

„Agile needs Systems Engineering“

Who we are – Working Group Agile SE

Kurzvorstellung Dirk Stüker



Motivation zur Mit-
Eigene Erfahrungen
Engineering teilen,
lernen und die Com-
Branche:
Automotive



Privates: 47 Jahre, verheiratet, 3 Töchter, 1 Katze
Hobbies: Jonglieren, Skifahren, Smart Home, Lego-Roboter, Raps
Beruflicher Werdegang:
-2000: Informatik, Oldenburg – 2 terms Computer Science Exeter
2000-2003 Dissertation: Sensorfusion (Radar, Kamera, Lidar)
2003-2010 Volkswagen Forschung Fahrerassistenz und
hochautomatisiertes Fahren
2010-2015 Volkswagen ADAS Architektur und Vorentwicklung
2015-2018 WABCO, ADAS Chief Engineer, Nutzfahrzeuge (Radar,
2018-2020 Volkswagen, ADAS Chief Engineer, Nutzfahrzeuge

Kurzvorstellung Alexander Neng



Motivation zur M-
Komplexe Systeme
bei der Umsetz-
sich ändernden
einen Austausch
Branche:
Medizin, Autom-



Privates: 40 Jahre, verheiratet, zwei Kids
Hobbies: Triathlon, Bergsteigen
Beruflicher Werdegang:
seit 2019 Head of S-
2015-2019 Systems
2012-2015 Studium Wlrfn, TH Nürnberg
2000-2012 IT Abteilung, Deutsche Luftwaffe

Kurzvorstellung Thaddäus Dorsch



Privates: wohnt in Landsberg am Lech (80 km westl. von München), 54 Jahre, verheiratet,
1 Tochter (15 J.), 1 (Stief)sohn (30 J.)
Hobbies: zeitgenössische klassische Komposition, Singen, Bratsche, Wandern, Fußball, Ski, Rad
Beruflicher Werdegang:
1995 Elektrotechnik TU München, Fachrichtung Nachrichtentechnik
2005 Promotion FAU Erlangen-Nürnberg Digitale Datenübertragung und Informationstheorie
2005 - 2014: DLR, Airbus AGT, Airbus Defense, Robotik & Schwarz
seit 2015: HOD GmbH als Berater und Trainer für SE, RE und Agilität
mit Projekten u.a. bei BMW, Bosch, Sartorius
Qualifikationen: Certified: CSEP (2013), IREB Requirements Engineer,
IREB Requirements Manager, Scrum Master, SAFe-Agilität, Lehrbeauftragter TH Rosenheim

Kontakt:
Email: thaddaeus.dorsch@hodgroup.com
Linked in: <https://www.linkedin.com/in/thaddaeus-dorsch-bb64baa/>
Xing: https://www.xing.com/profile/Thaddaeus_Dorsch/
Agile: <https://www.linkedin.com/company/agile-hardware-and-systems-group/>
Agile Hardware and Systems Group International linked in
Agile Hardware and Systems - Xing Group

Motivation zur Mitarbeit Agile SE:
Agilität + SE = mehr als die Summe.

Branche:
Beratung und Training,
vor allem Automotive, Biotech,
Telekom

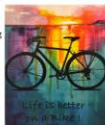
seit 2012 Certified Systems Engineering Professional (CSEP)
seit 2017 Provisional Assessor Automotive SPIKE

Kontakt:
<https://www.linkedin.com/in/stephan-teutsch-8755681a0/>
https://www.xing.com/profile/Stephan_Teutsch/

Kurzvorstellung Stephan Teutsch



Motivation zur Mitarbeit Agile SE:
Austausch von Erfahrungen
Agilität und Systems Engineering vernetzen
Branche:
Automotive, Softwareentwicklung,
Qualitätsmanagement, Prozessentwicklung



Jürgen Rambo

<https://de.linkedin.com/in/junggerambo>
Es geht mir darum ...
**(sich) gemeinsam
besser zu entwickeln**



