





www.uni-frankfurt.de

Modul: Programmierung B-PRG

Grundlagen der Programmierung 1

V19 Vorgehensmodelle und Prozessmodelle im SWE
→ Eine Vorlesung in PS 2



Prof. Dr. Detlef Krömker
Professur für Graphische Datenverarbeitung
Institut für Informatik
Fachbereich Informatik und Mathematik (12)



Übersicht

- Begriffsklärung: Projekt, Vorgehen, Prozess
- Die Entwicklung der Vorgehens- und Prozessmodelle in den letzten 60 Jahren.
 - Phasenorientierte Modelle: (Wasserfall ++)
 - Agile (= leichtgewichtige) Modelle, z.B. Scrum

2 Vorlesung PRG 1
Vorgehens- und Prozessmodelle Prof. Dr. Detlef Krömker

- Ralf Kneuper, "Sixty Years of Software Development Life Cycle Models", *IEEE Annals of the History of Computing*, vol. 39, no. , pp. 41-54, July-September 2017.
- READING: http://www.kneuper.de/Publikationen/Rundbrief-Geschichte_der_VGM.pdf
- READING: siehe Moodle-Plattform

3
Vorlesung PRG 1
Vorgehens- und Prozessmodelle
Prof. Dr. Detlef Krömker






Begriff "Vorgehensmodelle" im SWE

- *allgemein organisieren einen Prozess in verschiedene, strukturierte Abschnitte, denen wiederum entsprechende Methoden und Techniken der Organisation zugeordnet sind.*
- **Aufgabe** eines Vorgehensmodells ist es, die allgemein in einem Gestaltungsprozess auftretenden Aufgabenstellungen und Aktivitäten in einer sinnfälligen logischen Ordnung darzustellen.
- Mit ihren Festlegungen **sind** Vorgehensmodelle **organisatorische Hilfsmittel**, die für konkrete Aufgabenstellungen (Projekte) individuell angepasst werden können und sollen ...

aus Wikipedia: <https://de.wikipedia.org/wiki/Vorgehensmodell>

- Sind Teil des Projektmanagements.

4
Vorlesung PRG 1
Vorgehens- und Prozessmodelle
Prof. Dr. Detlef Krömker

Vorgehens- und Prozessmodelle -- Projekte

Wahl der Programmiersprache + einige Regeln (Style Guide) reichen für einzelne Software-Entwickler durchaus aus:

Programmieren im Kleinen (*programming in the small*)

Aber: In größeren Projekten (*programming in the large*) klappt die **Abstimmung und die Zusammenarbeit keineswegs** von selbst!

- Einigungen über **Vorgehen und Prozesse** und
- fast alle Software-Entwicklungen werden in **Projekten** durchgeführt.
- Was ist das alles genau?

5
Vorlesung PRG 1
Vorgehens- und Prozessmodelle
Prof. Dr. Detlef Krömker






Merkmale eines Projekts

Ein **Projekt** ist ein Vorhaben zur Lösung einer Aufgabe und ist im Einzelnen durch folgende Merkmale gekennzeichnet :

- Das **Projektziel** ist eindeutig definiert.
- Das Projektziel und das Projektergebnis sind **messbar** beschrieben.
- Das Projekt wird innerhalb eines **Zeitraumens** mit vorgegebenen **Terminen** durchgeführt.
- An einem Projekt arbeiten notwendigerweise **mehrere Beteiligte** zusammen.
- Voneinander abhängige **Termin-, Ressourcen- und Kostenvorgaben** wirken begrenzend.
- Ein Projekt ist **einmalig** (wird nicht alle Tage durchgeführt).
- Aus der Vielzahl der Teilaufgaben kann sich eine hohe **Komplexität** ergeben.
- Ein Projekt muss genau **geplant** werden.
- Ein Projekt benötigt eine **Projektorganisation**

Definition „Projekt“ nach DIN 69901: „Ein Vorhaben, das im wesentlichen durch **Einmaligkeit** der Bedingungen in ihrer Gesamtheit gekennzeichnet ist, wie z.B. Zielvorgaben, zeitliche, finanzielle, personelle oder andere Begrenzungen, Abgrenzung gegenüber anderen Vorhaben, projektspezifische Organisation.“

6
Vorlesung PRG 1
Vorgehens- und Prozessmodelle
Prof. Dr. Detlef Krömker





Projektziele



- Projektziele werden **schriftlich** fixiert:
- D.h. sicherstellen, dass Auftraggeber und Auftragnehmer gleiche Vorstellungen verfolgen, dass es keine Missverständnisse gibt und dass die Projektfortschritte jederzeit an den Zielen gemessen werden kann.
- Dazu ist es wichtig, dass die Ziele **objektiv messbar** und klar formuliert werden.

