

Auslöser für neue Anforderungen an das Software Engineering

J Bosch. 2016. Speed, Data, and Ecosystems: The Future of Software Engineering. Software, IEEE 33, 1: 82–88

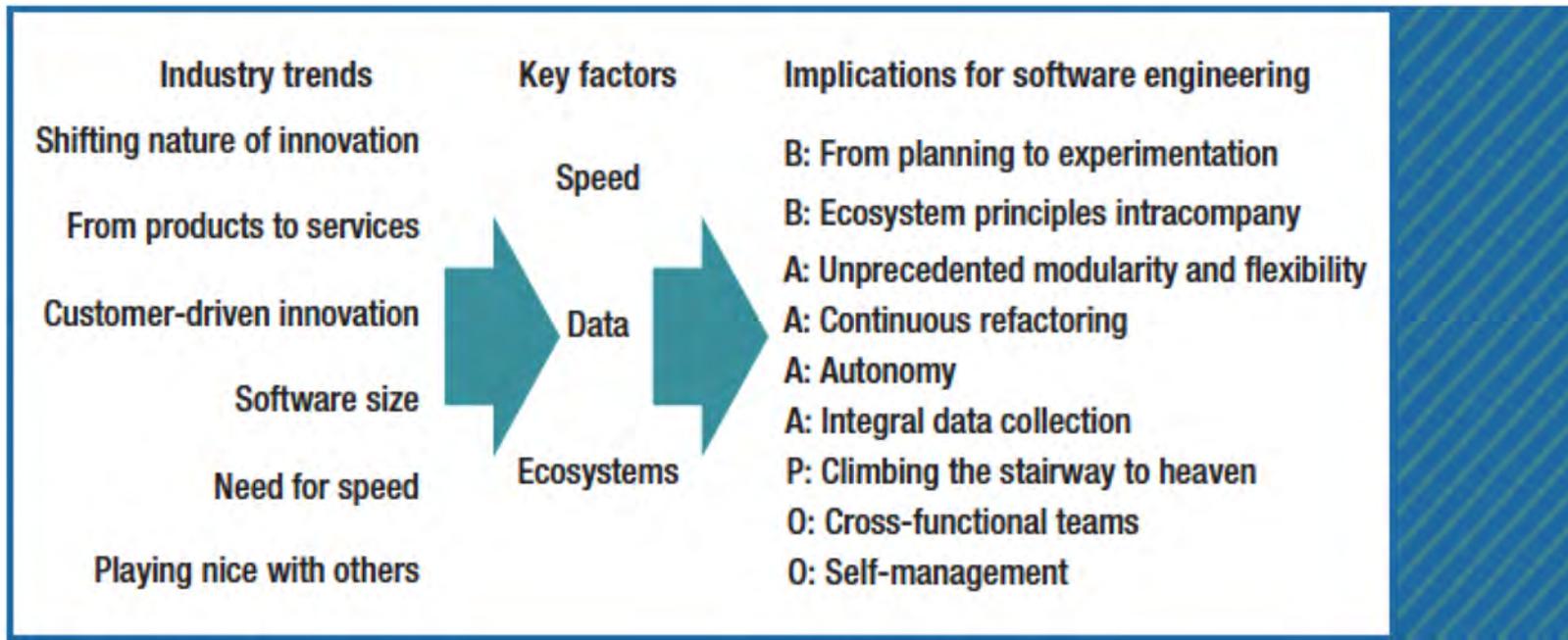


FIGURE 1. The trends and factors influencing software engineering's future, and the implications for business (B), architecture (A), process (P), and organization (O).

Havarievermeidung ist keine Utopie



Solange Software von Menschen gemacht wird, wird es Havarien geben. Denn die Gründe dafür sind vielfältig und vor allem bei soziotechnischen Systemen nicht immer vorhersehbar.



