

Hochschule Luzern, Technik & Architektur
CC Distributed Secure Software Systems
Technikumstrasse 21
CH 6048 Horw / Luzern

Lucerne University of
Applied Sciences and Arts

**HOCHSCHULE
LUZERN**

Technik & Architektur



Softwareentwicklung im Maschinenbau ein kooperativer Ansatz

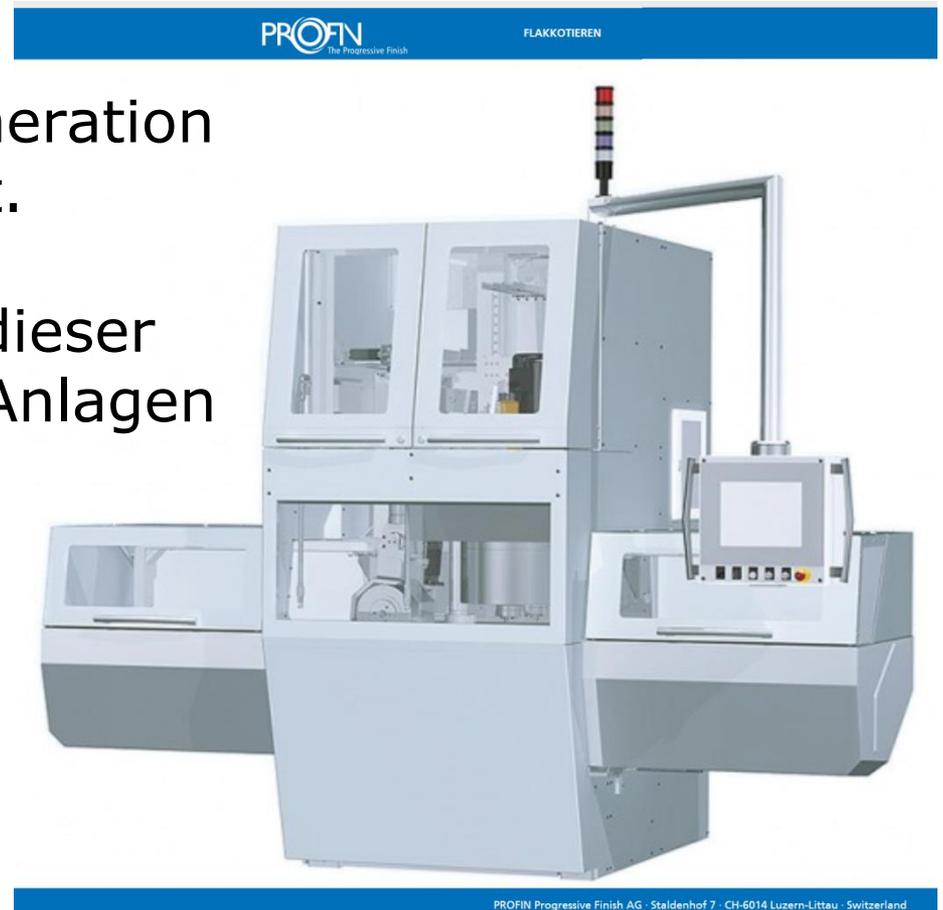
Prof. Martin Jud, Prof. Jörg Hofstetter,
martin.jud@hslu.ch
joerg.hofstetter@hslu.ch
www.hslu.ch/d3s

KMU setzt Millionen in den Sand

Im Anlagenbau wird alle paar Jahre eine neue Generation von Maschinen entwickelt.

In der Folge werden auf dieser Basis kundenspezifische Anlagen gebaut.

Software-Probleme, Projektverzögerungen und Fehler kosten Millionen.



Mut

Josef Vogel hatte den Mut Fehler nicht zuerst bei den andern sondern bei sich zu suchen.

So konnten wir ein KTI^{*)}-Projekt starten.



Über zwei Jahre begleiteten wir die Entwicklungen einer neuen Maschinengeneration. Wir beobachteten die Entwickler und Manager auf beiden Seiten. Wir stellten Fragen, mischten uns aber nicht ein.