Im einzelnen hat die Formulierung von Projektzielen folgende Funktionen:

- Präzisierung
- Orientierung
- Motivation
- Kontrolle



7
Vorlesung PRG 1
Vorgehens- und Prozessmodelle
Prof. Dr. Detlef Krömker

Projektmanagement (PM) im weiteren Informatik-Studium

Weitergehende Veranstaltungen zum PM:

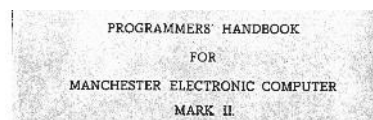
- Programmierpraktikum (3. oder 4. Semester im BSc Informatik)
Hauptthema: "Programmieren im Team"
- Einführung in das IT-Projektmanagement (2V+1Ü) 3 CP
- Prinzipien des IT-Projektmanagements (eLearning Kurs) 1 CP
- Praktikum zur Wirtschaftsinformatik (4 Pr) 8 CP
+ diverse weitere Praktika

8
Vorlesung PRG 1
Vorgehens- und Prozessmodelle
Prof. Dr. Detlef Krömker

Vorgehens- und Prozessmodelle

- "Prozess" beinhaltet mehr Aspekte als "Vorgehen"
- **Prozessmodell =**
 - **Vorgehensmodell (Kern)**
 - + Organisationsstrukturen
 - + Vorgaben für das Projektmanagement
 - + Vorgaben für die Qualitätssicherung
 - + Vorgaben für die Dokumentation
 - + Vorgaben für die Konfigurationsverwaltung
- aber im üblichen Sprachgebrauch wird i.A. nicht so streng unterschieden, so dass auch Prozessmodelle wie XP, Scrum, Unified Prozess oft auch als "Vorgehensmodelle" eingeordnet sind.

Vorgehensmodelle – Die sehr frühen Ansätze – (Turing, 1951)



by Alan Turing, 1951

Siehe: <http://www.alanturing.net/programmers-handbook/>

<http://de.dwhwiki.info/konzepte/cowboy-coding>

Das "Rezept":

*"Make a plan
Break the problem down
Do the programming of the new
subroutines
Programme the main routine"*

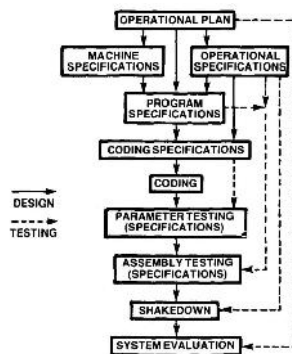
Tatsächlich:

Die Projekte waren oft sehr überschaubar:

- **Ein** Entwickler
- Der Entwickler war oft auch der Nutzer

→ **Cowboy Coding, Code-and-fix**

Die sehr frühen Ansätze (Benington, 1956)



Herbert D. Benington (1956): Production of Large Computer Programs in Proc. Symp. On Advanced Computer Programs for Digital Computers, June 1956. Republished in Annals of History in Computing, Oct. 1983, pp.350-361.

- Das "erste" Phasenmodell (*stagewise model*)
- Beachte die Hardware-Abhängigkeit.
- "möglicherweise hilfreich"

Achtung: leicht **missverständlich** sind die Phasen „Coding Specification“ und „Coding“, da sich die Bedeutung dieser Begriffe seither deutlich geändert hat. Die „**Coding Specification**“ entspricht etwa der heutigen Programmierung in einer der üblichen Programmiersprachen, während „**Coding**“ die Übersetzung der Coding Specification in Maschinencode umfasst, also eine Aktivität, die heute normalerweise von einem Compiler durchgeführt wird und keine eigene Entwicklungsphase darstellt.

11

Vorlesung PRG 1
Vorgehens- und Prozessmodelle

Prof. Dr. Detlef Krömer

Das (klassische) Wasserfall-Modell (Royce 1970)



Figure 1. Implementation steps to develop a small computer program for internal use.

Winston W. Royce: MANAGING THE DEVELOPMENT OF LARGE SOFTWARE SYSTEMS. Reprinted from Proceedings, IEEE WESCON, August 1970, pages 1-9. Originally published by TRW.
See: <http://www.cs.umd.edu/class/spring2003/cmsc838p/Process/waterfall.pdf>

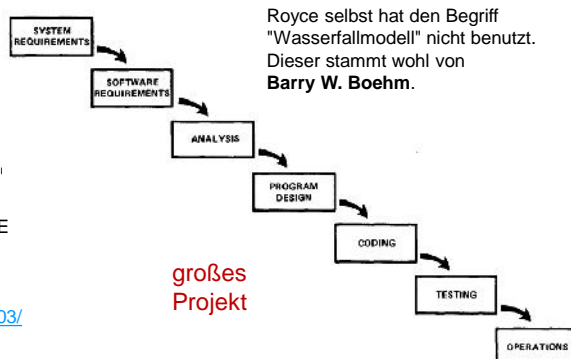


Figure 2. Implementation steps to develop a large computer program for delivery to a customer.