- Kind der 70'er & Student in den 90'er
- Allgemeiner **Maschinenbauer** der TU Darmstadt & ETSEIB Barcelona
Produktadatentechnologie, Produktentwicklung & Produktionstechnik
- beruflich seit 2001 in wechselnden in Rollen als
(IT-) Forscher, Business Consultant, Trainer, Coach ... „**Entwicklungshelfer**“
im (Requirements - und Systems) Engineering (agil) unterwegs



Kurzvorstellung Tim Weilkiens

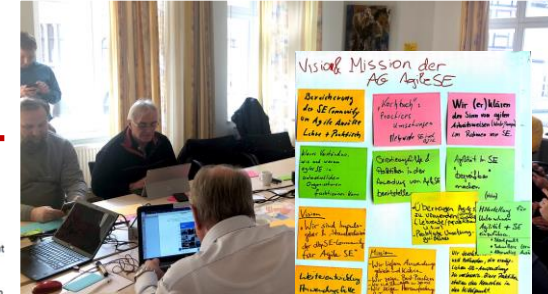


Motivation zur Mitarbeit Agile SE:
1. Agilität im Systems Engineering etablieren
2. Voneinander Lernen
Branche:
Beratung, Training

Privates: 49 Jahre, verheiratet, 2 Kinder
Hobbies: Joggen, Segeln, Wattenmeerichut
Beruflicher Werdegang:
Studium Informatik mit Nebenfach Medizin
Entwicklung Anästhesiegeräte bei Dräger Meazmechrmk
seit 2001 Berater, Trainer bei oose
seit 2001 Mitglied OMG und Entwicklung von Standards UML, SysML,
BPMN, ...
2010-2012 Geschäftsführer von oose
seit 2012 Vorstand der oose-Genossenschaft
seit 2015 Inhaber Verlag MBSE4U

Kontakt: <https://www.linkedin.com/in/timweilkiens/>
https://www.xing.com/profile/Tim_Weilkiens
tim.weilkiens@oose.de
tim@mbse4u.com

22.10.2020



Interested?
Drop us a line:

AGAgileSE@gfse.org

- What is Agility and what is Systems Engineering
- Agile vs. Systems Engineering – comparing apples with pears
- Agile Systems Engineering



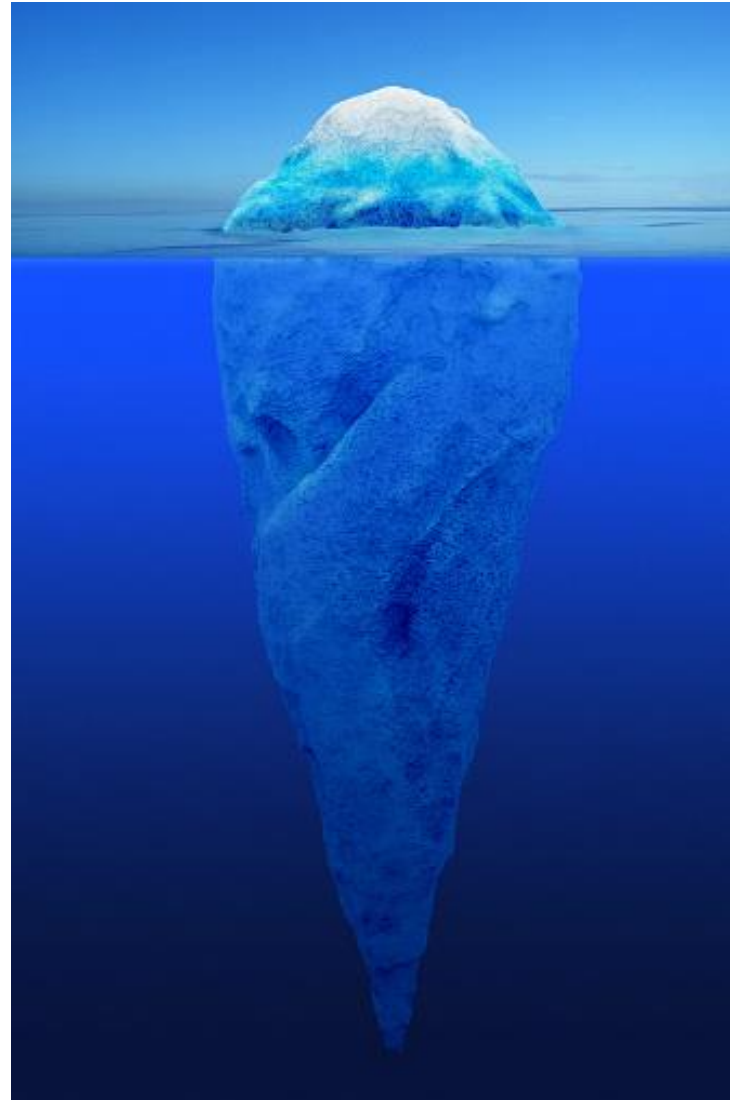
What is Agility and what defines Systems Engineering