^{*)} KTI = Förderagentur für Innovation in der Schweiz

culture clash

Wenn Maschinenbauer und Softwareentwickler zusammenarbeiten, treffen zwei Welten aufeinander.

Wissenslücken:

- Welche Konsequenzen hat eine neue Anforderung?
- Was bewirkt ein Designentscheid?

Vorgehensmodelle:

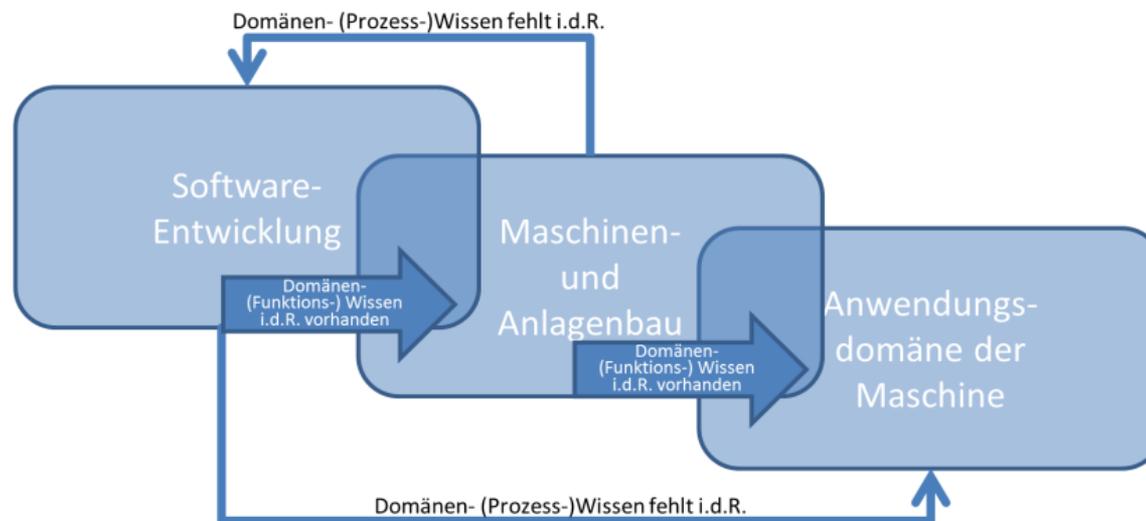
- klassisch-sequenzielles Vorgehen
- inkrementell-agiles Vorgehen

- Wie kann so Vertrauen aufgebaut & erhalten werden?

Wissenslücken

Maschinenbauer verstehen nicht, was SW-Entwickler tun

SW-Entwickler verstehen nicht den Kunden ihres Kunden.



Wissenslücken sind tabu, man arbeitet mit Hypothesen.

Hardware

- Trennung zwischen Konzipieren und Produzieren.



Quelle: Spiesser.de



Quelle: Superpanda.de

- Kosten im Bereich der Produktion höher.
 - Fortschritt sichtbar
- => *sequentielles Vorgehen*
Lieferung zur Abnahme

Software

- Konzipieren und Implementieren, durch die gleiche Person.



Quelle: Resa.de

- Produzieren günstig (Kopieren).
 - Fortschritt nicht sichtbar (keine Validierung ohne HW)
- => *iteratives Vorgehen*
inkrementelle Lieferung

Wer zahlt, befiehlt



Der Maschinenbauer als Auftraggeber legt das Vorgehen fest: klassisch-sequenziell