12

Vorlesung PRG 1
Vorgehens- und Prozessmodelle

Prof. Dr. Detlef Krömer

studiumdigitale
megadigitale

GOETHE
UNIVERSITÄT
FRANKFURT AM MAIN

Viele Varianten (1) – auch Phasenmodell genannt.

- Existiert von vier bis acht Stufen!
- Heute meist auf fünf Stufen kondensiert!

Aus Wikipedia: Paul Hoadley, Paul Smith and Shmuel Csaba Otto Traian

CC SOME RIGHTS RESERVED

13 Vorlesung PRG 1 Vorgehens- und Prozessmodelle Prof. Dr. Detlef Krömer

studiumdigitale
megadigitale



GOETHE
UNIVERSITÄT
FRANKFURT AM MAIN



Viele Varianten (2) - Phasen



- **Ergebnisse**
- Aus Vorphase
- Lastenheft
- Projektkalkulation
- Projektplan
- Pflichtenheft,
- Produktmodell,
- UI-Modell
- Benutzerhandbuch
- Architektur
- UML-Entwürfe
- Struktogramme
- Code
- Testberichte
- Einsatzberichte

nach W. Royce (1970), mit Rückkopplung B. Boehm (1981)

14 Vorlesung PRG 1 Vorgehens- und Prozessmodelle Prof. Dr. Detlef Krömer

 				
Viele Varianten (3) - Phasen				
Autoren	Nach Stahlknecht u.a.	Nach Balzert	Nach Kühlewein u.a.	Boehm 81
Phasen	Vorphase	Planung	Planung	Feasibility
	Analyse	Definition	Analyse	Plans and Requirements
				Product Design
	Entwurf	Entwurf	Entwurf	Programming
		Implementierung	Implementierung	Integration and Test Phase
	Realisierung und Test	Abnahme und Einführung		Maintenance
	Einführung	Pflege und Wartung	Abnahme und Einführung	Phaseout
15	Vorlesung PRG 1 Vorgehens- und Prozessmodelle		Prof. Dr. Detlef Krömker	



 	
Eigenschaften des Wasserfallmodells	
<ul style="list-style-type: none"> ▶ Aktivitäten sind in der vorgegebenen Reihenfolge und in der "vollen Breite vollständig" durchzuführen. ▶ Am Ende jeder Aktivität steht ein fertiggestelltes Dokument, d. h. das Wasserfallmodell ist ein „dokumentgetriebenes“ Modell. ▶ Der Entwicklungsablauf ist sequenziell, d.h., jede Aktivität muss beendet sein, bevor die nächste anfängt. ▶ Es orientiert sich am sogenannten Top-down-Verfahren. ▶ Es ist einfach und verständlich. ▶ Eine Benutzerbeteiligung ist "nur" in der Anfangsphase vorgesehen, anschließend erfolgen der Entwurf und die Implementierung ohne Beteiligung des Benutzers bzw. Auftraggebers. Weitere Änderungen stellen danach Neuaufträge dar. 	
16	Vorlesung PRG 1 Vorgehens- und Prozessmodelle Prof. Dr. Detlef Krömker

Vorteile des Wasserfallmodells

- Klare Abgrenzung der Phasen. Jede Phase endet mit einem Meilenstein ... und klar definierten Ergebnissen
- (Vergleichsweise) einfache Möglichkeiten der Planung und Kontrolle
- (Vergleichsweise) klare Abschätzung von Kosten und Umfang **bei stabilen** Anforderungen

17
Vorlesung PRG 1
Vorgehens- und Prozessmodelle
Prof. Dr. Detlef Krömker

Nachteile des Wasserfallmodells (1)

Schon Royce betonte die Notwendigkeit von **Rücksprüngen und Prototypen**.
Ohne dies würde das **Wasserfall-Modell sowieso nie funktionieren**.

- *Abgrenzungsproblem*: Klar voneinander abgegrenzte Phasen sind häufig unrealistisch – der Übergang zwischen ihnen ist fließend: Teile eines Systems können sich noch in der Planung befinden, während andere schon in der Ausführung oder im Gebrauch sind.
- *Abfolgeproblem*: Einzelne Phasen laufen "in der Theorie" nacheinander ab, in der Praxis sind jedoch Rückschritte oft unvermeidlich.
- Unflexibel gegenüber Änderungen und im Vorgehen (Phasen müssen sequenziell abgearbeitet werden)

18
Vorlesung PRG 1
Vorgehens- und Prozessmodelle
Prof. Dr. Detlef Krömker

Nachteile des Wasserfallmodells (2)

- Frühes Festschreiben der Anforderungen ist oft problematisch, da es eventuell zu teuren Änderungen führt (mehrmals wiederholtes Durchlaufen des Prozesses bei Änderungen).
- Einführung des Systems **sehr spät** nach Beginn der Entwicklung, deshalb ein später *Return on Investment*.
- Fehler werden unter Umständen (**sehr/zu**) **spät** erkannt (*Big Bang*) und müssen mit erheblichem Aufwand entfernt werden.

