Hochschule Bremen
City University of Applied Sciences

HSB

Agile needs Systems Engineering

Prof. Dr.-Ing. Jasminka Matevska

31. Mai 2023, Universität Hildesheim
Workshop "Software Engineering in Cyber-Physical Production Systems”

Agenda

01 What is Agility?

02 What is Systems Engineering?

03 Agile vs. Systems Engineering – comparing Apples and Oranges

04 Better agile with Systems Engineering

05 Who we are – Working Group Agile SE at GfSE

01

What is Agility?

What is Agility?

> A rapid whole-body movement with change of speed or direction in response to an impulse, but without loss of control.
>
> [Sheppard und Young, 2006]

Agile Approach

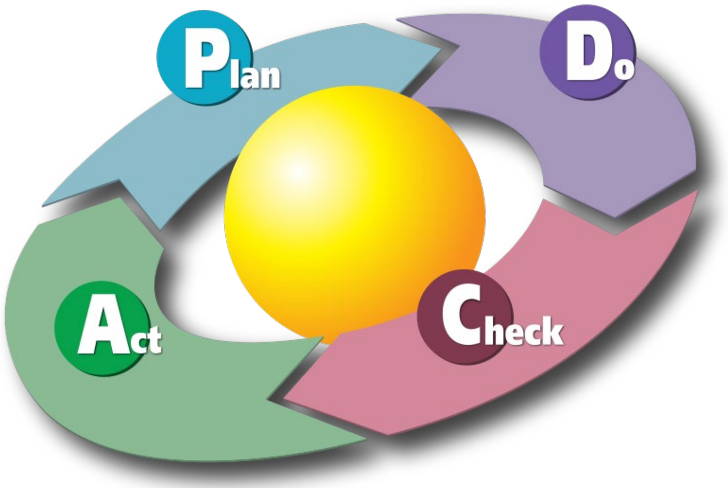
