Klassische und agile Vorgehensmodelle – Ein historischer Überblick

Prof. Dr. Ralf Kneuper

Beratung für Softwarequalitätsmanagement und Prozessverbesserung

Ralf Kneuper

- Dipl.-Mathematiker, Univ. Bonn
- PhD Computing Science, Univ. of Manchester
- 1989-1995: Software AG
  - Qualitätssicherung, Qualitätsmanagement, ISO 9000
- 1995-2005: Deutsche Bahn/TLC/DB Systems
  - Seniorberater, Projektleiter
  - Qualitätsmanagement, interner CMM(I)-Berater für Entwicklungsprozesse und Projektmanagement
- Seit 2003: freiberuflicher Berater für Qualitätsmanagement, insbesondere CMMI
- Seit 2012: Prof. für Wirtschaftsinformatik an der Internationalen Berufsakademie in Darmstadt
- Ehemaliger Sprecher der GI-FG Vorgehensmodelle
- SEI-zertifizierter SCAMPI Lead Appraiser für CMMI-DEV und CMMI-SVC
- Koordinator des German CMM(I) Lead Appraiser and Instructor Board (CLIB)

- **Beratungsthemen:**
  - CMMI
    - für Entwicklung, CMMI-DEV, und für Dienstleistungen, CMMI-SVC
    - CMMI Institute-zertifizierter SCAMPI Lead Appraiser
    - Autor mehrerer Fachbücher zu CMMI
  - Messung und Bewertung von Prozessqualität
    - Prozessqualitätsmodell Gokyo Ri
  - Vorgehensmodelle, z.B. V-Modell XT
    - Unterstützung beim Einsatz auf Auftraggeber- und Auftragnehmerseite
    - zertifizierter V-Modell XT Prozessingenieur (PING)
  - Reviews
- **Kontakt:**
  - http://www.kneuper.de
  - ralf@kneuper.de

Agenda

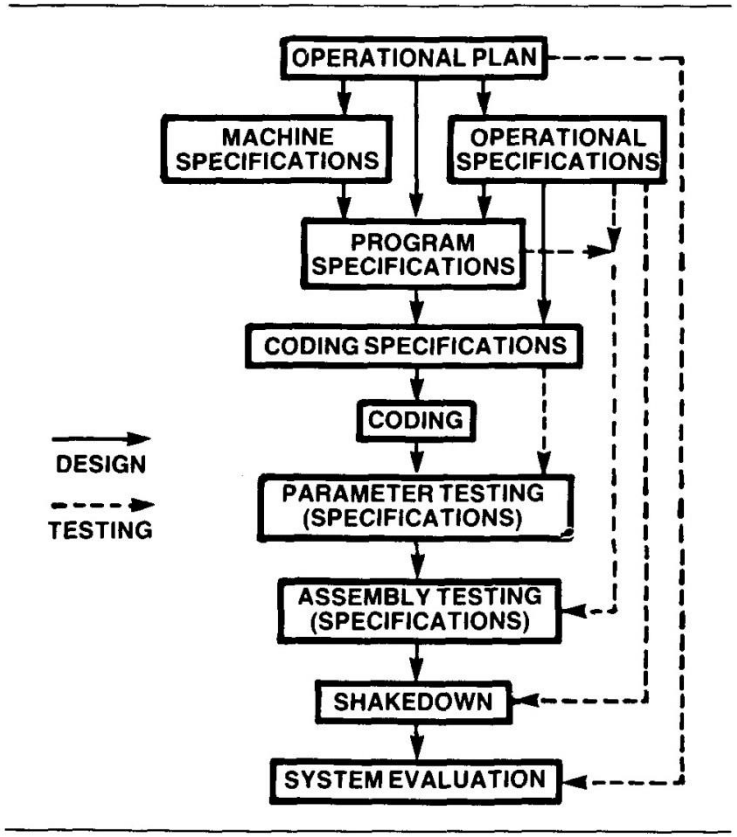
- Die „Frühzeit“ der Vorgehensmodelle
- Mitte der achtziger Jahre: Beginnende systematische Betrachtung der Softwareprozesse
- Mitte / Ende der neunziger Jahre: Leichtgewichtige und agile Prozesse
- 2000 bis 2010: Die Zeit der Religionskriege
- Seit ca. 2005: Allmähliche Annäherung

Die „Frühzeit“ der Vorgehensmodelle

- **Turing 1951:**
  - Make a plan
  - Break the problem down
  - Do the programming of the new subroutines
  - Programme the main routine

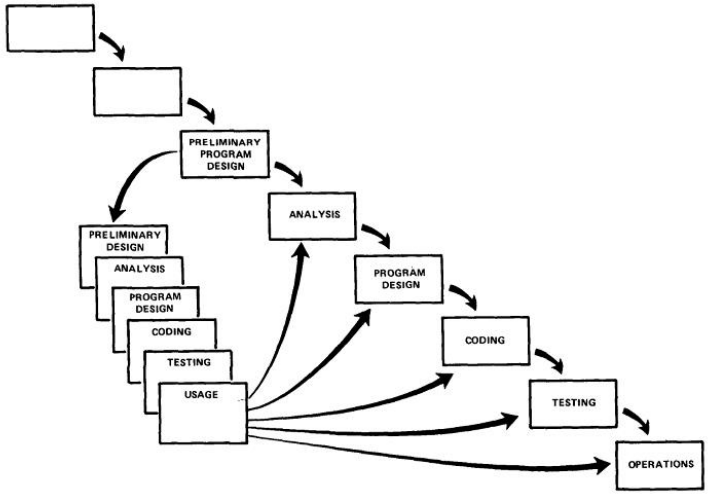
Die „Frühzeit“ der Vorgehensmodelle

- **Benington 1956**
  - Erstes Vorgehensmodell im heutigen Sinne
  - Damalige Rahmenbedingungen beeinflussen Vorgehen:
    - Computer (Speicher, Prozessor) sind teuer, Programmierer sind billig
    - Maschinensprache („Coding“ übersetzt Flussdiagramm in Maschinensprache)
  - Linearer Ablauf
    - aber Nutzung eines Prototyps