Folgen und Risiken,

- Das letztendlich fertiggestellte Ergebnis entspricht nicht den tatsächlichen Anforderungen.
- Oftmals ein unverhältnismäßig hoher Aufwand in der Analyse- und Konzeptionsphase. Gefahren: falsche Prioritäten (Fokus auf Anforderungen und Fokus auf Dokument anstelle Produkt)

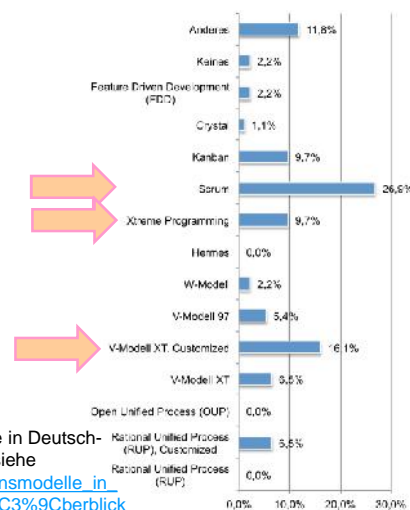
19

Vorlesung PRG 1
Vorgehens- und Prozessmodelle

Prof. Dr. Detlef Krömer

Aber trotzdem:

- "Niemand" nutzt das Wasserfallmodell mehr?
- "Jeder" kennt es.
- Einsatz nur noch gerechtfertigt bei (sehr) kleinen und wohlbekannten Problemen.





Kuhrmann, Linssen (2015): Vorgehensmodelle in Deutschland: Nutzung von 2006 - 2013 im Überblick, siehe https://www.academia.edu/24012262/Vorgehensmodelle_in_Deutschland_Nutzung_von_2006-2013_im_%C3%9Cberblick

20

Vorlesung PRG 1
Vorgehens- und Prozessmodelle

Prof. Dr. Detlef Krömer

Ab 1985: Beginnende systematische Betrachtung der Software(entwicklungs-)prozesse

Beginnende **Fokussierung auf Prozesse** nicht nur in der SW-Entwicklung



- ISO 9000-Reihe 1987
- CMM, USA (Humphrey 1989), V-Modell, Deutschland (Hummel 1990)
- Tagungsreihe International Software Process Workshop startet 1984

Vorgehensmodelle meist aus dem Umfeld großer, komplexer Systeme, **oft militärisch**

- umfangreiche, "**schwergewichtige**" Modelle

- Auch in der damals aufkommenden OO-Entwicklung, z.B. RUP



21
Vorlesung PRG 1
Vorgehens- und Prozessmodelle
Prof. Dr. Detlef Krömer

Aber auch "schon": Iteratives / inkrementelles / evolutionäres Vorgehen

- In einfacher Form (Prototyping / wenige Iterationen) bereits sehr früh (Benington 1956, Weinberg 1957, Zurcher/Randell 1968, Royce 1970)
- Ein Grund: Aufgrund der Kostenverteilung anfangs noch sehr eingeschränkt möglich, z.B.:
 - Barry Boehms erster Arbeitstag (Ende 50er):
„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 späten 80er Jahren wurde dieses Vorgehen stark ausgebaut:
 - Boehms Spiralmodell (1986),
 - Rapid Application Development (1991)

22
Vorlesung PRG 1
Vorgehens- und Prozessmodelle
Prof. Dr. Detlef Krömer






1995 - 2000

Leichtgewichtige und agile Prozesse

- Mit der wachsenden Fokussierung auf Prozesse auch Gegenbewegung
(SW-)Entwicklung als kreative Wissensarbeit (als Kunst?)
 - z.B. Takeuchi: 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.

23
Vorlesung PRG 1
Vorgehens- und Prozessmodelle
Prof. Dr. Detlef Krömker






2000

Leichtgewichtige und agile Prozesse

- 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**
 Führt den Begriff „**agil**“ ein, der „leichtgewichtig“ etc. ablöste

24
Vorlesung PRG 1
Vorgehens- und Prozessmodelle
Prof. Dr. Detlef Krömker



2000 bis 2010

Die Zeit der „Religionskriege“

(Ralf Kneuper: Klassische und agile Vorgehensmodelle – Ein historischer Überblick)

- „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“ / die "schwergewichtigen":
 - Agile Entwicklung ist Spielerei, unprofessionell
 - „Man weiß nicht, was herauskommt.“

25
Vorlesung PRG 1
Vorgehens- und Prozessmodelle
Prof. Dr. Detlef Krömer

Seit ca. 2005: Allmähliche Annäherung

- Erkenntnisse:
 - 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

26
Vorlesung PRG 1
Vorgehens- und Prozessmodelle
Prof. Dr. Detlef Krömer




PAL (peace at last ???)

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



Bertrand Meyer ist nicht irgendwer.

- Emeritierter Professor für Software Engineering an der ETH Zürich. Als Nachfolger auf dem Lehrstuhl von Niklaus Wirth.
- Entwickler der Programmiersprache Eiffel.