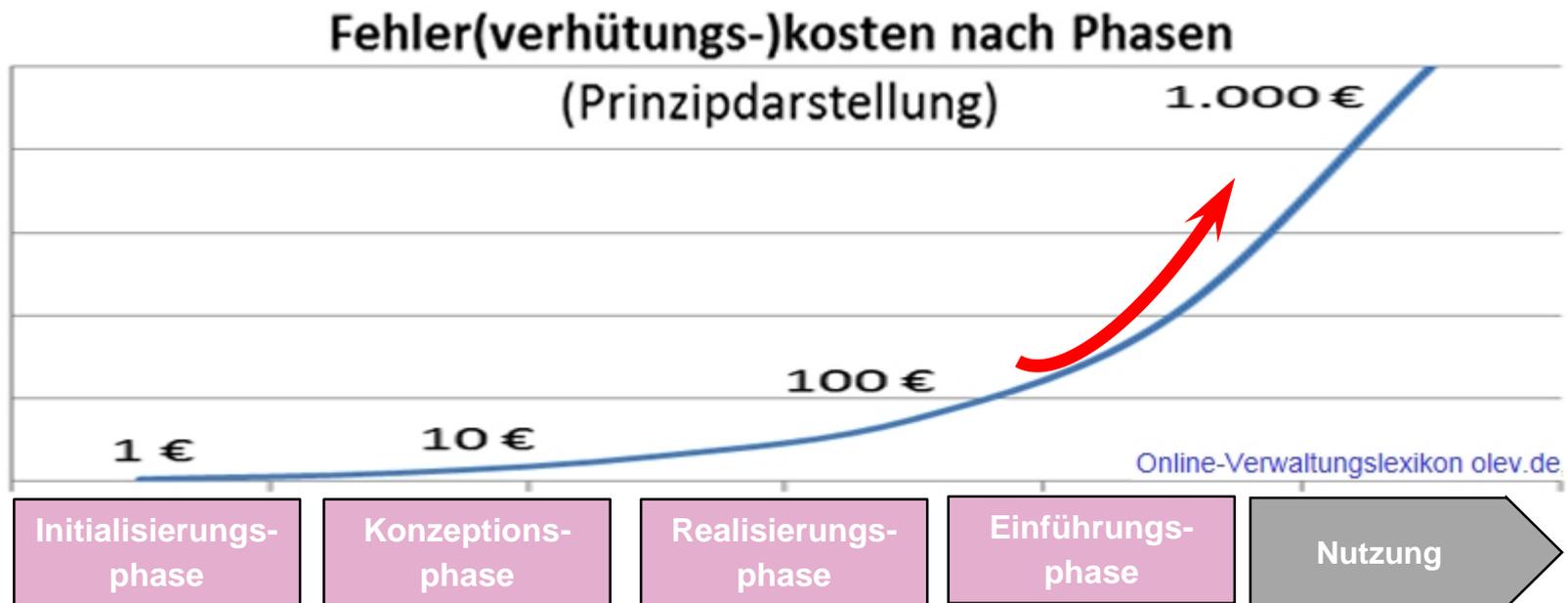
Softwareentwickler führen ein U-Boot Projekt

- Die Spezifikationen sind unklar
- Entwicklung «auf Vorrat» weil keine realitätsnahe Tests möglich sind
- Vorgehen pro forma agil-inkrementell

Controlling deckt Fehlentwicklung zu spät auf
(Sprint-Reviews sind virtuell)