A rapid whole-body movement with change of speed or direction in response to an impulse, but without loss of control.

Source: Sheppard und Young (2006)

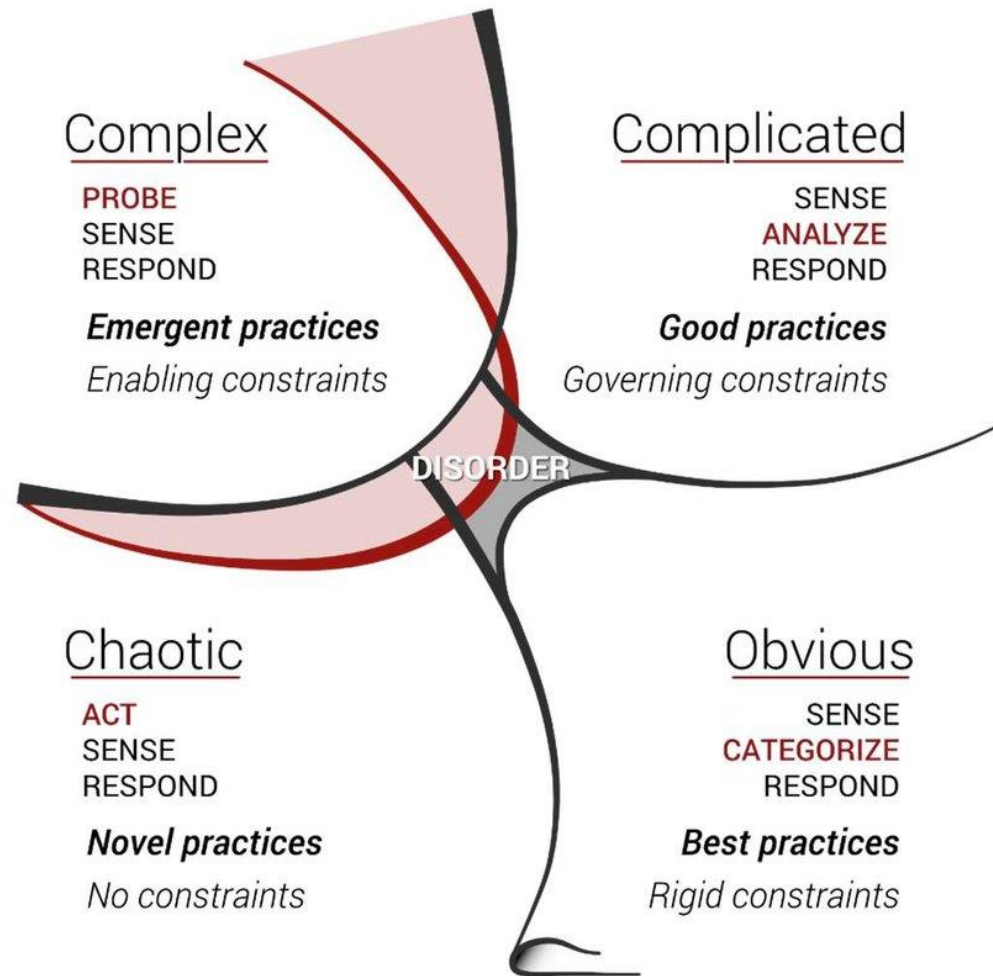
Methods (Scrum, XP,
SAfe...)

Practices:
Daily, Planning...)