27
Vorlesung PRG 1
Vorgehens- und Prozessmodelle
Prof. Dr. Detlef Krömer

Und was soll nun eine arme CS Student*in damit?

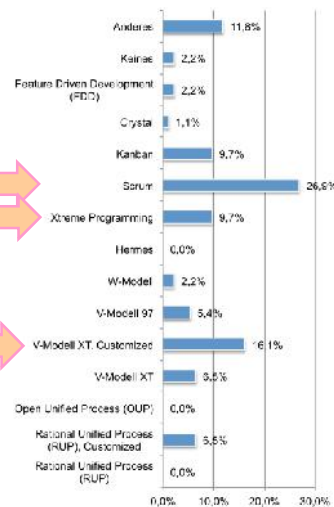
Aber vorsichtig: Datenbasis ist klein!

Dies ist keine SWE-Vorlesung.

Aber:



- V-Model (und Varianten)
- Xtreme Programming XP
- Scrum

in Grundzügen kennenlernen.



Model	Percentage
Anderes	11.8%
Kanban	9.7%
Scrum	26.6%
Xtreme Programming	9.7%
Hierarchisch	0.0%
W-Modell	2.2%
V-Modell 97	5.4%
V-Modell XT Customized	16.1%
V-Modell XT	3.3%
Open Unified Process (OUP)	0.0%
Rational Unified Process (RUP) Customized	5.5%
Rational Unified Process (RUP)	0.0%



28
Vorlesung PRG 1
Vorgehens- und Prozessmodelle
Prof. Dr. Detlef Krömer

V-Model

als Beispiel für einen "schwergewichtigen"

29
Vorlesung PRG 1
Vorgehens- und Prozessmodelle
Prof. Dr. Detlef Krömker

Scrum ...

... ist der **De-facto-Standard** in der agilen Softwareentwicklung.

... geht auf die US-amerikanischen Software-Entwickler **Jeff Sutherland und Ken Schwaber** zurück.

Der Begriff „Scrum“ stammt eigentlich aus dem **Rugby(!)** und bedeutet so viel wie „Gedränge“, wurde einer Veröffentlichung 1986 von Ikuji Nonaka entlehnt.

... ist eine Adaption der sogenannten *Lean Production*, der „schlanken Produktion“: Toyota, Daimler u.a.

Erste Präsentation auf der **OOPSLA 95** (Object-Oriented Programming, Systems, Languages & Applications) ist eine ACM Konferenz:
 Sutherland, Jeff; Schwaber, Ken (1995). *Business object design and implementation: OOPSLA '95 Workshop Proceedings*. The University of Michigan. p. 118. ISBN 3-540-76096-2.

2001 veröffentlichten eine Gruppe von 17 Entwicklern das "*Manifesto for Agile Software Development*" https://www.agilealliance.org/wp-content/uploads/2017/10/Agile-Manifesto-Color.pdf?utm_source=edit-profile-page-manifesto-link&utm_medium=manifesto-download-link

30
Vorlesung PRG 1
Vorgehens- und Prozessmodelle
Prof. Dr. Detlef Krömker

Agile Manifesto

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

While there is value in the items on the right, we value the items on the left more.

12 Principles

<div style="background-color: #4a86e8; color: white; border-radius: 50%; width: 40px; height: 40px; display: flex; align-items: center; justify-content: center; margin: 10px;">1</div> <p>Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.</p>	<div style="background-color: #4a86e8; color: white; border-radius: 50%; width: 40px; height: 40px; display: flex; align-items: center; justify-content: center; margin: 10px;">7</div> <p>Working software is the primary measure of progress.</p>
<div style="background-color: #4a86e8; color: white; border-radius: 50%; width: 40px; height: 40px; display: flex; align-items: center; justify-content: center; margin: 10px;">2</div> <p>Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.</p>	<div style="background-color: #4a86e8; color: white; border-radius: 50%; width: 40px; height: 40px; display: flex; align-items: center; justify-content: center; margin: 10px;">8</div> <p>Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.</p>
<div style="background-color: #4a86e8; color: white; border-radius: 50%; width: 40px; height: 40px; display: flex; align-items: center; justify-content: center; margin: 10px;">3</div> <p>Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.</p>	<div style="background-color: #4a86e8; color: white; border-radius: 50%; width: 40px; height: 40px; display: flex; align-items: center; justify-content: center; margin: 10px;">9</div> <p>Continuous attention to technical excellence and good design enhances agility.</p>