Iterative incremental way of working

Self-organized cross-functional teams

Early and continuous delivery to the customer

Continuous improvement trough reviews/re-evaluation

→ An „Inspect & Adapt“ Framework

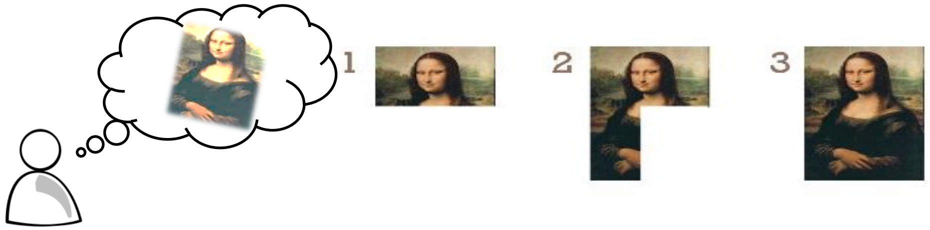


Iterative and Incremental

Iterative

Learn by doing

Incremental



Expose growing system

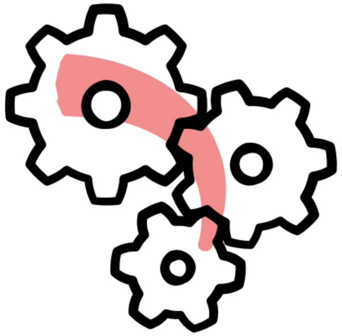
Iterative Incremental

Define the vision and then deliver incrementally

Agile Approach

Mindset

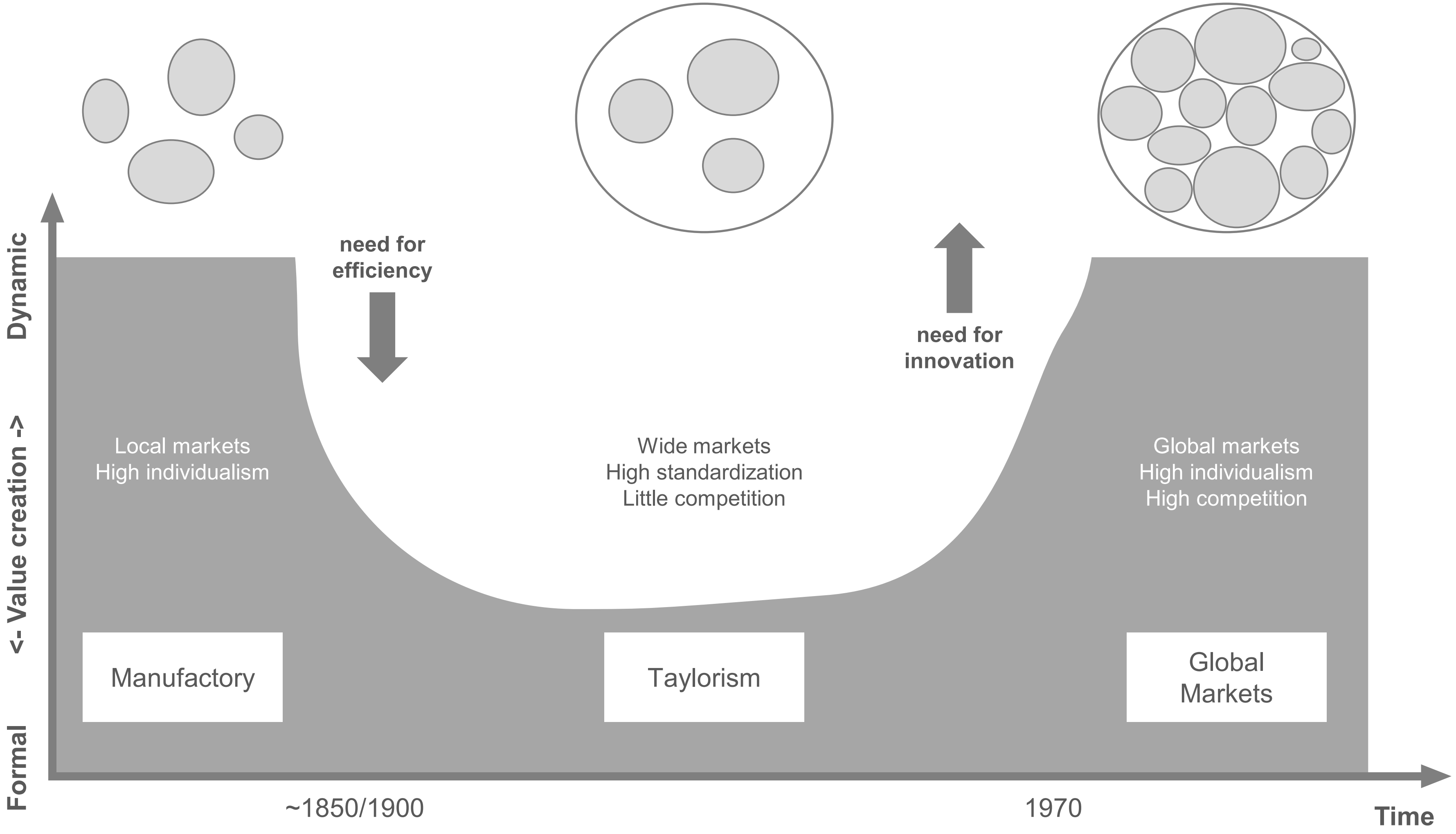
4 Values
12 Principles



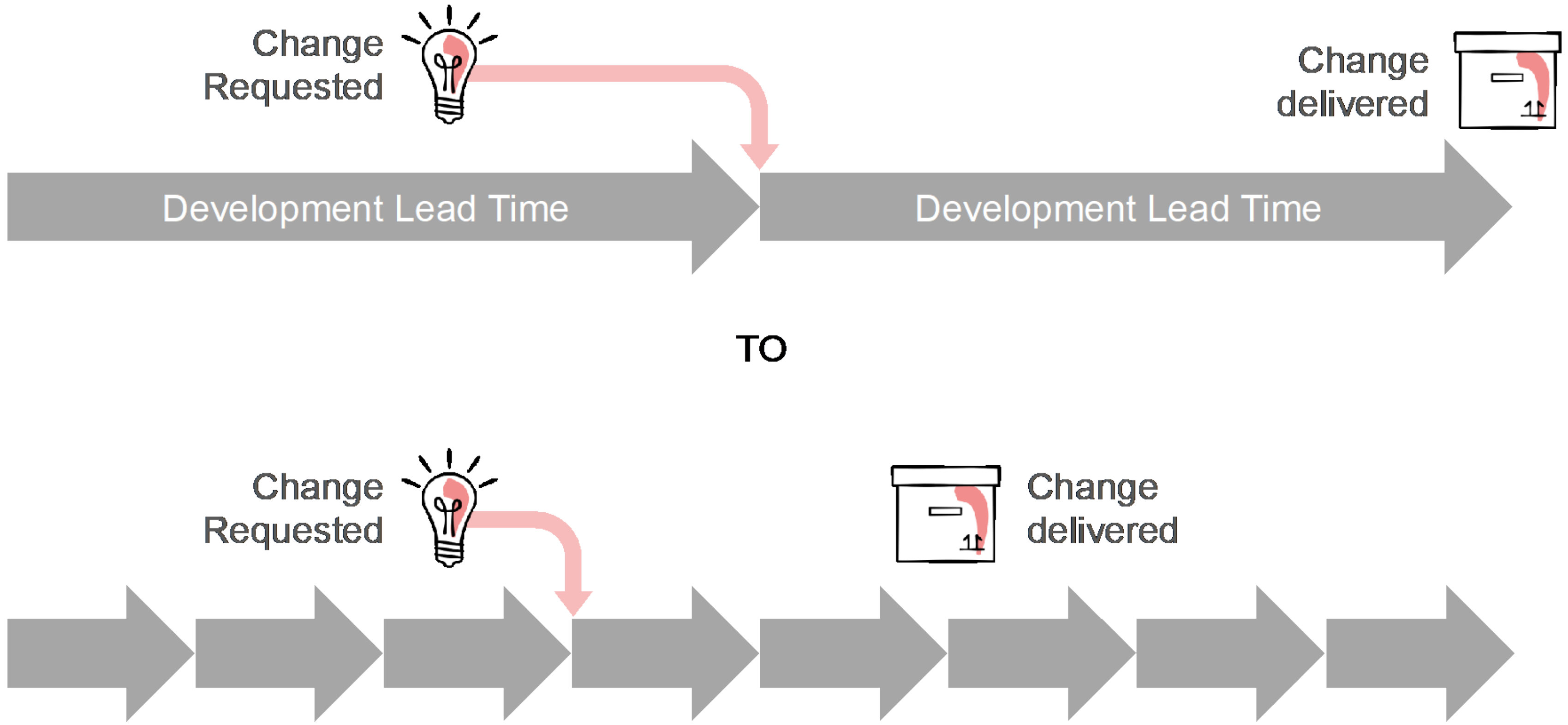
Frameworks
Tools
Methodologies

The real challenge is to **BE** Agile rather than **DOING** Agile

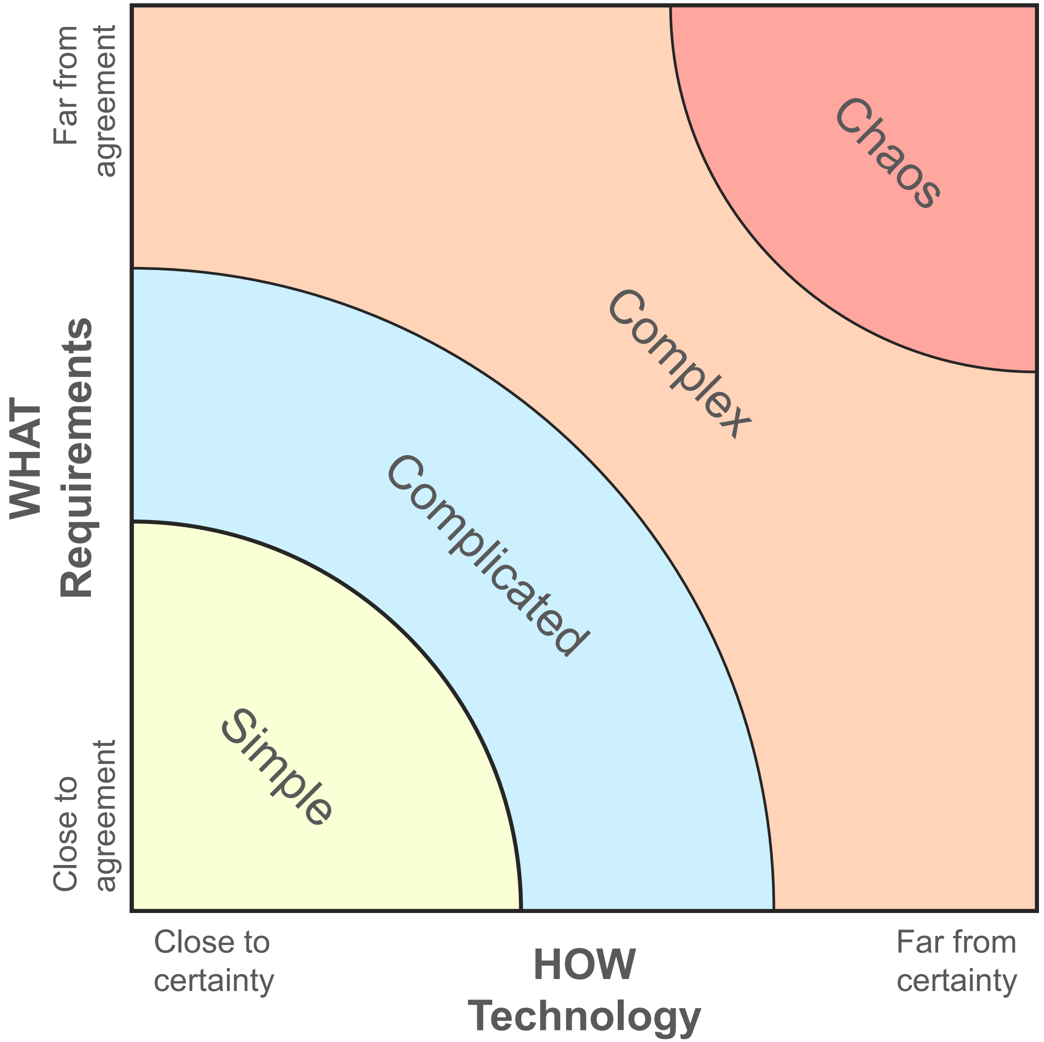