Principles

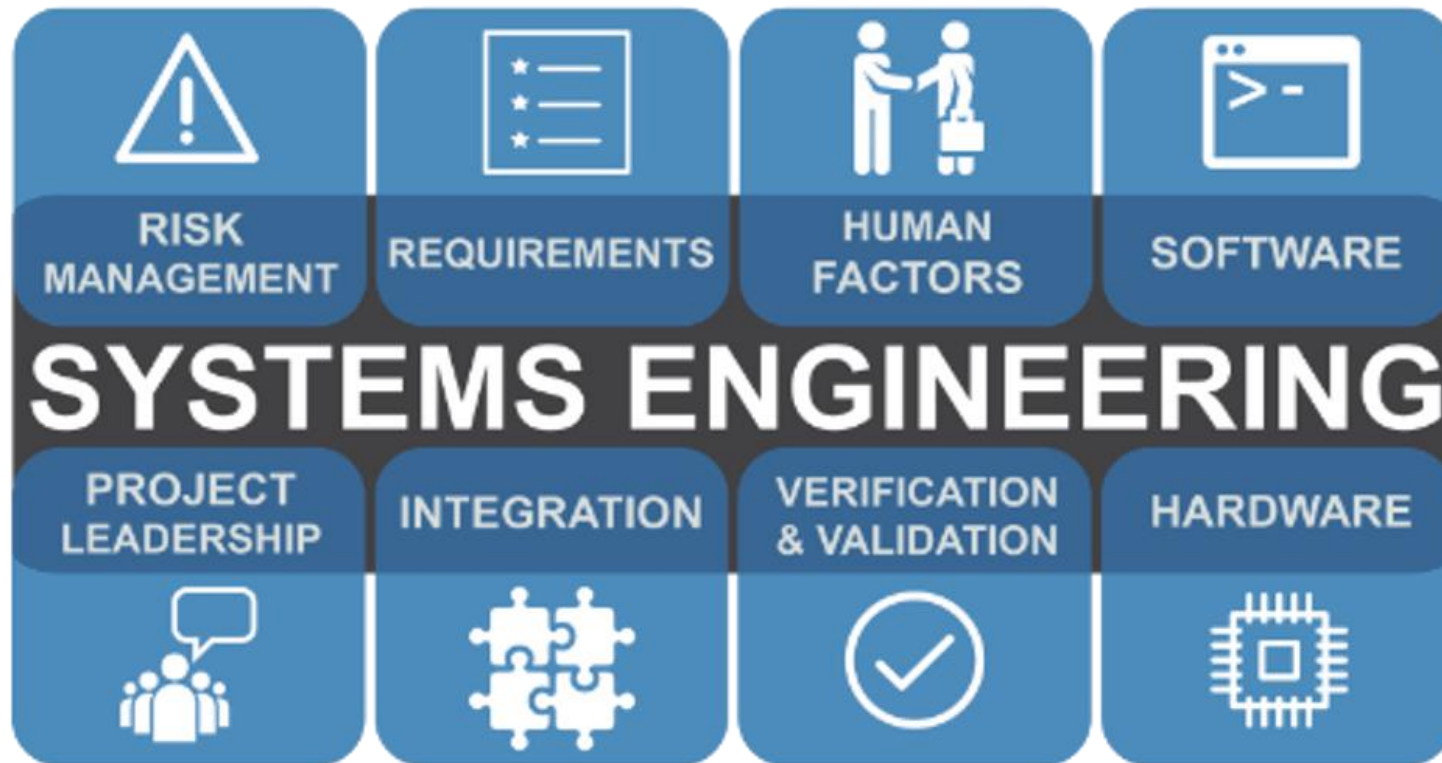
Values



Source: [Cognitive-Edge](#)

Systems Engineering is a transdisciplinary and integrative approach to enable the successful realization, use, and retirement of engineered systems, using systems principles and concepts, and scientific, technological, and management methods.

Source: Incose.org



Source: incose.org

Agile vs. Systems Engineering – Comparing Apples with Pears.