31
Vorlesung PRG 1
Vorgehens- und Prozessmodelle
Prof. Dr. Detlef Krömker

<div style="background-color: #4a86e8; color: white; border-radius: 50%; width: 40px; height: 40px; display: flex; align-items: center; justify-content: center; margin: 10px;">4</div> <p>Business people and developers must work together daily throughout the project.</p>	<div style="background-color: #4a86e8; color: white; border-radius: 50%; width: 40px; height: 40px; display: flex; align-items: center; justify-content: center; margin: 10px;">10</div> <p>Simplicity—the art of maximizing the amount of work not done—is essential.</p>
<div style="background-color: #4a86e8; color: white; border-radius: 50%; width: 40px; height: 40px; display: flex; align-items: center; justify-content: center; margin: 10px;">5</div> <p>Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.</p>	<div style="background-color: #4a86e8; color: white; border-radius: 50%; width: 40px; height: 40px; display: flex; align-items: center; justify-content: center; margin: 10px;">11</div> <p>The best architectures, requirements, and designs emerge from self-organizing teams.</p>
<div style="background-color: #4a86e8; color: white; border-radius: 50%; width: 40px; height: 40px; display: flex; align-items: center; justify-content: center; margin: 10px;">6</div> <p>The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.</p>	<div style="background-color: #4a86e8; color: white; border-radius: 50%; width: 40px; height: 40px; display: flex; align-items: center; justify-content: center; margin: 10px;">12</div> <p>At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.</p>

AgileAlliance.org

Manifesto Authors

Kent Beck	Alistair Cockburn	Martin Fowler	Steve Vellor	James Grenning	Andrew Hunt	Jon Kern	Ken Schwaber
Niklas Becks	Ward Cunningham	Robert C. Martin	Casey Thomas	John Highsmith	Ron Jeffries	Brian Marick	Jeff Sutherland
Alle van Bommel							

© 2001, the Agile Manifesto authors
This declaration may be freely copied in any form, but only in its entirety through this notice.

32
Vorlesung PRG 1
Vorgehens- und Prozessmodelle
Prof. Dr. Detlef Krömker



Scrum-Charakteristika -- Eine Skizze

- Scrum ist einer von mehreren existierenden sogenannten **agilen Prozessen** für Softwareentwicklung und Projektmanagement - andere sind z.B. Crystal, Extreme Programming (XP), Feature Driven Development (FDD).
- In Scrum gibt es kennzeichnend
 - **fünf Aktivitäten:** Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospektive, Product Backlog Refinement
 - **drei Artefakte:** Product Backlog, Sprint Backlog, Product Increment
 - **drei Rollen:** Product Owner, Entwicklungsteam, Scrum Master außerhalb (Stakeholder)
- Die "Regeln" sind im sogenannten *Scrum Guide* beschrieben, siehe: <http://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-German.pdf>

Rollen

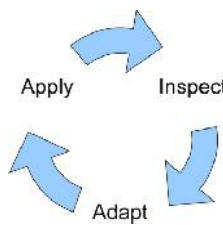
Scrum kennt drei **Rollen** für direkt am Prozess Beteiligte:

- **Product Owner** (stellt fachliche Anforderungen und priorisiert sie),
- **ScrumMaster** (managt den Prozess und beseitigt Hindernisse) und
- **Team** (3-9 Entwickler entwickeln das Produkt).
- Daneben gibt es außerhalb des Projektteams als Beobachter und Ratgeber noch die **Stakeholders**.






Der empirischer Grundansatz von Scrum

- ▶ **Transparenz:** Fortschritt und Hindernisse eines Projektes werden regelmäßig und für **alle Mitglieder** sichtbar festgehalten.
- ▶ **Überprüfung:** In regelmäßigen Abständen werden Produktfunktionalitäten geliefert und sowohl das Produkt als auch das Vorgehen beurteilt.
- ▶ **Anpassung:** Anforderungen an das Produkt, Pläne und Vorgehen werden nicht ein für alle Mal festgelegt, sondern kontinuierlich detailliert und angepasst. Scrum **reduziert die Komplexität** der Aufgabe **nicht**, strukturiert sie aber in kleinere und weniger komplexe Bestandteile, die **Inkrement**e.



35
Vorlesung PRG 1
Vorgehens- und Prozessmodelle
Prof. Dr. Detlef Krömer

Scrum bietet ...

... eine **schlanke Projektmanagement-Methode** mit folgenden Charakteristika:

- ▶ einfache Regeln
- ▶ nur drei Rollen
- ▶ mehrere Arten von Meetings mit bestimmten Zwecken
- ▶ einige Schlüssel-Artefakte, deren Pflege Overhead vermeidet und die maximale Transparenz auf einfache Weise bieten
- ▶ Pragmatismus statt Dogmatik
- ▶ iteratives Vorgehen
- ▶ Selbstorganisation und Eigenverantwortung in interdisziplinären Teams
- ▶ Konzentration auf hochqualitative Arbeit anstatt auf eine Papierflut bei der Spezifikation
- ▶ Änderungen der Kundenanforderungen während des Projekts gelten als normal, nicht als Störfaktor (es gibt keine „fertige“ Spezifikation)
- ▶ speziell geeignet für hochkomplexe Projekte mit unklaren Anforderungen (nach der persönlichen Erfahrung des Autors der Normalfall)

36
Vorlesung PRG 1
Vorgehens- und Prozessmodelle
Prof. Dr. Detlef Krömer

Prinzipieller Ablauf

Die Anforderungen (**Requirements**) werden in einer Liste (**Product Backlog**) gepflegt, erweitert und priorisiert.

