fundament, Rahmen,
Bausteine, ...
womit entwickelt wird.

- „nur“ **kompliziert**
- planbar & messbar
- standardisierbar

Kultur

als Verhalten, Werte,
Interaktionen, ...
wie tatsächlich entwickelt
wird.

- **komplex**
- dynamisch
- emergent und
nur „beobachtbar“



Beide sind wichtig und **beeinflussen** sich wechselseitig!

„Kultur ist in der Lage,
Irrsinn unsichtbar zu machen,
und das Irrsinnige logisch
erscheinen zu lassen.“
Niels Pflaeging



SEBRAS

Systems
Engineering for
Business
Reactivity and
Agility with
Stripes

"Ein Denkmodell "

3

Die Arbeitsgruppe **AG agileSE** der GfSE

Arbeitsgruppe **Agilität im Systems Engineering**

Gesellschaft für Systems Engineering e.V.



- erneute Gründung 2018
- **“Besser agil mit Systems Engineering”**
- Prinzipien, Praktiken, Methoden, Frameworks ...
und zugrundeliegenden Muster, die das Beste aus
Agilität & Systems Engineering zusammenbringen
- 10-20 wechselnd **aktive Mitstreiter*innen**
- 2-3 **Workshops/Jahr** (vor Ort & online)
und **monatlicher Austausch** (online)
- Präsentationen, Veröffentlichungen
und weitere Infos unter:

<https://www.gfse.org/de/arbeitsgruppe-agilese.html>



SEBRAS Gründungstreffen, Hamburg 2024, mit (v.l.n.r.):
Thaddäus Dorsch, Tim Weilkiens, Paul Dahlke, Jürgen Rambo
Fotograph: Marty

4

Auf dem Weg zu den SEBRAS

Ein Leitgedanke, 4 Werte & 12 Prinzipien

Der Kern des „agile Manifesto“ von 2001



Ein Leitgedanke, 4 Werte & 12 Prinzipien

Der Kern des „agile manifesto“ von 2001

Wir erschließen **bessere Wege, Lösungen zu entwickeln,**
indem wir es selbst tun und anderen dabei helfen.

Durch diese Tätigkeit haben wir diese **Werte** zu schätzen gelernt:

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

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

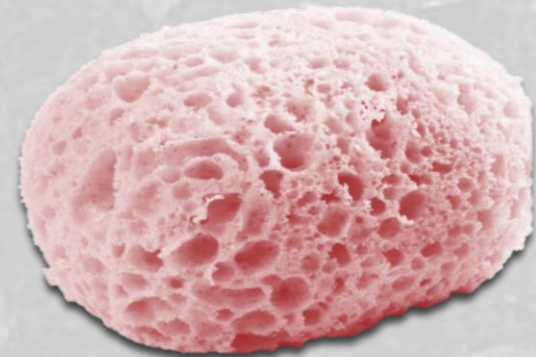
Das „Agile Manifesto“ ist schwammig, anarchisch, ... und vor allem **kultur- & menschenzentriert**

*Zitat: „ ... im Kern geht es [...] um „schwammige“ Aussagen –
um das Liefern guter Produkte an Kunden, in einem Arbeitsumfeld, das mehr tut, als
nur von "Menschen als unserem wichtigsten Gut/Kapital" zu reden.*

*Sondern **wir „handeln“ tatsächlich so,
dass die Menschen das Wichtigste sind
und nicht das Wort „Güter/Kapital“.***

*Letztendlich geht es also [...] um den ...
schwammigen Inhalt von Werten und Kultur“*

frei übersetzt aus <https://agilemanifesto.org/history.html>



Der lesenswerte Geist der 17 Organisationsanarchisten des Agile Manifestos ...

Vom 11. bis 13. Februar 2001 trafen sich 17 Personen in der Lodge des Snowbird-Skigebietes in den Wasatch Mountains in Utah, um zu reden, Ski zu fahren, zu entspannen und natürlich zu essen - und dabei eine gemeinsame Basis zu finden. Was dabei herauskam, war das „Manifest für Agile Softwareentwicklungs“.

Vertreter von Extreme Programming, SCRUM, DSDM, Adaptive Software Development, Crystal, Feature-Driven Development, Pragmatic Programming und andere, die sich für **Alternativen zu den dokumentations-getriebenen, schwergewichtigen Software-entwicklungsprozessen** einsetzten, kamen zusammen. Ein größeres Zusammentreffen von **Organisationsanarchisten** ist schwer zu finden, so dass das, was aus diesem Treffen hervorging, symbolisch war - ein Manifest für Agile Softwareentwicklung - das von allen Teilnehmern unterzeichnet wurde. [...]

Wir nennen uns "The Agile Alliance", eine **Gruppe unabhängiger Denker** über Softwareentwicklung - und manchmal auch Konkurrenten untereinander – wir einigten uns auf das Manifest für Agile Softwareentwicklung, das auf dieser Website veröffentlicht wird.

Während das Manifest einige spezifische Ideen liefert, existiert aber auch ein tiefgreifenderes Thema, das viele, aber nicht alle Mitglieder der Allianz antreibt. Am Ende des zweitägigen Workshops scherzte Bob Martin, dass er im Begriff sei, eine vielleicht „etwas schwammige“ Aussage zu machen. Während diese Aussage von Humor gefärbt war, stimmten jedoch nur wenige nicht mit Bobs Einschätzung überein - dass wir uns alle **privilegiert fühlten, mit einer Gruppe von Menschen zusammenzuarbeiten, die eine Reihe kompatibler Werte besitzen. Werte, die auf Vertrauen und Respekt füreinander basieren und die Organisationsmodelle fördern, die auf Menschen und Zusammenarbeit basieren** und die Arten von Organisation schaffen, in denen wir alle gerne arbeiten möchten.