Die „Frühzeit“ der Vorgehensmodelle

- **Royce 1970**
  - Oft als „die“ Referenz für Wasserfallmodell dargestellt
  - Betont allerdings Notwendigkeit von Rücksprüngen und Prototypen



Mitte der achtziger Jahre: Beginnende systematische Betrachtung der Softwareprozesse

- **Beginnende Fokussierung auf Prozesse nicht nur in der SW-Entwicklung**
  - ISO 9000-Reihe 1987
  - CMM (Humphrey 1989),V-Modell (Hummel 1990)
  - Tagungsreihe International Software Process Workshop startet 1984
  - Werkzeugunterstützung durch Process-Centered Software Engineering Environments (PCSEE), Integrated Project Support Environments (IPSE) etc.
  - Vorgehensmodelle meist aus dem Umfeld großer, komplexer Systeme, oft militärisch
    - umfangreiche, “schwergewichtige” Modelle
  - Auch in der damals aufkommenden OO-Entwicklung, z.B. RUP
  - Osterweil 1987: “Software Processes are Software Too”
    - “Programmierung” von Softwareprozessen

Iteratives / inkrementelles / evolutionäres Vorgehen

- **In einfacher Form (Prototyping / wenige Iterationen) bereits sehr früh (Benington 1956, Weinberg 1957, Zurcher/Randell 1968, Royce 1970)**
- **Aufgrund der Kostenverteilung anfangs noch sehr eingeschränkt möglich**
  - Barry Boehms erster Arbeitstag: „Now listen. We are paying $600 an hour for this computer and $2 an hour for you, and I want you to act accordingly.”
- **In den achtziger Jahren ausgebaut**
  - Prototyping, Boehms Spiralmodell (1986), Rapid Application Development (1991)

Mitte / Ende der neunziger Jahre: Leichtgewichtige und agile Prozesse

- **Mit der wachsenden Fokussierung auf Prozesse auch Gegenbewegung**
  - (SW-)Entwicklung als kreative Wissensarbeit
  - Z.B. Takeuchi, Nonaka (Wissensmanagement, bekannt für SEKI-Modell) 1986: Scrum als Ansatz zur Produktentwicklung
    - Weiterentwickelt und auf SW-Entwicklung konkretisiert durch Schwaber, Sutherland 1995
  - Tom Peters 1987: „Thriving on Chaos“
- **Verstärkt Mitte / Ende der neunziger Jahre**
  - Scrum 1995
  - Extreme Programming XP 1999
  - Crystal, FDD etc.

Mitte / Ende der neunziger Jahre: Leichtgewichtige und agile Prozesse

- **Mitte / Ende der neunziger Jahre: Entwicklung bzw. Ausbau der heute als „agil“ bezeichneten Ideen / Werte / Prinzipien**
  - Keine „Big Upfront“-Schritte
  - Viele kurze Zyklen mit nutzbarem Ergebnis → Weiterentwicklung der iterativen Ansätze
  - Selbst-Organisation
  - Änderungen (insbesondere der Anforderungen) als normal akzeptieren
  - Qualität durch schnelle Rückmeldungen (aus automatisierten Tests, vom Kunden etc.)
- **2001: Agiles Manifest**
  - Führte den Begriff „agil“ ein, der „leichtgewichtig“ etc. ablöste

2000 bis 2010: Die Zeit der „Religionskriege“

- **„Hardcore-Agilisten“:**
  - Agile Entwicklung löst alle Probleme
  - Plan-getriebene Entwicklung ist ein Dinosaurier und wird aussterben
  - Pseudo-Agilisten: Endlich haben wir eine gute Begründung, warum wir nichts zu dokumentieren brauchen
- **„Hardcore-konventionelle Entwickler“:**
  - Agile Entwicklung ist Spielerei, unprofessionell
  - „Man weiß nicht, was herauskommt“

Seit ca. 2005: Allmähliche Annäherung

- **Erkenntnis:**
  - Plan-getriebene und agile Entwicklung haben das gleiche Ziel (hohe Qualität in der Zeit, im Budget), aber unterschiedliche Ansätze, diese zu erreichen
  - je nach Rahmenbedingungen unterschiedlich gut geeignet
  - können sich gut ergänzen
- **Boehm, Turner 2003: „Balancing Agility and Discipline“**
  - Es gibt viele Grautöne zwischen Plan-getriebener und agiler Entwicklung
  - Benötigt wird eine Kombination von beiden Ansätzen, je nach Rahmenbedingungen mehr von dem einen oder mehr von dem anderen
- **Meyer 2014: „Agile! The Good, the Hype and the Ugly“**
  - Good: Neue, hilfreiche Konzepte der agilen Entwicklung
  - Hype: Gab es vorher schon, wird von agiler Entwicklung „hochgejubelt“
  - Ugly: Neue Konzepte, die eher schaden

Fragen