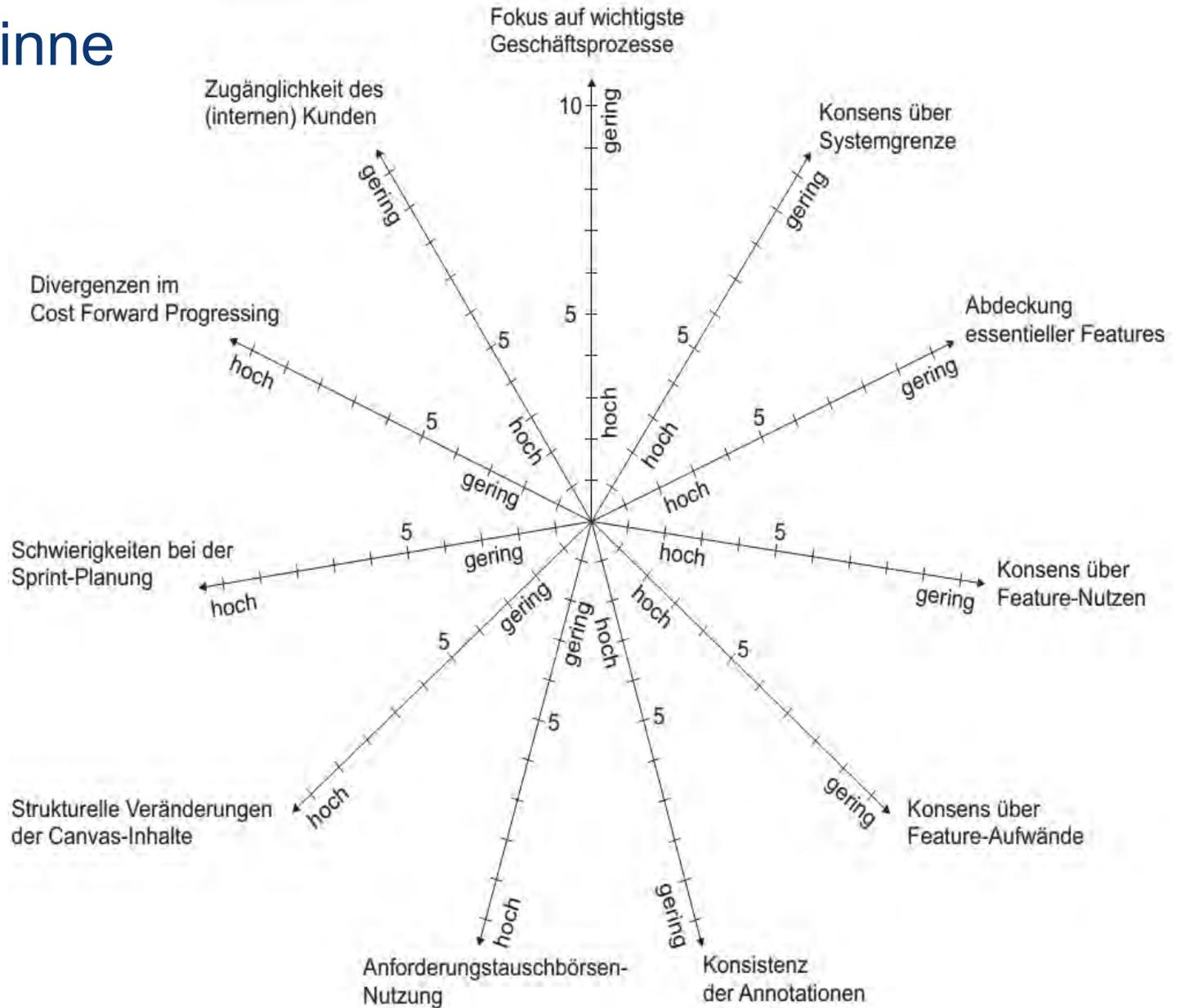
In den letzten Jahren hat die akademische Welt zahlreiche Regelwerke entwickelt, die das Problem von Havarien in Softwareprojekten lösen. Wenn die Praxis sich an die entsprechenden Konzepte hält, wird es auch keine Havarien mehr geben.

Havarievermeidung ist keine Utopie



Havarien lassen sich deutlich reduzieren, aber nur durch Konsequenz und Systematik.

Havariespinne



Gezähmte Agilität ist der Königsweg



Derzeit entwickelt sich eine „IT of 2 Speeds“: Kritische Systeme werden weiterhin traditionell entwickelt, während Unternehmen, alle anderen radikal agil angehen.



Vorgehensmodelle für die Softwareentwicklung machen nach wie vor Sinn. Agilität ist keine grundlegende Veränderung, sie bietet jedoch nützliche Aspekte, die in Vorgehensmodelle integriert werden sollten.