Im Kern geht es bei den Agile Methodikern meiner Meinung nach wirklich um „schwammige“ Aussagen - um das Ausliefern guter Produkte an ihre Kunden, indem sie in einem Umfeld arbeiten, das mehr tut, als von "Menschen als unserem wichtigsten Kapital" zu sprechen. Sondern wir handeln tatsächlich so, dass die Menschen das Wichtigste sind und nicht das Wort „Güter/Kapital“.

Letztendlich geht es also bei dem rasanten Anstieg des Interesses - und manchmal auch der heftigen Kritik an Agile Methoden - um den **schwammigen Inhalt von Werten und Kultur**.

Zum Beispiel darum, dass Extreme Programming letztendlich an Bedeutung und Interesse gewonnen hat, nicht wegen der Paarprogrammierung oder des Refactorings, sondern weil die damit verbundenen Praktiken insgesamt **eine Entwicklergemeinschaft beschreiben, die von der Last der „dilberttesken“ Zusammenarbeit befreit ist**.

Kent Beck erzählt die Geschichte eines frühen Auftrags, bei dem er einen Programmieraufwand von sechs Wochen für zwei Personen veranschlagte. Nachdem sein Vorgesetzter den anderen Programmierer zu Beginn des Projekts versetzte, schloss er das Projekt schließlich in zwölf Wochen ab - und fühlte sich dabei schrecklich! Der Chef schimpfte natürlich über Kent, wie langsam er in den zweiten sechs Wochen war. Kent, war etwas niedergeschlagen, weil er als Programmierer so ein "Versager" war. Er erkannte aber schließlich, dass seine ursprüngliche Schätzung von 6 Wochen extrem genau war - für zwei Leute - und dass sein "Versagen" in Wirklichkeit das Versagen des Managers war. Ja, das Versagen der standardmäßigen "fixen" Prozessmentalität, die unsere Branche so häufig plagt.

Diese Situationen gibt es jeden Tag bei Marketing oder Management, oder externe Kunden, interne Kunden und sogar Entwickler: **es will keiner harte Entscheidungen treffen. So werden irrationale Forderungen auf Basis kollektiven Machtstrukturen durchgesetzt**. Das ist kein reines Problem der SW-Entwicklung, sondern diese Probleme gibt es in allen „dilberttesken“ Organisationen.

Um in der New Economy erfolgreich zu sein, um aggressiv in die Ära des E-Business, des E-Commerce und des Internets vorzudringen, **müssen sich Unternehmen von ihren Dilbert'schen Manifestationen, von Machtspielen und obskuren Richtlinien befreien**. Diese Freiheit von den Unsinnigkeiten des Unternehmensalltags zieht die Befürworter der Agilen Methodik an und jagt den Traditionalisten eine „Heidenangst“ ein (man darf das Wort "Scheiße" in einer Fachartikel ja leider nicht verwenden).

Offen gesagt machen die agilen Ansätze den Unternehmensbürokraten Angst, weil ihnen die Verstecke ausgehen - zumindest denjenigen, die gerne Prozesse um des Prozesses willen vorantreiben, anstatt zu versuchen, das Beste für den "Kunden" zu tun und etwas pünktlich und greifbar und "wie versprochen" zu liefern.

Die agile Bewegung ist nicht generell gegen Methodologie oder Methodik. Tatsächlich wollen viele von uns **dem Wort Methodik wieder seine Glaubwürdigkeit verleihen**. Wir wollen ein Gleichgewicht wiederherstellen.

Wir begrüßen die Modellierung, aber nicht, um irgendwelche Diagramme in einem staubigen Repository abzulegen.
Wir begrüßen Dokumentation, aber nicht Hunderte von Seiten nie gepflegter und selten genutzter Wälzer.
Wir planen, erkennen aber auch die Grenzen der Planung in einer turbulenten Umgebung.

Diejenigen, die die Befürworter von XP oder SCRUM oder einer der anderen Agilen Methodiken als "Hacker" brandmarken würden, kennen wohl weder die Methodiken noch die ursprüngliche Definition des Begriffs "Hacker".

Das Treffen in Snowbird wurde bei einem früheren Treffen von Befürwortern des Extreme Programmings und einigen „anderen“ ins Leben gerufen, das von Kent Beck in der Rogue River Lodge in Oregon im Frühjahr 2000 organisiert wurde. Bei dem Treffen in Rogue River äußerten die Teilnehmer zwar ihre Unterstützung für eine Vielzahl an

„Leichtgewichtigen“-Methoden (engl: lightweight, light), aber daraus passierte weiter nichts Formelles.

Im Laufe des Jahres 2000 wurde eine Reihe von Artikeln geschrieben, die sich auf die Kategorie der "Light"- oder "Lightweight"-Methoden bezogen. Eine Reihe dieser Artikel bezog sich auf „Light“-Methoden, wie Extreme Programming, Adaptive Software Development, Crystal und SCRUM. In Gesprächen mochte niemand wirklich den Begriff "Light", aber er schien vorerst stehen zu bleiben.

Im September 2000 brachte Bob Martin von Object Mentor in Chicago den Ball des nächsten Treffens mit einer E-Mail ins Rollen: "Ich würde gerne eine kleine (zweitägige) Konferenz im Zeitrahmen von Januar bis Februar 2001 hier in Chicago einberufen. Der Zweck dieser Konferenz ist es, alle führenden Experten der „Light“-Methoden zusammenzubringen. Sie alle sind eingeladen; und es würde mich interessieren, wen ich noch ansprechen sollte." Bob richtete eine Wiki-Seite ein und die Diskussionen tobten. Schon früh meldete sich Alistair Cockburn mit einem Beitrag zu Wort, der die allgemeine Verstimmung über das Wort "Light" aufzeigte: "Ich habe nichts dagegen, dass die Methodik als 'leichtgewichtig' bezeichnet wird, aber ich bin mir nicht sicher, ob ich als „Leichtgewicht“ bezeichnet werden möchte, das an einem Treffen von „Leichtgewichtsmethodikern“ teilnimmt. Das klingt irgendwie nach einem Haufen dünner, schwachsinniger Leichtgewichte, die versuchen, sich zu erinnern, welcher Tag heute ist.