Das Product Backlog ist ständig im Fluß.

Monatlich wird vom **Team** in Kooperation mit dem **Product Owner** ein definiertes Arbeitspaket dem oberen, höher priorisierten Ende des Product Backlogs entnommen und **komplett** in Funktionalität umgesetzt (**inkl. Test und notwendiger Dokumentation**).

Dieses Arbeitspaket, das **Increment**, wird während der laufenden Iteration, des sog. **Sprints**, **nicht** durch Zusatzanforderungen **modifiziert**, um seine Fertigstellung nicht zu gefährden.

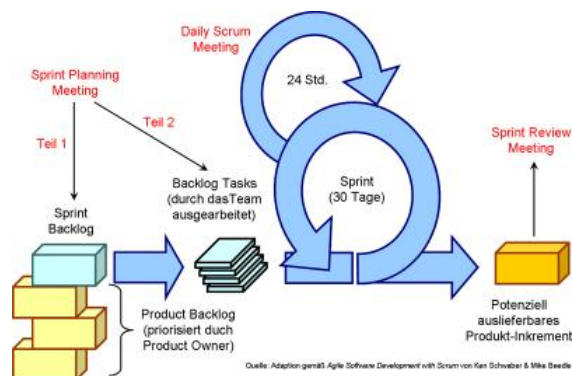
Alle anderen Teile des Product Backlogs können vom Product Owner in Vorbereitung für den nachfolgenden Sprint verändert bzw. neu priorisiert werden.

37

Vorlesung PRG 1
Vorgehens- und Prozessmodelle

Prof. Dr. Detlef Krömker

Übersicht zum Vorgehen bei Scrum



38

Vorlesung PRG 1
Vorgehens- und Prozessmodelle

Prof. Dr. Detlef Krömker

Der Sprint

Das Arbeitspaket wird in kleinere Arbeitspakete (**Tasks**) zerlegt und mit jeweils zuständigem Bearbeiter und täglich aktualisiertem Restaufwand in einer weiteren Liste, dem **Sprint Backlog**, festgehalten.

Während des Sprints arbeitet das Team konzentriert und ohne Störungen von außen daran, die Tasks aus dem Sprint Backlog in ein **Increment of Potentially Shippable Functionality** umzusetzen.

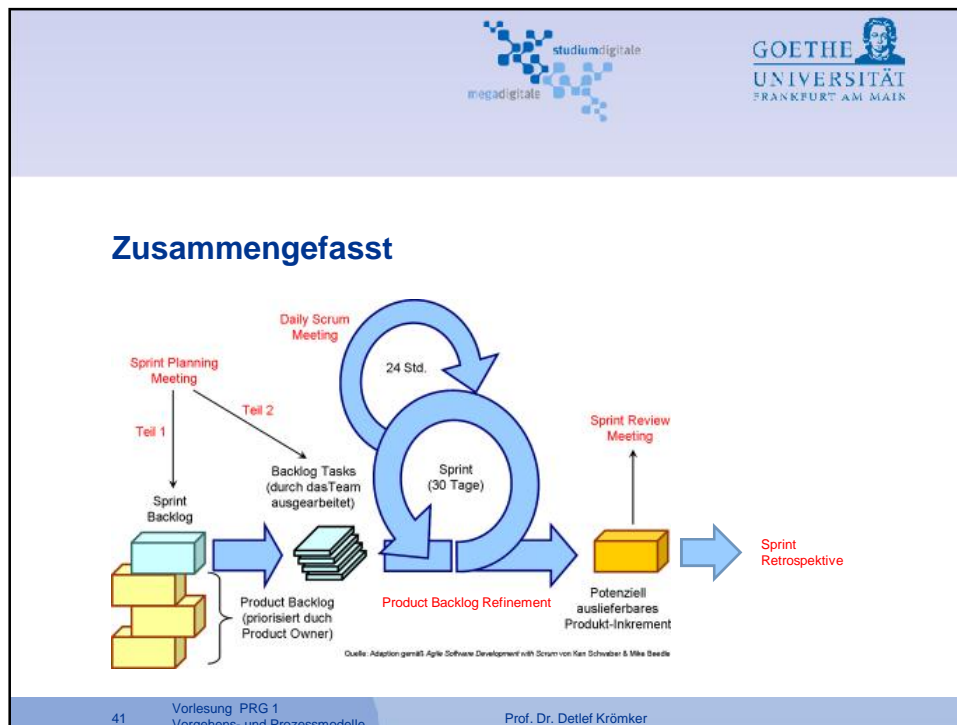
Das Team gleicht sich in einem täglichen, streng auf 15 Minuten begrenzten Informations-Meeting, dem **Daily Scrum Meeting**, ab, damit jeder weiß, woran der andere zuletzt gearbeitet hat, was er als nächstes vor hat und welche Probleme es evtl. gibt.