Agile vs. Systems Engineering

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Dave Thomas
Martin Fowler	Brian Marick	

Source: <http://agilemanifesto.org/>

SE Research Consortium

Systems Engineering Principles

Systems engineering postulates form the basis of the principles of systems engineering. Principles are accepted truths which apply throughout the discipline. These truths serve as a guide to the application of systems engineering.

Reference Section 3.2 in
[nasa_tp_20205003644_interactive2.pdf](#)

- Principle 1: Systems engineering integrates the system and the disciplines considering the budget and schedule constraints.
- Principle 2: Complex systems build complex systems.
- Principle 3: A focus of systems engineering during the development phase is a progressively deeper understanding of the interactions, sensitivities, and behaviors of the system, stakeholder needs, and its operational environment.
 - Sub-Principle 3(a): Mission context is defined based on the understanding of the stakeholder needs and constraints.
 - Sub-Principle 3(b): Requirements and models reflect the understanding of the system.
 - Sub-Principle 3(c): Requirements are specific, agreed-to preferences by the developing organization.
 - Sub-Principle 3(d): Requirements and design are progressively elaborated as the development progresses.
 - Sub-Principle 3(e): Hierarchical structures are not sufficient to fully model system interactions and couplings.
 - Sub-Principle 3(f): A Product Breakdown Structure (PBS) provides a structure to integrate cost and schedule with system functions.
 - Sub-Principle 3(g): As the system progresses through development, a deeper understanding of the organizational relationships needed to develop the system are gained.
 - Sub-Principle 3(h): Systems engineering achieves an understanding of the system's value to the system stakeholders.
 - Sub-Principle 3(i): Systems engineering seeks a best balance of functions and interactions within the system budget, schedule, technical, and other expectations and constraints.
- Principle 4: Systems engineering has a critical role through the entire system lifecycle.
 - Sub-Principle 4(a): Systems engineering obtains an understanding of the system.
 - Sub-Principle 4(b): Systems engineering defines the mission context (system application).
 - Sub-Principle 4(c): Systems engineering models the system.
 - Sub-Principle 4(d): Systems engineering designs and analyzes the system.
 - Sub-Principle 4(e): Systems engineering tests the system.
 - Sub-Principle 4(f): Systems engineering has an essential role in the assembly and manufacturing of the system.
 - Sub-Principle 4(g): Systems engineering has an essential role during operations, maintenance, and decommissioning.
- Principle 5: Systems engineering is based on a middle range set of theories.
 - Sub-Principle 5(a): Systems engineering has a physical/scientific basis specific to the system.
 - Sub-Principle 5(b): Systems engineering has a mathematical basis.
 - Sub-Principle 5(c): Systems engineering has a sociological basis specific to the organization(s).
- Principle 6: Systems engineering maps and manages the discipline interactions within the organization.
- Principle 7: Decision quality depends on the system knowledge present in the decision making process.
- Principle 8: Both policy and law must be properly understood to not overly constrain or under constrain the system implementation.
- Principle 9: Systems engineering decisions are made under uncertainty, accounting for risk.
- Principle 10: Verification is a demonstrated understanding of all the system functions and interactions in the operational environment.
- Principle 11: Validation is a demonstrated understanding of the system's value to the system stakeholders.
- Principle 12: Systems engineering solutions are constrained based on the decision timeframe for the system need.
- Principle 13: Stakeholder expectations change with advancement in technology and understanding of system application.
- Principle 14: The real physical system is the only perfect representation of the system.

Source: <https://www.nasa.gov/consortium/SystemsEngineeringPrinciples>

Let's compare the principles of agile and SE

Agiles Prinzip 1

Our highest priority is to **satisfy** the **customer** through early and **continuous** delivery of valuable software.

Systems Engineering Prinzip 3

A focus of systems engineering during the development phase is a **progressively deeper understanding** of the interactions, sensitivities, and behaviors of the system, **stakeholder needs**, and its operational environment.

The customer as one stakeholder is in central focus of the development of the product in both viewpoints. The term stakeholder is widening the focus to additional aspects.