Why agile? – Taylor Tub



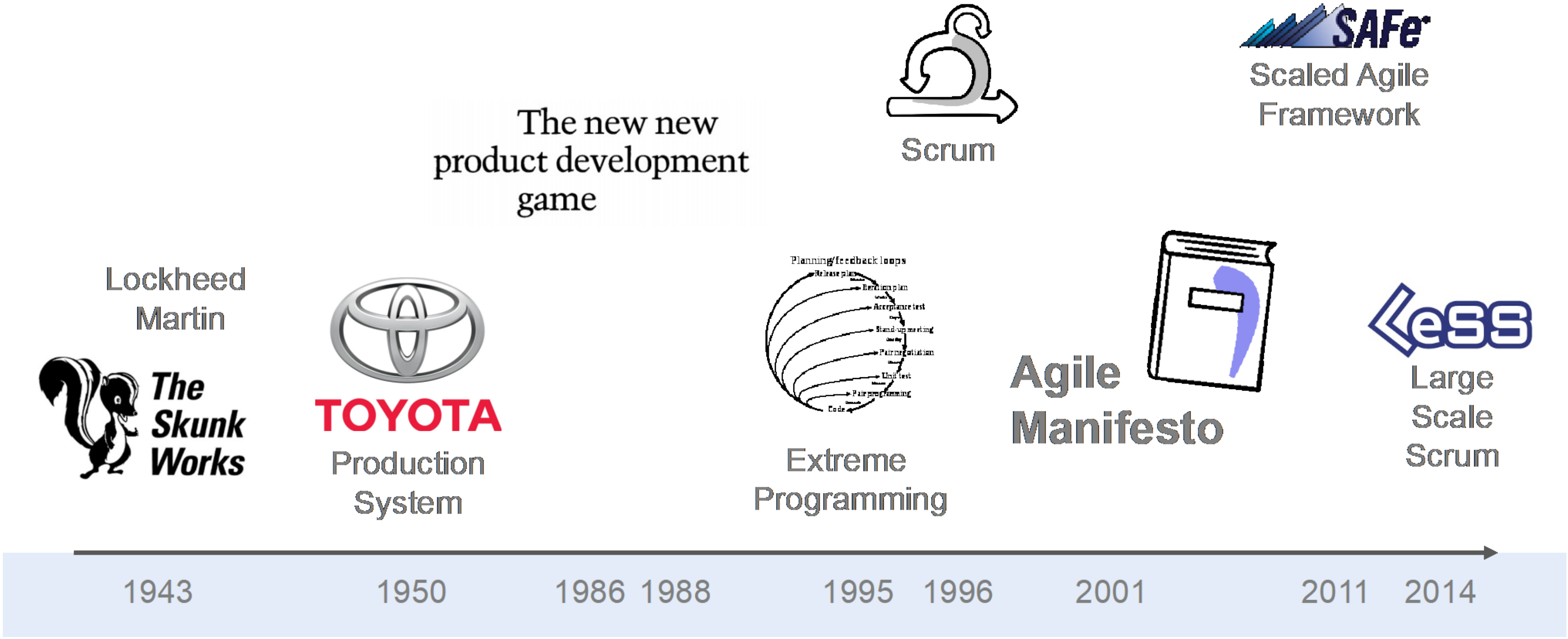
Why agile? – Fast reaction to changes



Why agile? – Stacey Matrix



Roots of Agile



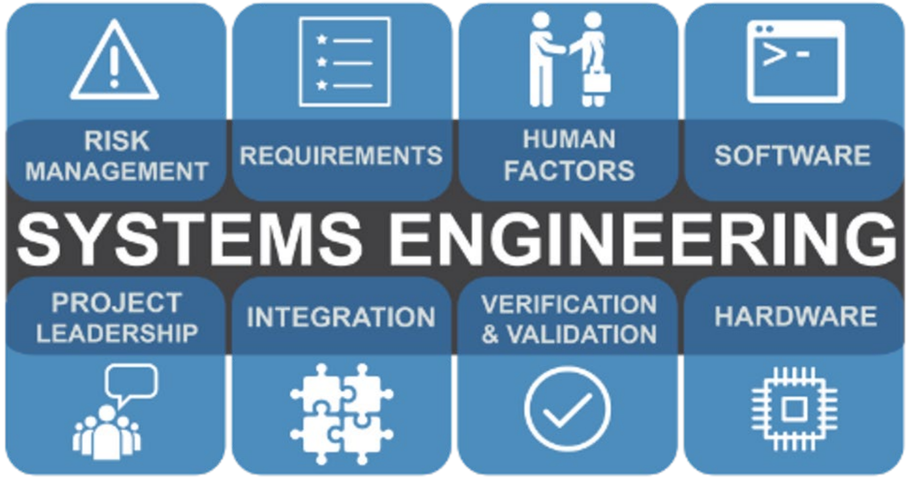
02

What is Systems Engineering?

What is Systems Engineering?

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

[INCOSE.org]



