REGINA e.V.
Kompetenznetz IT / Informatik Aachen

REGINA Managementtreff
Agile Methoden

Einführung

Stefan Kowalewski, *RWTH, i11*

Überblick und Techniken

Carsten Weise, *embility GmbH*

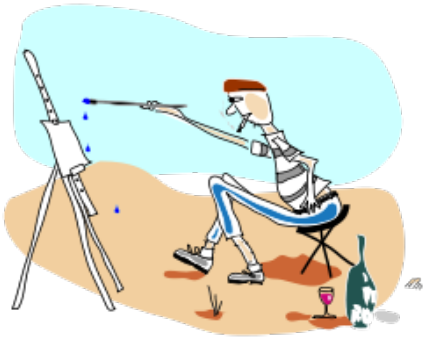
Enterprise Transition, Scaling of Scrum, Tools

Vanessa Fränkl, Jörg Kottig, Raimund Fuchs, *Ericsson*

Historische Perspektive: Vom Künstler zum Bauarbeiter

(vgl. Fowler: The New Methodology)

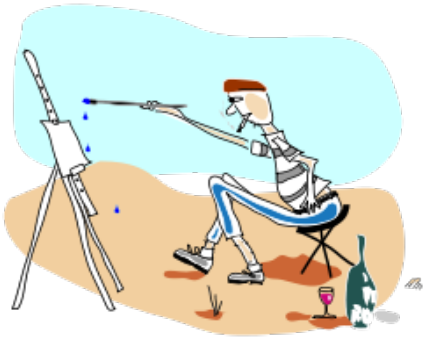
- **Bis Ende der 60er Jahre:**
  - Softwareerstellung als rein kreativer Akt
  - Programmierer = Künstler
  - heute noch: Urheberrecht für Software



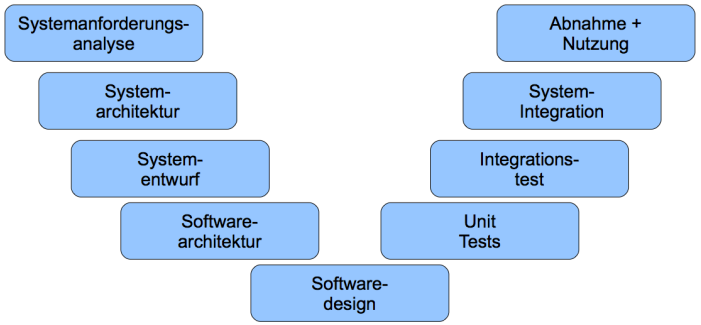
Historische Perspektive: Vom Künstler zum Bauarbeiter

(vgl. Fowler: The New Methodology)

- **Bis Ende der 60er Jahre:**
  - Softwareerstellung als rein kreativer Akt
  - Programmierer = Künstler
  - heute noch: Urheberrecht für Software



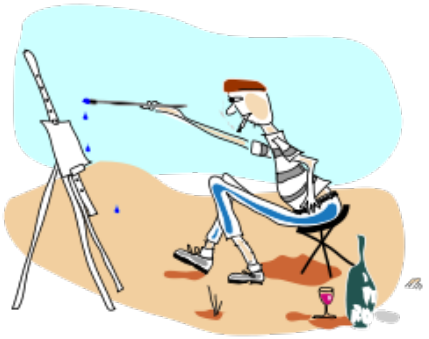
- **Reaktion auf erste Software-Krise: Software *Engineering***
  - Planung, Aufgabenverteilung, Wiederverwendung
  - Software-Konstruktion
  - Vorgehensmodelle: Wasserfall, V-Modell



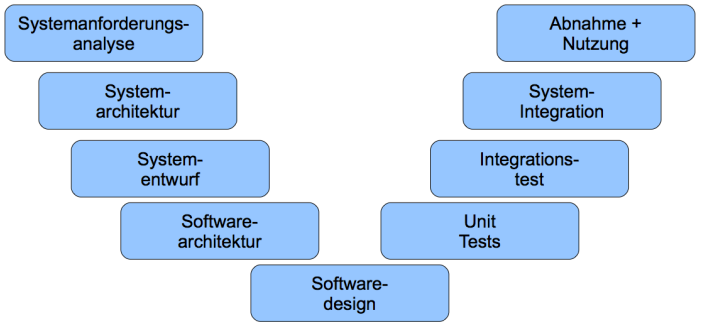
Historische Perspektive: Vom Künstler zum Bauarbeiter