[...]

Wir hoffen, dass unsere gemeinsame Arbeit als Agile Alliance anderen hilft, über Softwareentwicklung, Methoden und Organisationen auf neue - agilere – Art und Weise nachzudenken. Wenn dem so ist, haben wir unsere Ziele erreicht.

eigene Übersetzung aus
<https://agilemanifesto.org/history.html>

5

SEBRAS in a Nutshell

Entwicklung ist wie ein **Zebra**

Jede ist einzigartig, aber sie verläuft nach ähnlichen Mustern!

Augen & Ohren stets
offen halten
und mit **Instinkt** und
dem richtigen **Riecher**
vorgehen ...

... auch wenn Dir manchmal
die **Haare zu Berge** stehen, ...

... wenn die **Hochs & Tiefs**
im Projekt kommen werden
...

Alle sind verschieden, aber
alle haben **ähnliche Muster!**

→ Wir empfehlen dafür **Stripes!**

... am Ende sind
vielen Optionen
möglich, und ...



... man muss **aufpassen**,
dass kein Sch... dabei
rauskommt ;-)

Grundmuster aus Streifen

Wir unterscheiden vier.

*Head
Stripe*

*Dorsal
Stripe*

*Tail
Stripe*

*Pattern-
Stripes*



Muster von Stripes erkennen und nutzen ...

Mit Fokus auf Bewährtes, z.B. aus SCRUM-Sprints, **ABER** nicht blind.

Ein Stripe basiert auf Menschen & Kommunikation im Mittelpunkt ...

- gemeinsame **Werte** & Prinzipien
- **Vertrauen** & psychologische Sicherheit
- **Transparenz** – Klarheit für alle
- **Feedbackschleifen** für ständiges „Inspect & Adapt“
- **Retrospektiven** mit kontinuierliche Verbesserung
- ...

Ein Stripe nutzt ...

- priorisierte **Ziele**, Backlogs mit Anforderungen & Architekturen
- dediziertes **Team**, ...
- klare **Verantwortung** (Stripe-Keeper),
- kurze **Zyklen** (timeboxes), ...

ABER nicht immer die Gleichen!

ABER nicht immer die Gleichen!

ABER nicht immer die Gleichen!



... und weitere **erfolgreiche Muster**

damit wir (uns) gemeinsam besser entwickeln

Individuelle Arbeit

- **klare Zieldefinition:**
 - Zweck, Scope, Zeit, ... ist definiert
 - sowohl das **große Ganze** als auch die **Details**
- Trennung von **Problem & Lösung**
- geeignete **Perspektiven & Sichten**
- konsistente **Modelle**
- **iterativ inkrementelle** Entwicklung:
vom Ziel über Umsetzung und Test zum Feedback von den Stakeholdern
- ...

Zusammenarbeit

- regelmäßige **Synchronisationspunkte**
vor allem in in arbeitsteiligen Prozessen
- gute **Kommunikation**
- und gute **Teamgrößen** (7 +/- 2)
- gemeinsame **Wertvorstellungen**,
- Vertrauen und Sicherheit
- ...

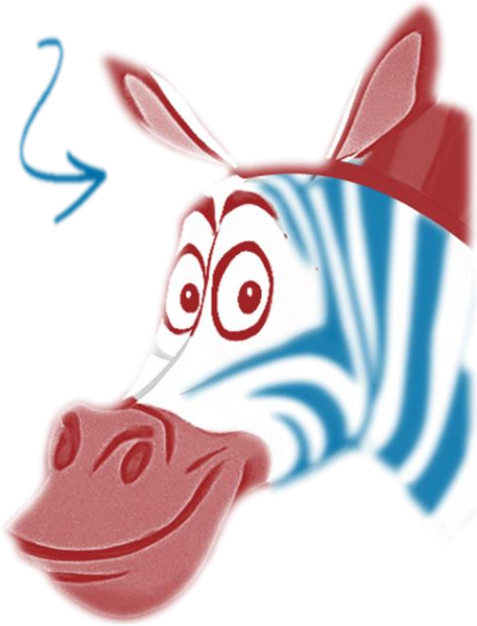
Zusammenarbeit von Teams & Organisationen

+ alle weiteren Angaben aus dem agilen Manifest

Head - & Tail Stripe

Alles hat ein Ende!

Head
Stripe



- beide oft sehr verschieden zum Rest des Projektes
- sind **NICHT DAS Muster** für den Rest des Projektes
- haben oft andere Verantwortungen oder sind Orte der **Verantwortungsübergabe** zwischen Lebenszyklusphasen darstellen

Tail
Stripe



- Starten eines Projekts oder Produktentwicklung
- Initiale Planungen
- Rahmenbedingungen bestimmen
 - Festlegen der Vision, Mission (B, T)
 - Vereinbaren von Grundsätzen, Werten, Mindset (O),
 - Gesamt-Auftragsklärung, Scope (T, B)
 - Aufbau einer initialen vernetzten Architektur (T), die später weiterentwickelt wird
 - Vereinbaren der möglichen Teams, Dorsal-Team, und Verantwortlichkeiten (O). Wird in den einzelnen Stripes später noch detailliert geklärt
- Kann auch beinhalten, (je nach Bedarf):
 - Vorentwicklungsphase, Design Thinking, Marktanalyse-Aktivitäten, Vertragsverhandlungen, ...