Example – Space Systems Engineering


ECSS-E-ST-10C Rev.1
15 February 2017

EUROPEAN COOPERATION
ECSS
FOR SPACE STANDARDIZATION

Space engineering

System engineering general requirements

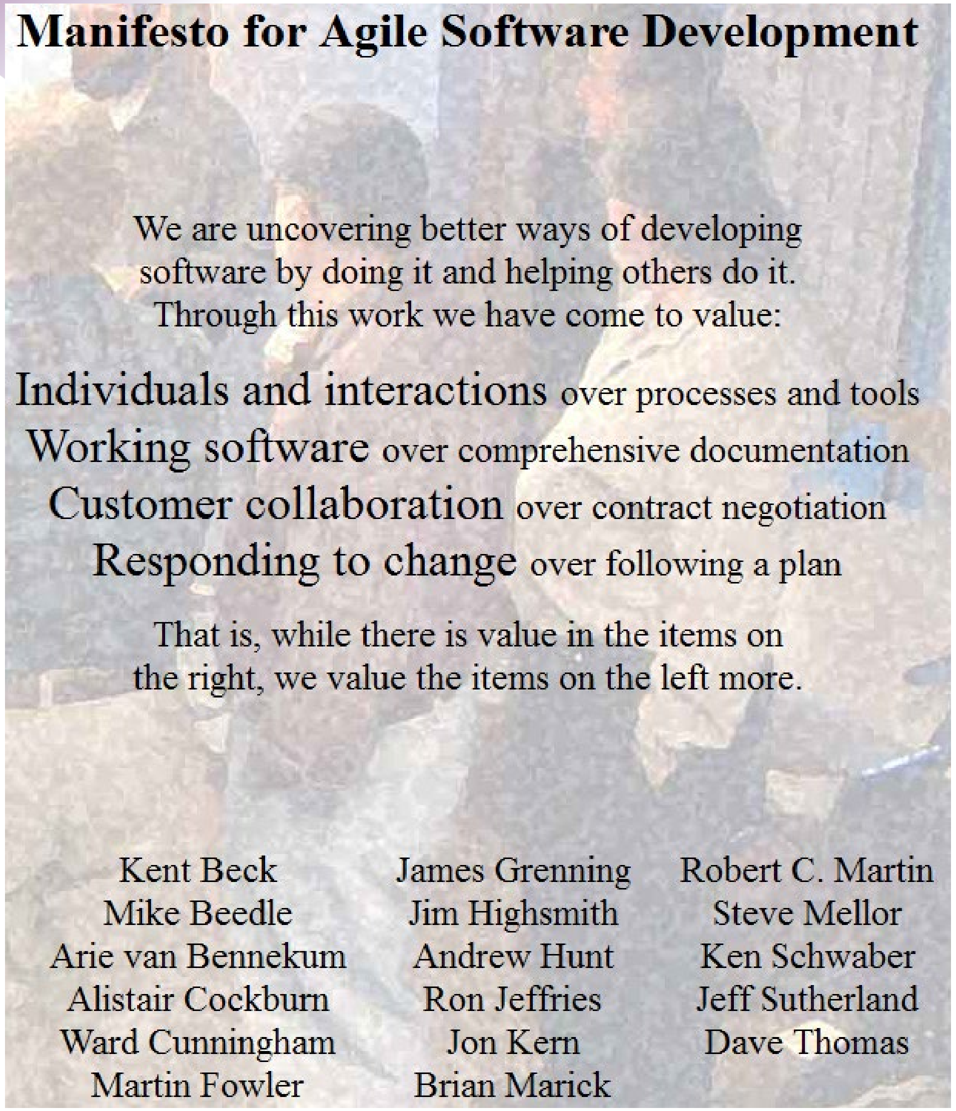
ECSS	NASA
Phase 0 – Mission Analysis/Needs Identification	Pre-Phase A – Concept Studies
Phase A – Feasibility	Phase A – Concept & Technology Development
Phase B – Preliminary Definition	Phase B – Preliminary Design & Technology Completion
Phase C – Detailed Definition	Phase C – Final Design and Fabrication
Phase D – Qualification and Production	Phase D – System Assembly, Integration, Test, Launch
Phase E – Operations/Utilization	Phase E – Operations & Sustainment
Phase F – Disposal	Phase F – Closeout