(vgl. Fowler: The New Methodology)

- **Bis Ende der 60er Jahre:**
  - Softwareerstellung als rein kreativer Akt
  - Programmierer = Künstler
  - heute noch: Urheberrecht für Software



- **Reaktion auf erste Software-Krise: Software *Engineering***
  - Planung, Aufgabenverteilung, Wiederverwendung
  - Software-Konstruktion
  - Vorgehensmodelle: Wasserfall, V-Modell

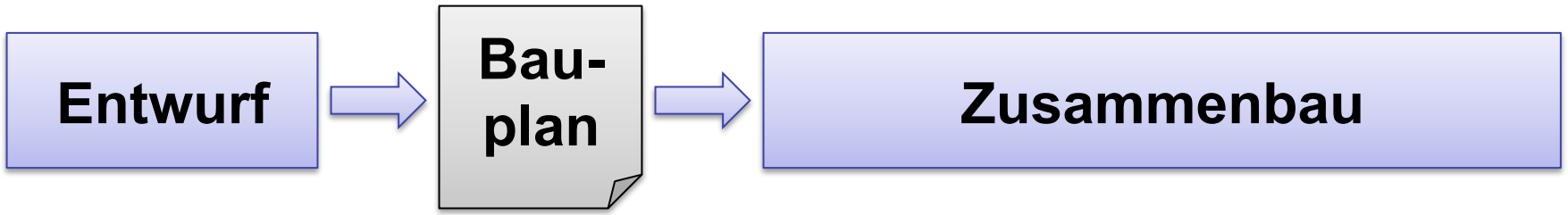


- **Trotzdem scheitert immer noch die Hälfte aller Softwareprojekte.**

Warum?

Häufig falsche Annahme: Softwareentwicklung ist planbar

- **Analogie Ingenieurprojekt, z.B. Gebäudebau, Fertigung**



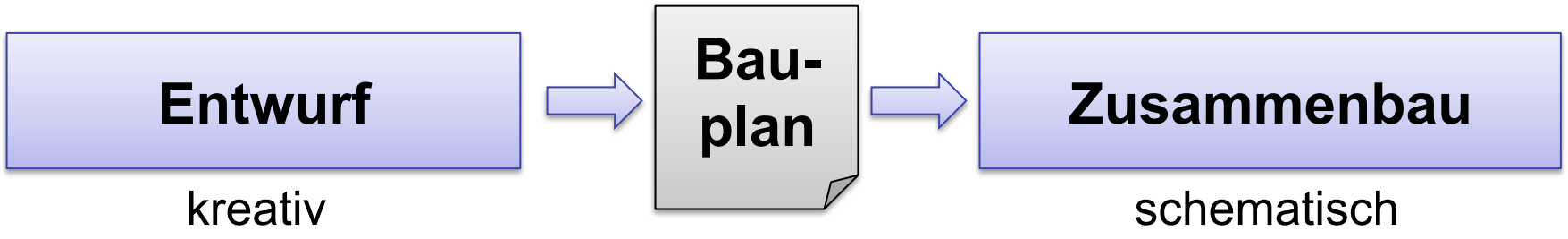
- kreative Tätigkeit
- schlecht planbar
- hochwertige Ausbildung
- Individuell wichtige Mitarbeiter

- schematische Tätigkeit
- gut planbar
- einfache Ausbildung
- austauschbare Mitarbeiter

- **Frederick Taylor (1856-1915): Vollständig geplante Arbeitsabläufe (Taylorismus)**