Benutzt Ideen von erfolgreichen anderen Mustern zu Organisationsmodellen!

Wie zum Beispiel:

- Dynamic Shared Ownership
- Soziokratie, Humanocracy, Holokratie
- Beyond Budgeting
- BetaKodex
- Liberating Structures
- ...

Die SEBRAS Streifen

"Streifen statt Sprints"

Stripe = 1 SEBRAS-Iteration

- SEBRAS strukturiert den Entwicklungsprozess in verschiedene Arten von Stripes
- verschiedene Stripe-Arten mit verschiedenen Zielen und Aufgaben
- ermöglicht eine flexible und anpassungsfähige Arbeitsweise
- orientiert sich an Sprints aus Scrum etc.
- Stripe-Länge: timeboxed, kann aber variieren, kein strenger Takt! - *"Rhythmus statt Takt"*
- Ein SEBRAS-Stripe kann gespalten sein, es gibt auch parallele (Unter-)Stripes für parallele Ziele und Teams
- Ein definiertes Team pro (Unter-)Stripe

Ein Denkmodell der AG agileSE der GfSE:

SEBRAS: *Systems Engineering for Business Reactivity and Agility with Stripes*

**Zweck: Re-Aktivierung der nutzbringenden Ideen aus
Agilität und Systems Engineering**

Der (Pattern-)Stripe

Die Entwicklungs-Streifen

- **Stripe-Aktivitäten:**

1. **Planning**
2. **Entwicklungs-Aktivitäten**
3. **Review**
4. **Retrospective**

- **Stripe-Dauer** zwischen 3 Tagen bis 6 Wochen
- **Stripe Team:** besteht aus den üblicherweise empfohlenen **7+/-2**, maximal jedoch 12 Mitgliedern

Die Entwicklungsaktivitäten im Pattern-Stripe

...bestehen i. Allg. aus einem **kompletten V-Modell-Zyklus** zur **Umsetzung des Stripe-Ziels**

- Trennen von **Problem** und **Lösungs-Phase** !
- **Modelliere iterativ mit MBSE** !
 - z.B. zur Identifikation des Stripe-Scopes, zur Problem-Analyse, zur eigentliche Lösungsfindung, ..
 - mit SysML-Modellen (Block, Use Case) und
 - mit agilen und informellen Modellen (Zielmodell, Personas, Tabellen..)
 - Agiles Modellieren und Implementieren (Wave-Model)
- Stelle **Konsistenz zum Gesamt-Modell** spätestens zum Stripe-Review her

Siehe dazu auch unser Paper vom Tag des Systems Engineering 2023:

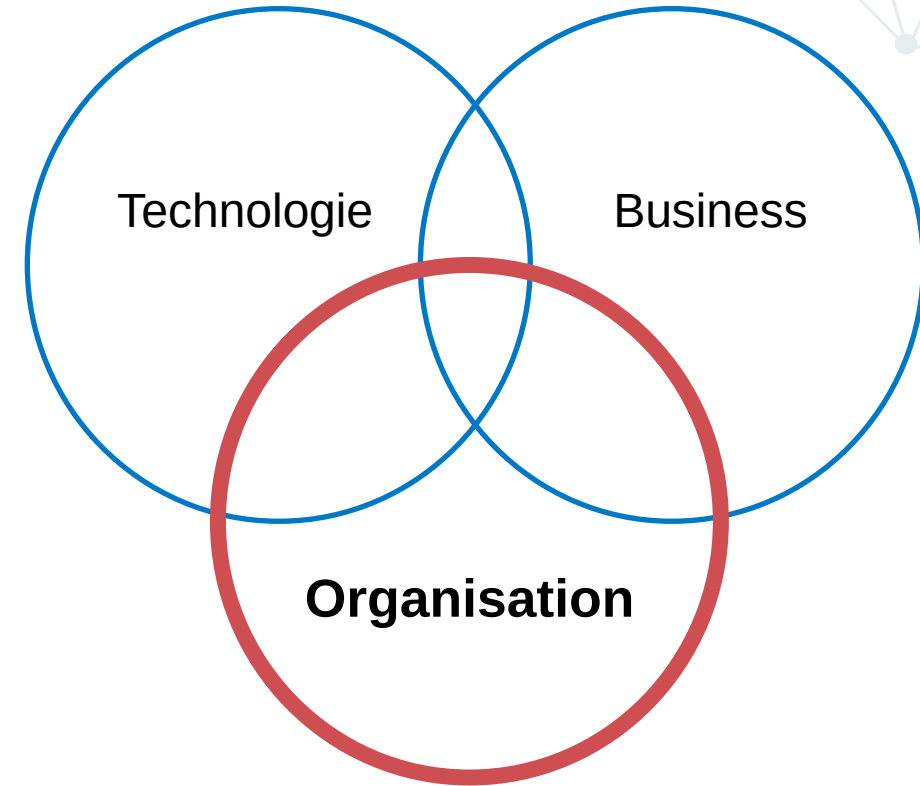
[TdSE23] P. Dahlke *et al.*, "Besser agil im Systems Engineering mit Modellen", *Tag des Systems Engineering*, Jg. 2023, S. 172–178, 2023.