Gezähmte Agilität ist der Königsweg



Agilität in der Softwareentwicklung hat großes Potenzial, das nicht verschenkt werden darf: Es entstehen in kurzer Folge Veränderungsimpulse, die führen zu einer gänzlich neuen Rolle der IT-Abteilungen in Unternehmen.

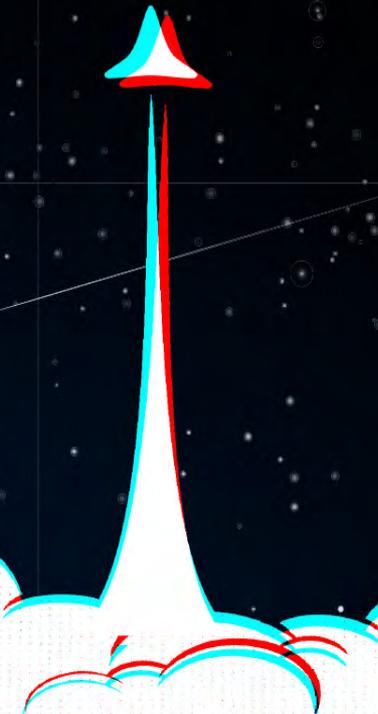
Ungewissheit ist die einzige Gewissheit



Vollständige Vorabbeschreibungen für soziotechnische Systeme (eine organisierte Menge von Menschen und Technologien) kann es nicht geben. Deshalb braucht es Verfahren, um mit Anforderungen umgehen zu können, die erst während der Entwicklung auftauchen.



Die frühen Phasen der Softwareentwicklung bleiben entscheidend. Denn hier kommt es darauf an, einen groben, aber sehr guten Überblick über Anforderungen zu gewinnen. Im Laufe des Projektes können die Verantwortlichen dann gut erkennen, worauf es ankommt.



Ungewissheit ist die einzige Gewissheit



Es lohnt sich, das große Bild ausgiebig zu studieren. Details werden während der Entwicklung ergänzt. Dabei muss immer wieder neu priorisiert und mutig ersetzt werden.

Wertorientierung ist der Maßstab



Es gibt viele Metriken zur Software-Produktivität (angefangen bei Function Points), aber nur wenige Ansätze zur wertorientierten Entwicklung.



Wenn alle wissen, worum es geht, entsteht auch werthaltige – im Sinne des Kunden wertorientierte – Software.



Wertorientierung ist der Maßstab



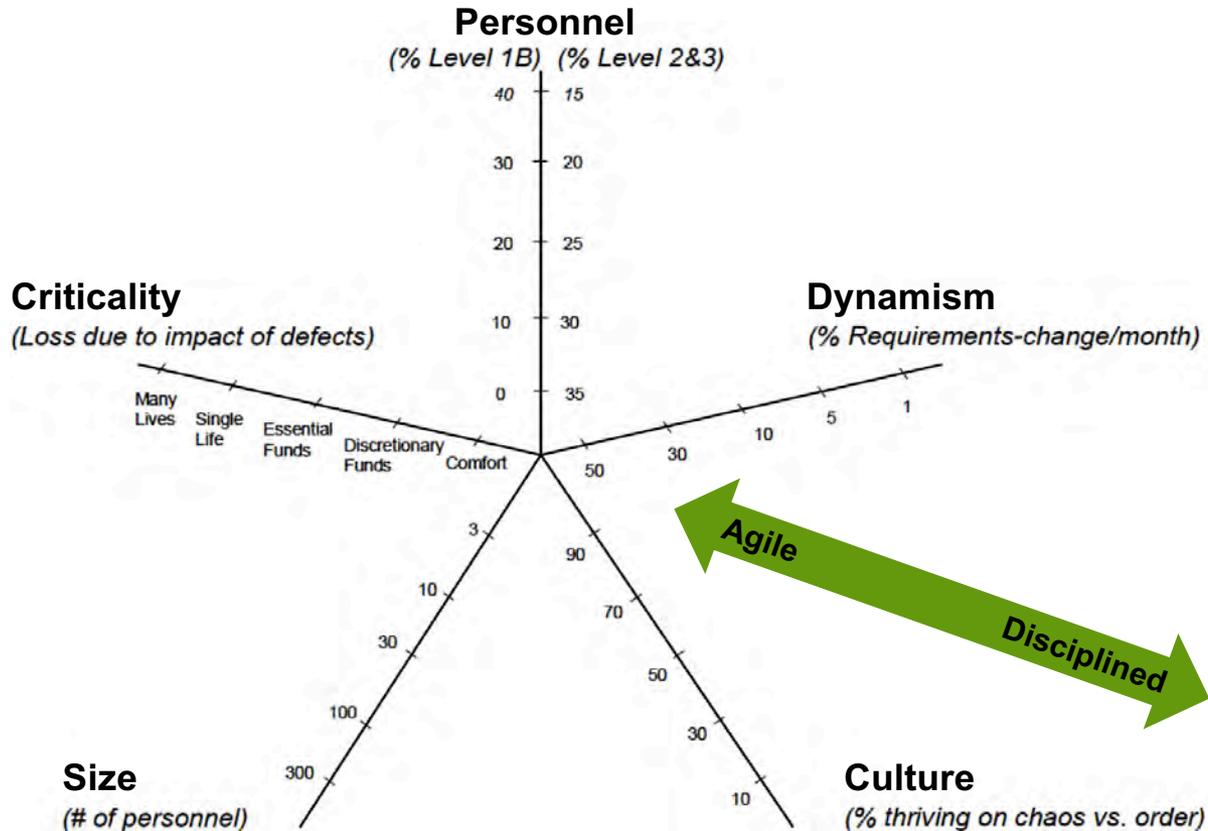
Wertorientierung von Software ist essenziell. In Entwicklungsprojekten muss es deshalb eine systematische, in das Vorgehen integrierte Fokussierung auf Wertorientierung geben.