Häufig falsche Annahme: Softwareentwicklung ist planbar

- **Analogie-Idee beim Engineering-Ansatz für Software**



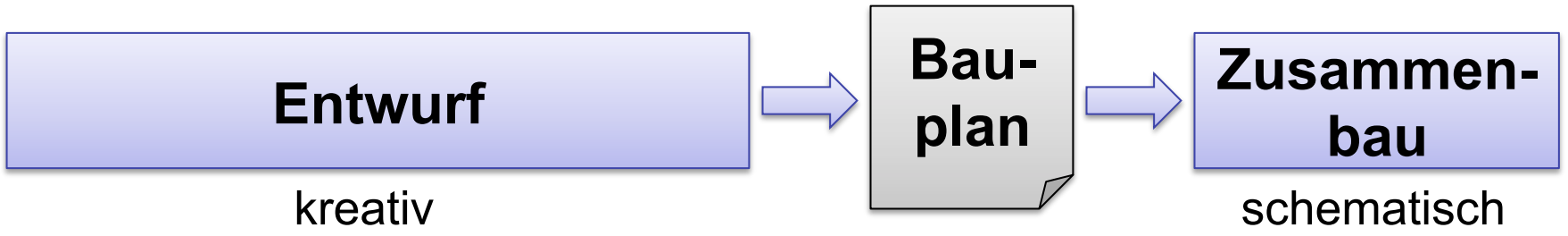
Entwurf, Modellierung

Architektur, Modell

Codierung

Häufig falsche Annahme: Softwareentwicklung ist planbar

- **Häufig sieht die Analogie aber so aus:**



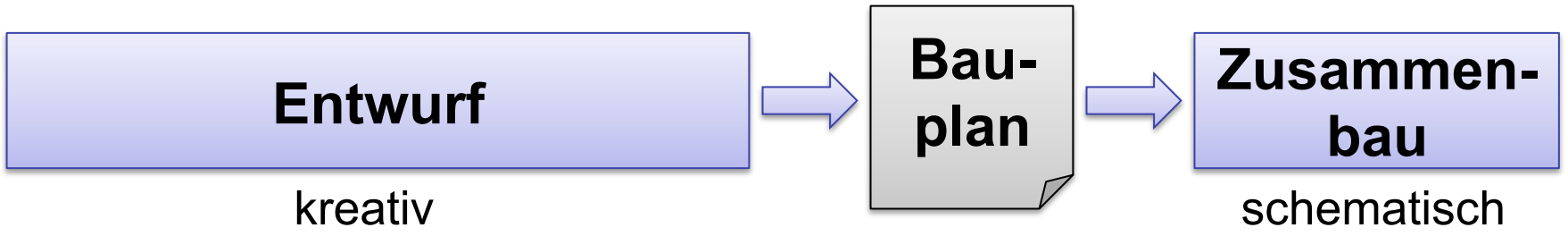
Entwurf, Modellierung, Codierung

Code

Compilierung

Häufig falsche Annahme: Softwareentwicklung ist planbar

- **Häufig sieht die Analogie aber so aus:**



Entwurf, Modellierung, Codierung

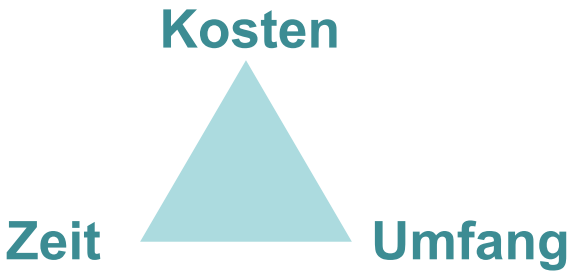
Code

Compilierung

- **Konsequenzen?**

Konsequenzen

- Mangelnde Planbarkeit
  - Häufigere Iterationen
  - Fähigkeit, auf Änderungen zu reagieren (Adaptivität)
  - Engerer **Kunde**nkontakt
  - Anpassung der Zielgrößen:



- Keine austauschbaren **Mitarbeiter**
  - Respektvoller Umgang
  - Behandlung als Individuum
  - Anpassung der Prozesse und Arbeitsumgebung

➔ **Neue Werte ➔ Prinzipien ➔ Techniken ➔ Methoden**