T-O-B Perspektiven

Jeder Stripe berücksichtigt immer **ALLE** - wenn auch in unterschiedlicher Intensität

- **Technologie** - das zu entwickelnde System
Technischer Umfang, technische Ziele, Anforderungen, Architektur, Werkzeuge, Implementierung, Validierung, ...
- **Organisation** - das Entwicklungssystem
Strukturelle und kulturelle Aspekte, Organisationsstruktur, Teams, Rollen, Verantwortlichkeiten, Zusammenarbeitsmodelle, Kommunikationsschnittstellen, Werte und Prinzipien, ...
- **Business** - das Unternehmungssystem
Geschäftsziele, wirtschaftliche Ziele, strategische Ausrichtung, Stakeholder, Wertschöpfung, Anforderungsmangement, Risikomanagement, ...

Jeder Stripe **priorisiert immer ALLE Perspektiven**
- wenn auch in unterschiedlicher Intensität.
Ein der Perspektive hat im Stripe immer den Fokus.



Berücksichtige Conways Law
und **vernetze** permanent die Perspektive der
Organisation mit Business & Technologie

*„Organisationen, die Systeme entwerfen, [...] sind **gezwungen**,
Entwürfe zu erstellen, die die **Kommunikationsstrukturen der
Organisationen abbilden.**“*

M. D. Conway, 1968

Dieses **Muster** basiert auf der Beobachtung des , dass in einer auf verschiedenen Menschen/Teams/Unternehmen ... aufgeteilten Systementwicklung für die Definition der **Schnittstellen** im System bzw. zu dessen Umfeld stets (formalen und informelle) zwischenmenschliche **Kommunikation** notwendig ist.

Diese **Kommunikationsschnittstellen beeinflussen die Lösung** und deren Teile und Schnittstellen ganz entscheidend!

- haben die Aufgabe und einen Blick dafür, **das große Ganze & die Details zugleich** in den Augen zu behalten!
- Jede T-O-B Perspektive hat eine **Verantwortlichkeit** (Accountability)
- **Keeper** des Stripes ist immer **die Person der Fokusperspektive**
- kann - je nach Erfahrung - pro Stripe wechseln
- **sind NICHT „die Chefs“** sondern Facilitator & Servant Leader



Stripe Planning

Planung des Pattern-Stripes

Input aus den
TOB-Backlogs

- Ein jeder Stripe startet mit einem Planning
- Der "**Stripe-Keeper**" fungiert als Planning-Moderator

ZIELE

Vereinbarung von **Zielen und Umfang** für den jeweiligen Stripe,
aus den Inputs der TOB-Backlogs

DAUER & STRUKTUR

Vereinbarung von **Dauer** des Stripes
Vereinbarung der **Struktur des Stripes**: parallele Stripes, Sub-Stripes

TEAM & RESSOURCEN

Vereinbarung von **Team-Zusammensetzung** und **Ressourcen** des Stripes
Vereinbarung der **Kommunikations-Schnittstellen** (intern und extern)

AUFGABEN

Aufgabenplanung und Erörterung der notwendigen Entwicklungsaktivitäten

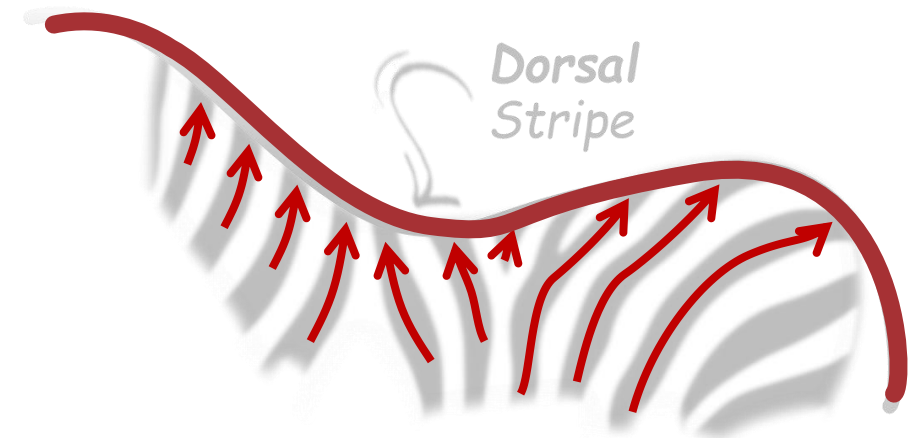
Stripe Planning – weitere Notizen

Weitere Aspekte bei der Planung des Pattern-Stripes



- **Auftragsklärung:** Definition des technischen **Stripe-Scopes (Umfang der betroffenen Subsysteme)** ist wichtig, in Abstimmung mit dem Dorsal-Team.
- **Stripe-Ziel:** ist, ähnlich dem Scrum Guide, ein “**useful increment**“, so dass für jeden Reifegrad bzw. jedes Modell der Zweck (T-O-B) für einen Stakeholder als nützliches und nutzbares Ergebnis klar definiert ist.
- **Muster und Methoden aus Scrum-Planning** oder SAFe PI-Planning etc. können hier gut angewandt werden
- **Mindset:** In jedem Planning sollten die vereinbarten **SEBRAS Werte und Prinzipien** (definiert in einem Head-Stripe) kommuniziert werden.
(Eine Anpassung kann in der Stripe-Retrospektive passieren)
- **Betrachtung aller T-O-B-Perspektiven im Planning:**
 - **B:** Die **Business-Ziele** müssen dem gesamten Team bewußt sein und auf den Pattern-Stripe heruntergebrochen werden
 - **T:** Klare Definition des **technischen Umfangs und Aufgaben** (Stripe-Scope)
 - **O:** Klare Definition des aktuellen **Teams, Zusammenarbeitsmodell** intern und mit Externen, inklusive aller Kommunikationsschnittstellen während des Stripes
- **Bei Unklarheit der Stripe-Aufgaben:** Abstraktere Ziele werden festgelegt. Die detaillierte Klärung der Aufgaben wird dann in die "Problemphase" des Stripes verlegt. Möglicherweise kann auch der genaue Stripe-Scope dann erst exakt definiert werden.