Agiles Prinzip 2

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

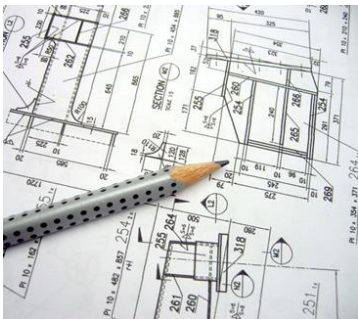
Systems Engineering Prinzip 13

Stakeholder expectations change with advancement in technology and understanding of system application.

Changing requirements is part of the system evolution.

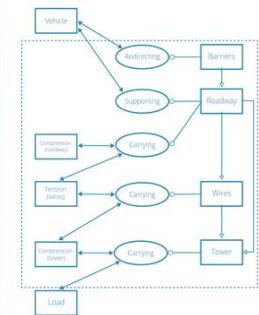
Agiles Prinzip 9

Continuous attention to **technical excellence** and **good design enhances agility.**



Systems Engineering Prinzip 7

Decision quality depends on the **system knowledge** present in the decision-making process.



Thrive for excellence and good decision by design.

The Foundation complex systems engineering as conclusion of agile manifesto

Almost the „Software Manifesto values“

- Individuals and **interactions** over processes and tools
- Working **software-solutions** over comprehensive documentation
- Customer **collaboration** over contract negotiation
- **Responding to change** over following a plan

lead to the foundation of **complex systems engineering**

- **Multifunctional teams** over engineering silos
- Focus on **purpose** over focus on requirements
- **Empowered teams** over tasked individuals
- **Early learning** over late failures

▶ **„Agile Systems Engineering Manifesto“ ?** (agile-systems-engineering.com)

Better Agile with Systems Engineering!

Agile does not contradict Systems Engineering!



Agile as development leadership mindset focusing on how the Organization is shaped and how the team operates and interacts.



Systems Engineering as engineering discipline is focusing on what is to be developed and what are the outcomes / products to be delivered to achieve the working solution.



Neither Systems Engineering nor Agile dictate one or the other process model.

We will need agile systems engineering to develop complex systems in an agile environment.

- Sheppard und Young (2006) - <https://de.abcdef.wiki/wiki/Agility>
- [Manifesto for Agile Software Development \(agilemanifesto.org\)](http://agilemanifesto.org)
- [Agile Systems Engineering \(agile-systems-engineering.com\)](http://agile-systems-engineering.com)
- [Systems Engineering Principles | NASA](#)
- David F. McClinton: 25 Laws of Systems Engineering
- <https://www.sebokwiki.org/wiki>
- INCOSE Systems Engineering Handbook
- ISO/IEC/IEEE 15288:2015
- [System and SE Definitions \(incose.org\)](http://incose.org)
- Melvin E. Conway: How Do Committees Invent? In: F. D. Thompson Publications, Inc. (Hrsg.): Datamation. Band 14, Nr. 5, April 1968, S. 28–31
- Matthew Skelton and Manuel Pais "Team Topologies"

Thanks for all the fish!

- Agile Principle 3,
- **Deliver working software solutions frequently**, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Agile Principle 7
- **Working software solution** is the primary measure of **progress**.
- Agile Principle 8
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a **constant pace** indefinitely.
- Systems Engineering does **not** define the **time-frame** or **cycle of development** as this is to be defined by the used development process (e.g., agile).
- BUT the definition of **working solution** with systems engineering methods enables agile development.

Development pace and cycles are not predefined by systems engineering, but SE helps to define the working solution.

- Agile Principle 4
- Businesspeople and developers must **work together daily** throughout the project.
- Agile Principle 6
- The most efficient and effective method of **conveying information** to and within a development team is **face-to-face conversation**.

→ Systems Engineering does emphasize the **interdisciplinary collaboration and communication tasks** of systems engineers.

Communicate, communicate, communicate...

- Agile Principle 5
- Build projects around **motivated individuals**. Give them the environment and **support** they need and **trust** them to get the job done.
- Agile Principle 11
- The best architectures, requirements, and designs emerge from **self-organizing teams**.
- Agile Principle 12
- At **regular intervals**, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

→ Systems Engineering does **not define how** Teams are formed and motivated.

Organization of teams and motivation of people is not in the scope of systems engineering.