03

Agile vs. Systems Engineering -

Comparing Apples and Oranges

Agile vs. Systems Engineering



Manifesto for Agile Software Development

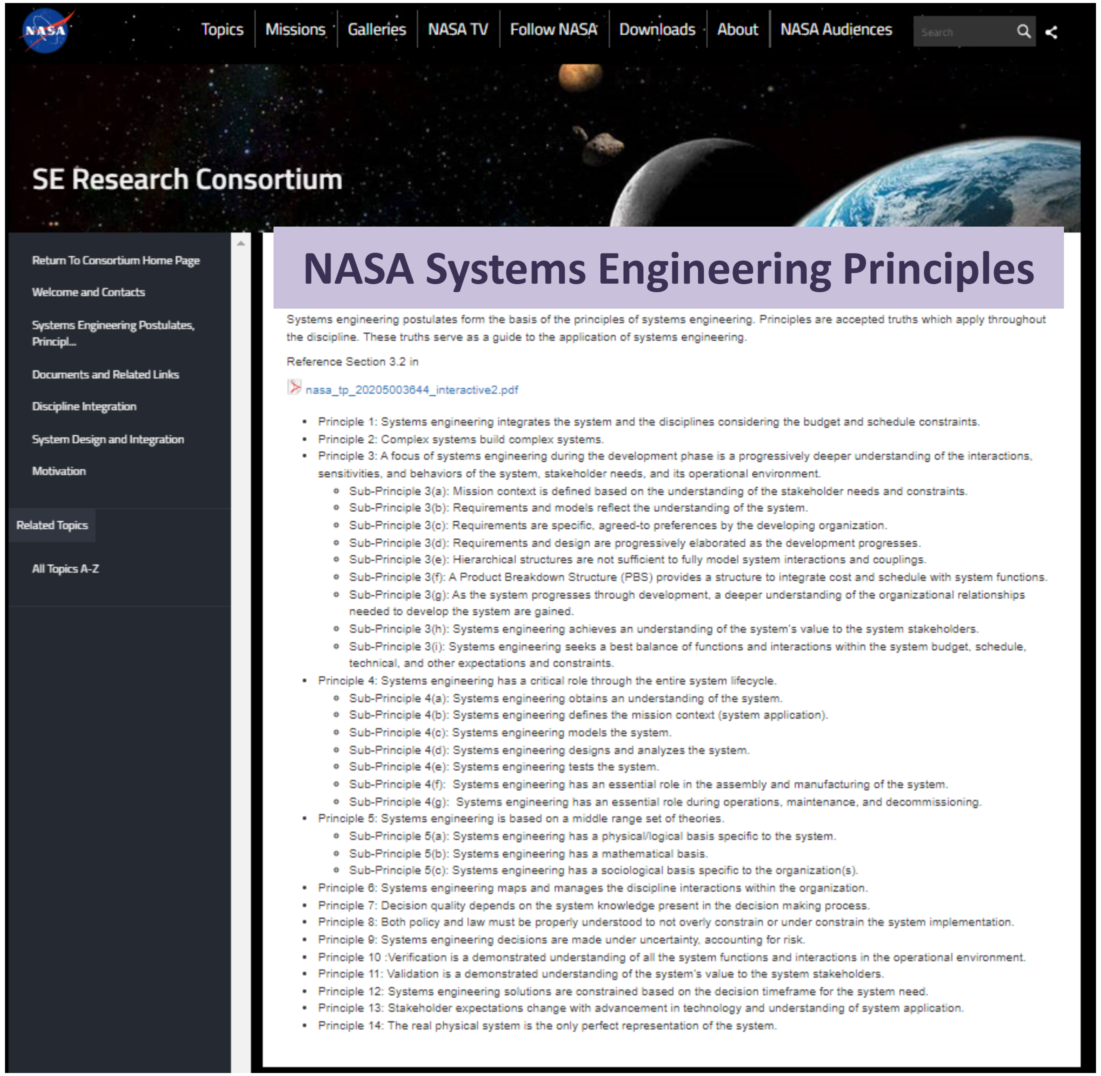
We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

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

[agilemanifesto.org]



Topics | Missions | Galleries | NASA TV | Follow NASA | Downloads | About | NASA Audiences

SE Research Consortium

Return To Consortium Home Page
Welcome and Contacts
Systems Engineering Postulates, Principl...
Documents and Related Links
Discipline Integration
System Design and Integration
Motivation
Related Topics
All Topics A-Z

NASA 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/logical 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.

[www.nasa.gov/consortium/SystemsEngineeringPrinciples]

HSB

AGILE PRINCIPLE 1

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

SYSTEMS ENGINNERING 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.

The customer as one stakeholder is in central focus of the development of the product in both viewpoints.