- **DER Synchronisationspunkte** aller Stripes im **Dorsalmodell**
(Single Source of Truth, Digital Thread, ALM/PLM, Digital Master, Digitaler Zwilling, ...)
- Vernetzung, Kohärenz und Nachverfolgbarkeit aller **Stripe Ergebnisse** über **alle T-O-B Perspektiven**
- Ausgangspunkt für die Backlog bzw. **Planung** der nächsten Pattern Stripes
- **getaktet nach Zyklen** der Stripes **oder kontinuierlich** (CI/CD) gepflegt
- Dorsal-Stripe **Keeper** festlegen
- bei mehr als 5 Teams ein dediziertes **Dorsal-Stripe Team** bilden



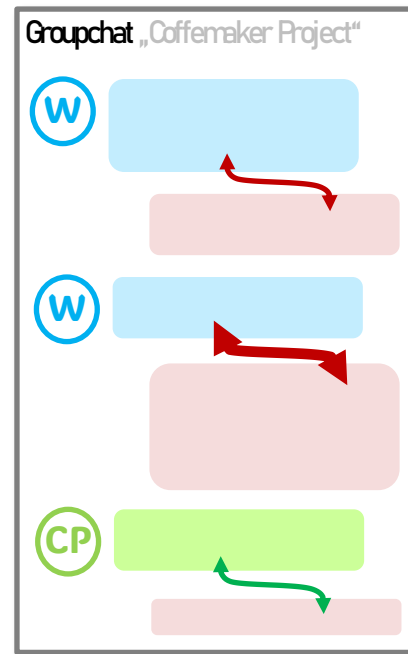
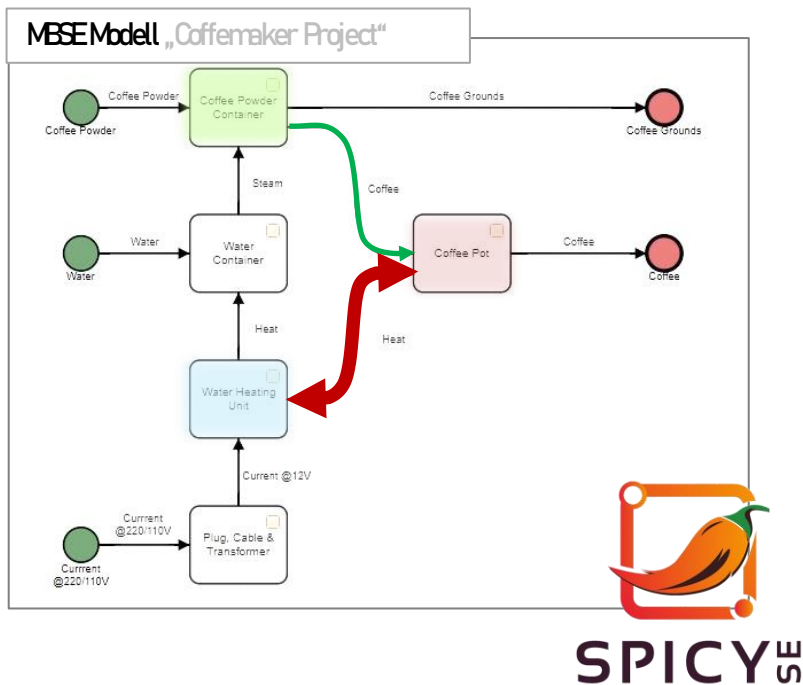
6

Ausblick

SEBRAS brauchen die richtigen Werkzeuge

Damit in Summe ein **schmackhaftes Produkt** bei allem herauskommt

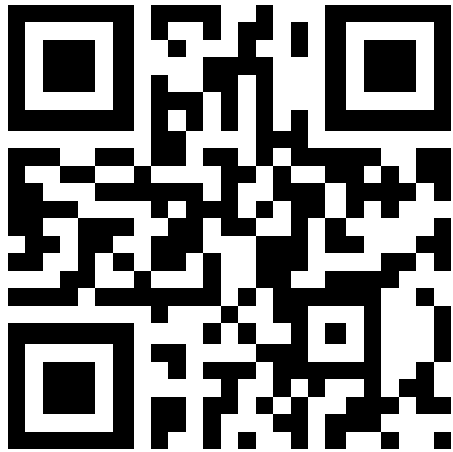
- **einfach und gerne** von allen zu verwenden!
- **Vernetzung** der Daten aller B-O-T Perspektiven,
 - insb. **Technologie & Organisation (z.B. Kommunikation)**; Conways Law



→ Ideen dazu bei
[Robert Hämisch - Co-Founder - Spicy SE | LinkedIn](#)

Mit **KI** wird das alles noch besser ...