Fehler werden zu spät erkannt

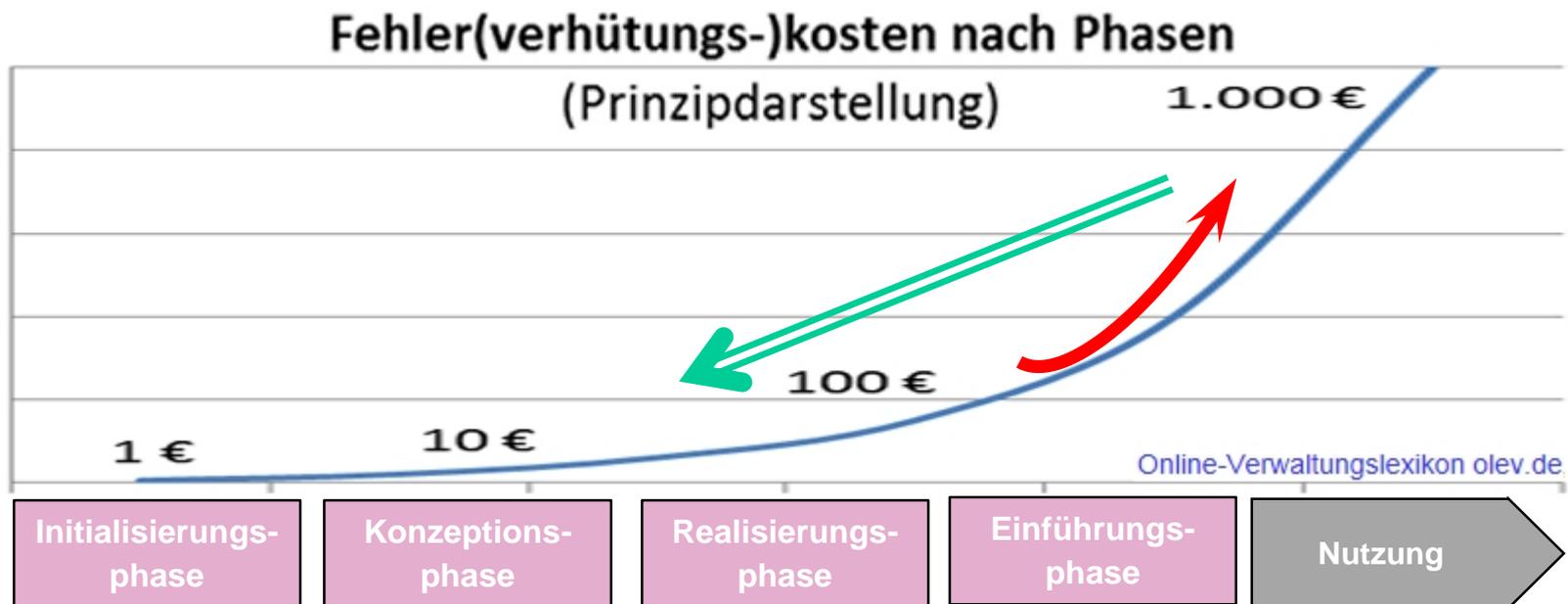
Softwarefehler, Missverständnisse und *implied needs* zeigen sich oft erst bei Inbetriebsetzung, d.h. erst wenn Hard- und Software zusammen kommen.



fail fast

Softwarefehler früher entdecken, reduziert Risiko und Kosten.

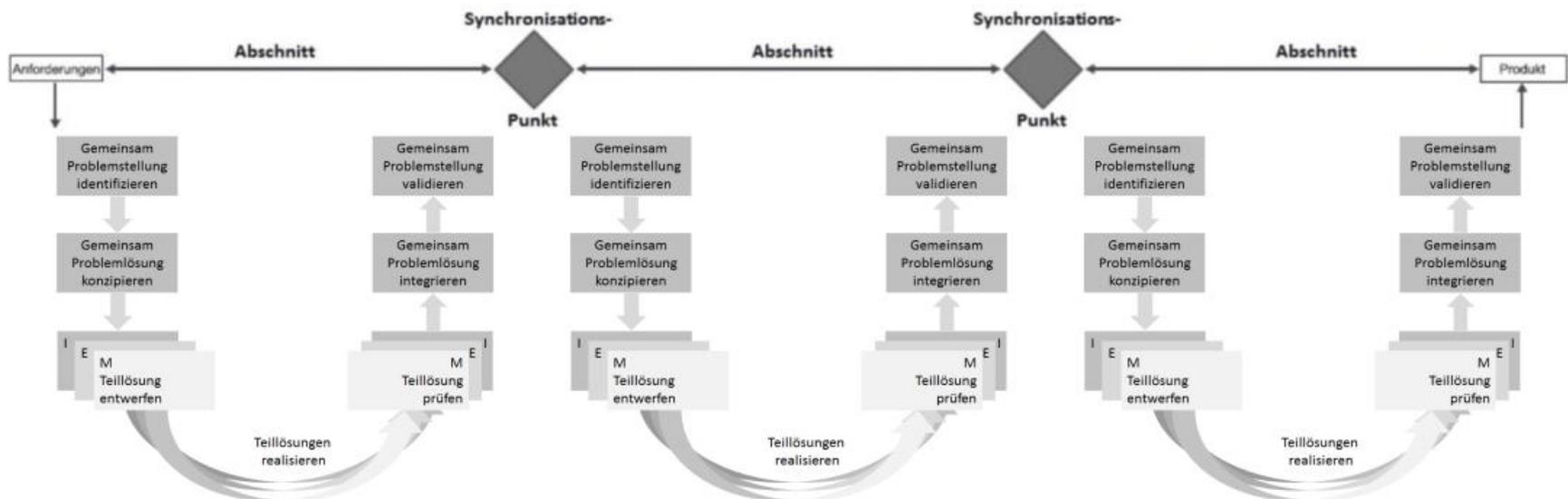
Dazu müssen Hard- und Software früher zusammen kommen.