Agility without discipline cannot scale, and discipline without agility cannot compete.

Mary Poppendieck, Mai 2013

Gezähmte Agilität



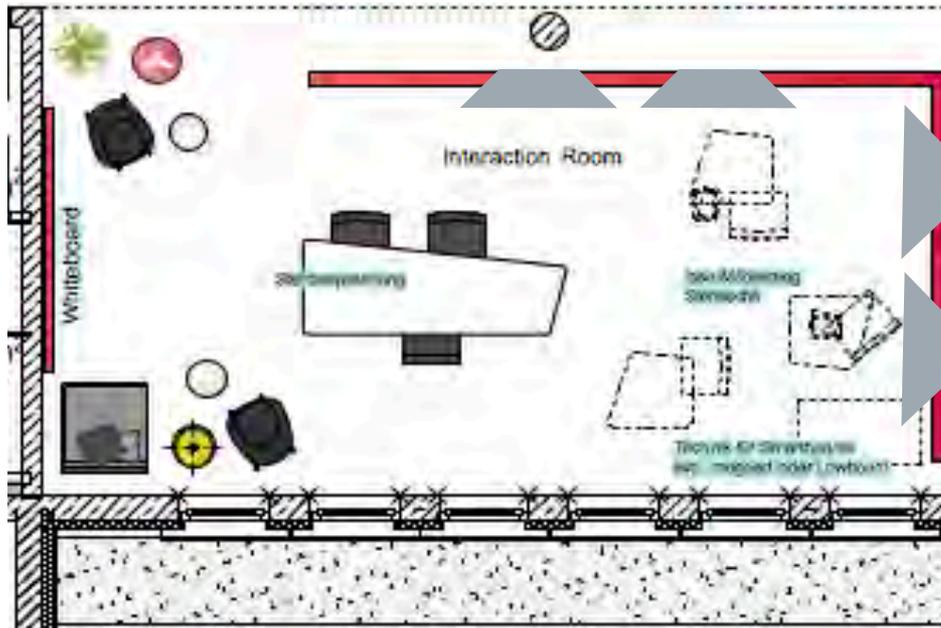
- Zusätzlich: Zustand der Anwendungslandschaft