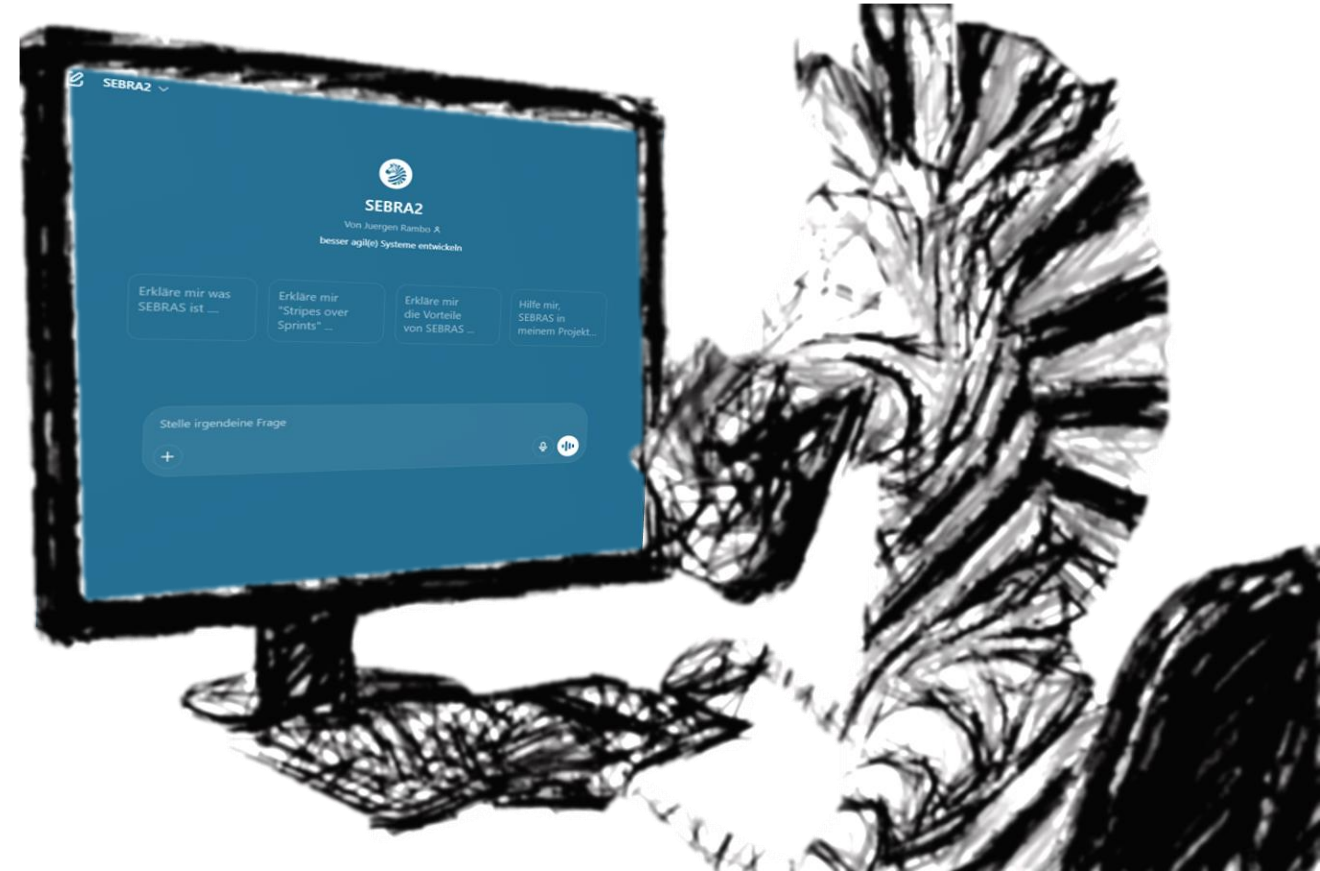
Ein SEBRAS ChatBot scharrt in ChatGPT bereits mit den Hufen



Als OpenAI ChatGPT-Nutzer gerne testen unter:

<https://tinyurl.com/SEBRAS>

<https://chatgpt.com/g/g-nqiQ0wH85-sebra2>



Skaliere über eine Herde von Zebras

Stripes ermöglicht auch ein flexibleres Verändern und Wachsen von Teams.

- Projekte verändern sich in ihrer Dauer von oftmals Monaten/Jahren in ihrer Teamzusammensetzung!

erste Überlegungen:

- **Dorsal-Stripe**
für Stabilität und strategische Ausrichtung etablieren
- **Koordinationsmechanismen ab 4–5 Teams**
z.B. unfix® (Jürgen Appelo) und Soziokratie / Holacracy® nutzen
- **Anzahl von Teams klein halten** (auch 7+/-2 Teams)
und **Dunbarzahl** ~100-250 beachten