Agile vs. Systems Engineering

AGILE PRINCIPLE 2

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

SYSTEMS ENGINNERING PRINCIPLE 13

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

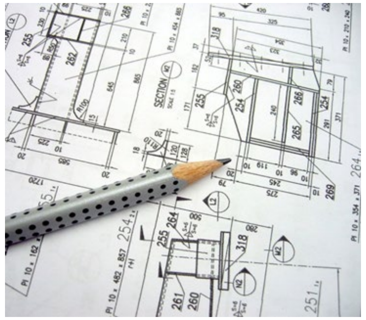
Changing requirements is part of the system evolution.

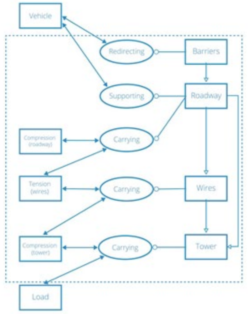
AGILE PRINCIPLE 9

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

SYSTEMS ENGINNERING PRINCIPLE 7

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





Thrive for excellence and good decision by design.

HSB

AGILE PRINCIPLE 10

Simplicity - the art of maximizing the amount of work not done - is essential.

SYSTEMS ENGINNERING PRINCIPLE 2

Complex systems build **complex** systems.

The main driver of useless complexity is the complexity of the organisation, who is building the system. [inspired by Conways Law]

Use Conways Law instead of being haunted by it!

Agile vs. Systems Engineering

HSB

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 ENGINNERING

Systems Engineering methods do **not** define the **time-frame** or **cycle** of development as this is to be defined by the used development process (e.g., agile).

BUT they do provide the definition of **working solution.**

Systems Engineering provides the definition of the working solution and thus enables agile methods.

Agile vs. Systems Engineering

AGILE PRINCIPLE 4

Business people 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 ENGINNERING

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

Communicate, communicate, communicate…

04

Better Agile with

Systems Engineering!

HSB

„TWEAK“ THE AGILE 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

FOUNDATION OF COMPLEX SYSTEMS ENGINNERING

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 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 a rapid moving environment.

Who we are – Working Group Agile SE at GfSE

Who we are – Working Group Agile SE at GfSE

Interested?

Drop us a line:

AGAgileSE@gfse.org

References

- Sheppard und Young (2006) - https://de.abcdef.wiki/wiki/Agility
- Manifesto for Agile Software Development (agilemanifesto.org)
- Agile Systems Engineering (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)
- 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"

Hochschule Bremen
City University of Applied Sciences

Many thanks!

Prof. Dr.-Ing. Jasminka Matevska
jasminka.matevska@hs-bremen.de
Head of Study Programmes
Engineering and Management of Space Systems M.Sc.
Informatik: Software- und Systemtechnik B.Sc.

Hochschule Bremen
Zentrum für Informatik und Medientechnologien (ZIMT)
Fakultät 4 - Elektrotechnik und Informatik
Flughafenallee 10
D- 28199 Bremen

HSB

Prof. Dr.-Ing. Jasminka Matevska

Education

- **10.1987 – 06.1992** Dipl.-Ing. Electrical Engineering (Information and Automation), University of Skopje, Macedonia
- **08.2001 – 07.2009** Ph.D., Dr.-Ing. (Software Engineering, (summa cum laude), University of Oldenburg
- **since 12.2009** – ISTQB Certified Tester
- **10.2013 – 07.2014** Space Systems Engineering Qualification (SEQ), Airbus Defence and Space
- **since 04.2015** – INCOSE CSEP (Certified Systems Engineering Professional)

Professional Experience

- **07.1992 – 04.2009** software development, network administration, responsible software engineering lab engineer, teaching, research (different positions and companies)
- **05.2009 – 02.2016** Team Lead Software Engineering, On-board Software; Software System Engineer, Systems Engineer, Technical Lead and Project Manager for the Columbus Module / International Space Station (ISS); Operations Architect for the European Data Relay Satellite (EDRS)-C at Airbus Defence and Space, Bremen
- **since March 2016** Software and Systems Engineering Professor; Head of Computer Science: Software- and System Engineering B.Sc. and Engineering and Management of Space Systems M.Sc. Study Programmes at the Bremen City University of Applied Sciences

Private: born 1969 in Gostivar, Macedonia, married, two sons (24 and 18), one cat ;-)