- Frequenz Innovationen durch Software

- Branche – Unternehmen – Projekt - Teilprojekt

Quelle: Barry Boehm, Agile versus plandriven

Methode: Interaction Room

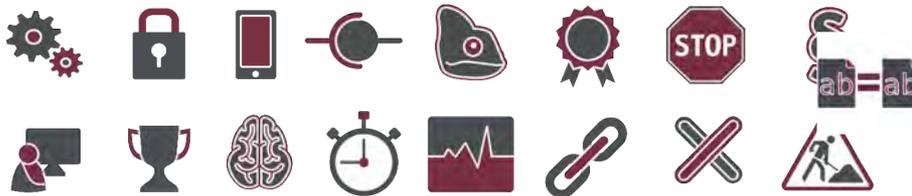


Interaction-Room-Annotationen

- **Werttreiber** spiegeln Kundenbedürfnisse wider, identifizieren Innovation und zeigen positiven Effekt auf die Ziele der Organisation



- **Aufwandstreiber** stellen Qualitätsanforderungen und Rahmenbedingungen dar

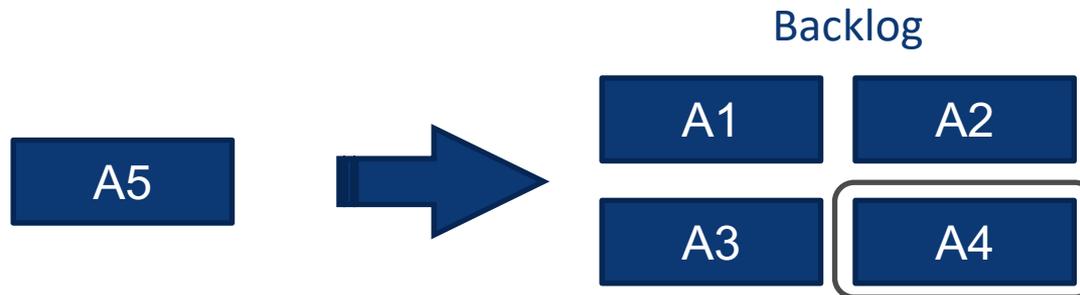


- **Risikotreiber** bergen besonderes Risiko für das Projekt und müssen priorisiert behandelt werden



Anforderungstauschbörse

- Späte Anforderungen sind unvermeidlich, aber allzu einfaches Hinzufügen führt zur Verfettung des Projekts und der entstehenden Software
- Anforderungstauschbörse
 - Verstärkung von konsequentem Tauschen später Anforderungen gegen nicht umgesetzte frühe Anforderungen
 - Regeln für das Tauschen
 - Aufwand muss vergleichbar sein
 - Eigene frühe Anforderung weglassen
 - Konsens unter den Stakeholdern, was weggelassen wird



Microservices sind eine tolle Idee



Architekturparadigmen sind eigentlich stabil. Mal heißt es Modularisierung, mal serviceorientierte Architektur und aktuell halt Microservices.



Microservices sind ein großer Schritt nach vorne. Sie erlauben den Experten, heterogene Architekturen miteinander zu verbinden.



Microservices sind eine tolle Idee



Microservices sind großartig als Integrationsparadigma und für Neuentwicklungen; existierende Landschaften lassen sich aber kaum auf dieser Grundlage umbauen.



Softwaresysteme werden Softwarelandschaften



Komplexe Softwarelandschaften entwickeln sich evolutionär, denn Unternehmen bauen (oder kaufen) jedes einzelne System auf die Art und Weise, wie es für sie in Bezug auf Geschwindigkeit und Wirtschaftlichkeit gerade optimal ist.



Die Experten realisieren komplexe Softwarelandschaften als „Systems of Systems“, das heißt: Ein einheitlicher Architekturstandard sorgt dafür, dass autonome Systeme für bestimmte Anwendungsgebiete miteinander gekoppelt werden.

Softwaresysteme werden Softwarelandschaften



Integrationszenarien sind vorab nur wenig bekannt. Jedes einzelne System muss so gebaut werden, dass es sich zur Integration anbietet und seinen Beitrag in der Systemevolution leistet.

Cognitive Computing is King.



Die Versprechungen der Künstlichen Intelligenz aus den 1980ern lassen sich mit skalierender Rechenleistung nach und nach umsetzen.



Mit großen Datenmengen umzugehen, Muster in solchen Datenmengen zu erkennen und automatische Anpassungen vorzunehmen ist ein wichtiger Treiber der Digitalen Transformation. Wie die Techniken dazu zu kombinieren sind, ist noch nicht immer klar.

Cognitive Computing is King.



Cognitive-Computing-Techniken müssen mit klassischen Informationssystemen integriert werden.