→ >10 Teams besser eine **SEBRAS-Herde** bilden



Zertifikatskurse sind angedacht

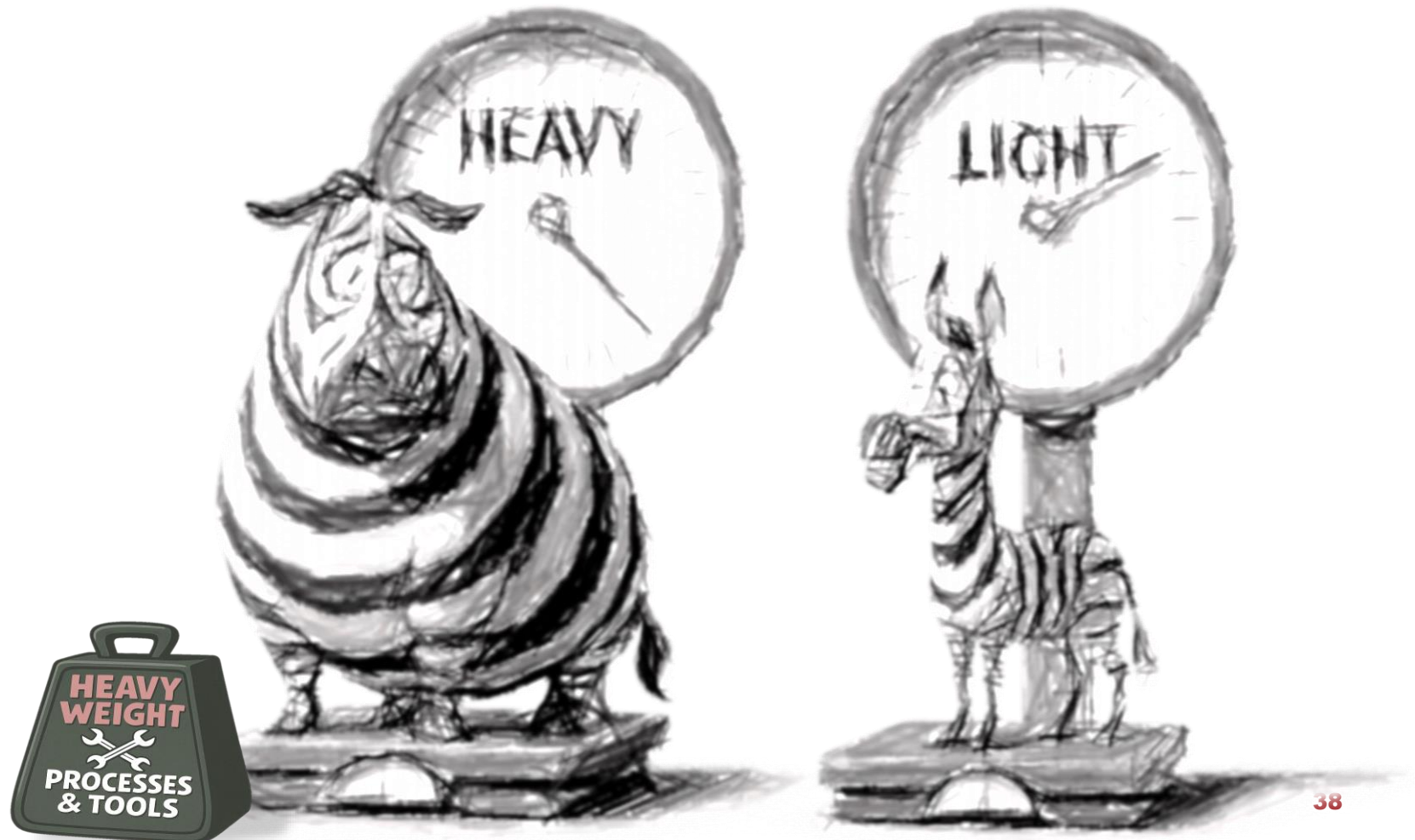
Rabatt für Herden und Mitglieder der GfSE Arbeitsgruppe Agile SE geplant.

Natürlich nicht.
Das Denken bleibt frei!



Aber bitte alles leichtgewichtig ...

Dicke Zebras sind meist träge und wenig agil ;-)



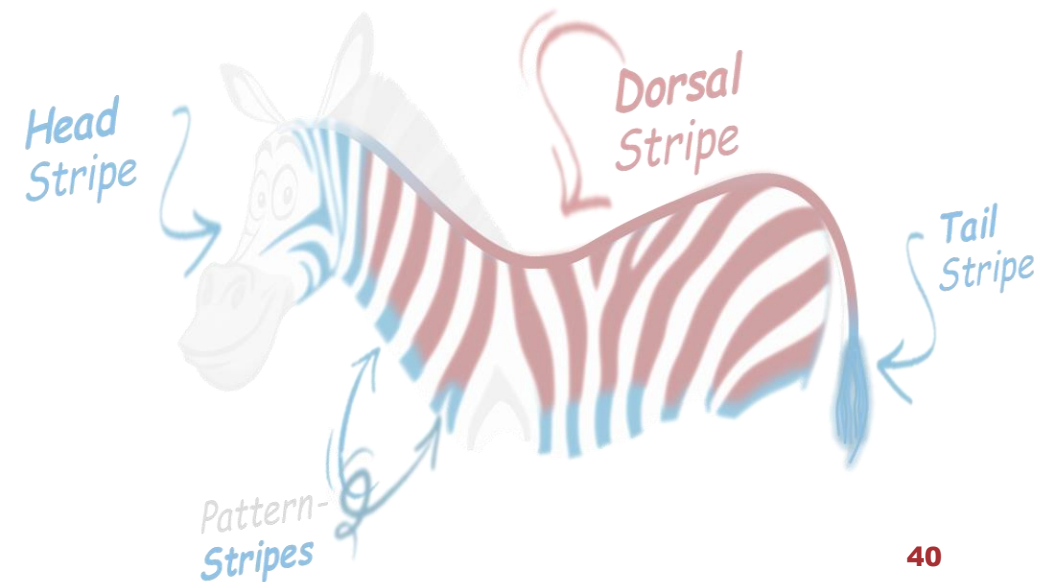
7

Zusammenfassung & Diskussion

Zusammenfassung

Systems **E**ngineerin for **B**uisiness **R**eactivity and **A**gility with **S**tripes

- **SEBRAS**
als Denkmodell für modellbasiertes **agiles Systems Engineering**
- **Stripes over Sprints** - für die Muster erfolgreicher Entwicklung
- **Perspektiven** von **Technologie** – **Organisation** – **Business** ineinander **vernetzen**
- **Dorsal Stripe** als zentrales (MB)SE-Modell hält T-O-B zusammen
- **Stripe Keeper** tragen mehr als eine Verantwortung
- **leichte Tools** und **KI** werden helfen
- **Herden**, wenn ein SEBRA nicht reicht
- machs **leicht** und nimms nicht zu schwer
- und lass Dich nicht zertifizieren!



SEBRAS ist ein Denk**MODELL**

Es soll zum Denken anregen, was in der Entwicklung **wirklich wirklich wichtig** ist!



*„Wir hoffen, dass unsere gemeinsame Arbeit [...] anderen hilft,
über Entwicklung, Methoden & Organisationen
auf neue -agilere- Art und Weise nachzudenken.*

*Wenn dem so ist, haben wir unsere Ziele erreicht.“ **

*frei zitiert in Anlehnung an: <https://agilemanifesto.org/history.html>