Hybrides Vorgehen für Maschinenbau & Informatik

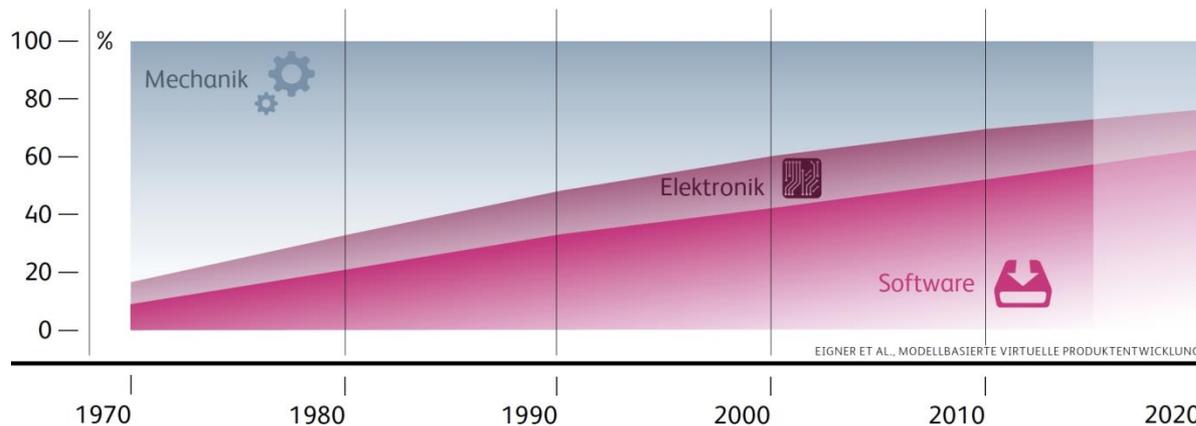
So früh wie irgend möglich Funktionsmuster der Maschine bereitstellen damit Software darauf getestet werden.

Gesamtes Projekt so in Entwicklungsabschnitte gliedern und diese mit HW/SW-Synchronisationspunkten abschliessen.



Kosten / Nutzen des hybrides Vorgehens

Mechanik und Elektronik machen heute nicht einmal mehr die Hälfte der Kosten einer Maschine aus.
- inkrementelle Entwicklung in der Mechanik ist teurer . .



. . aber lohnend, wenn nur schon Software Risiko- und Fehlerkosten im gleichen Umfang gesenkt werden

Publikationen zum Projekt

- **KTI-Projektergebnisse**

auf der HSLU Site:

<https://www.hslu.ch/de-ch/hochschule-luzern/forschung/projekte/detail/?pid=1016>

- SoVeCo Abstract (213,7 KB) .PDF
- **SoVeCo Vorgehen Tools Verträge** (593,2 KB) .PDF
- SoVeCo Anhang Musterverträge (395,6 KB) .PDF

- „**Maschine und Software aus einem Guss**“

Technischen Rundschau Ausgabe 03/2015:

http://www.technische-rundschau.ch/archiv/2015/3/maschine-und-software-aus-einem-guss_45391/

- „**Brückenschlag zwischen zwei Disziplinen**“

HSLU Magazin Ausgabe 02/2015:

<https://www.hslu.ch/de-ch/hochschule-luzern/ueber-uns/medien/magazin/archiv/2015/06/09/soveco/>