Der Sprint-Abschluss

Am Ende des Sprints präsentiert das Team dem Product Owner, den Stakeholders u.a. interessierten Teilnehmern in einem sog. **Sprint Review Meeting** live am System die implementierte Funktionalität.

Halbfertiges oder gar Powerpoint-Folien sind während des Reviews verboten.

Das Feedback der Zuseher und die neuen Anforderungen des Product Owners für den kommenden Sprint fließen dann wieder in das nächste **Sprint Planning Meeting** ein, und der Prozess beginnt von neuem.





Meetings und Aktivitäten

- Scrum kennt einige Arten von **Meetings**, die jeweils einem bestimmten Zweck dienen und zu definierten Zeitpunkten im Scrum-Prozess stattfinden. Alle Meetings (und auch sonstige Aktivitäten) finden in wohldefinierten **Time-Boxes** statt.

Time-Boxing

- Eine Time-Box ist ein Zeitabschnitt, **der nicht überschritten werden darf** und in dessen Grenzen Meetings oder Entwicklungs-Inkmente ablaufen. Die Time-Box ist eines der grundlegenden Konzepte in Scrum und trägt maßgeblich zur Effizienz des Prozesses bei.

42 Vorlesung PRG 1 Vorgehens- und Prozessmodelle Prof. Dr. Detlef Krömkner

Aktivität: Sprint Planning

- Im **Sprint Planning** werden zwei Fragen beantwortet:
 - Was kann im kommenden Sprint entwickelt werden?
Festlegung des Was
 - Wie wird die Arbeit im kommenden Sprint erledigt?
Festlegung des Wie
- **Time Box:** häufig in zwei Teile geteilt.
Sie dauert in Summe maximal 2 Stunden je Sprint-Woche, also zum Beispiel 2x4 Stunden.

43
Vorlesung PRG 1
Vorgehens- und Prozessmodelle
Prof. Dr. Detlef Krömker




Aktivität: Daily Scrum

- Teilnehmer: Entwicklerteam, bei dem Scrum Master und Product Owner häufig anwesend, jedoch nicht aktiv beteiligt sind. (Falls sie nicht selbst Backlogelemente bearbeiten.)
- Zweck des Daily Scrum ist der Informationsaustausch. Im Daily Scrum werden keine Probleme gelöst.
- Bewährt: Jedes Teammitglied sagt mit Hilfe des Taskboards, was es seit dem letzten Daily Scrum erreicht hat, was es bis zum nächsten Daily Scrum erreichen möchte, und was dabei im Weg steht.
- **Time Box: Daily Scrum** höchstens 15 min.

44
Vorlesung PRG 1
Vorgehens- und Prozessmodelle
Prof. Dr. Detlef Krömker

Aktivität: Sprint Review

- Das Sprint Review steht am Ende des Sprints.
- Aufgabe: Überprüfung des Inkrements: KEINE Powerpoints. Die Stakeholder benutzen die fertige Funktionalität des Inkrements und validieren diese.
- Das Entwicklungsteam präsentiert seine Ergebnisse (und es wird überprüft, ob das zu Beginn gesteckte Ziel erreicht wurde).
- Das Scrum Team und die Stakeholder besprechen die Ergebnisse und was als Nächstes zu tun ist
- Ergebnis ist das vom Product Owner notierte Feedback der Stakeholder
→ zur weiteren Gestaltung des Product Backlogs im nächsten Product Backlog Refinement.

Time Box: Das **Sprint-Review** dauert maximal 1 Stunde je Sprint-Woche, also z.B. 4 Stunden.

Aktivität: Sprint Retrospektive

- steht am Ende eines Sprints.
- Das Scrum Team überprüft seine bisherige Arbeitsweise, um sie in Zukunft effizienter und effektiver zu machen.
- Der Scrum Master unterstützt das Scrum Team darin, gute Praktiken und Verbesserungen zu finden, die im nächsten Sprint umgesetzt werden.
- Die Verbesserungsmaßnahmen werden dokumentiert und geplant.

Time Box: Die Sprint-Retrospektive dauert maximal 45 min je Sprint-Woche, also max. drei Stunden für einen 4-Wochen-Sprint.

Aktivität: Product Backlog Refinement

ist ein fortlaufender Prozess, bei dem der Product Owner und das Entwicklungsteam gemeinsam das Product Backlog weiterentwickeln. Hierzu gehören:

- Ordnen der Einträge
- Löschen von Einträgen, die nicht mehr wichtig sind
- Hinzufügen von neuen Einträgen
- Planung von Releases
- Detaillieren von Einträgen
- Zusammenfassen von Einträgen
- Schätzen von Einträgen

Time Box: nicht mehr als 10 % der Zeit des Entwicklungsteams.
(Bei 4 Wochen Sprint und 3 Entwicklern: $160 \times 3 / 10 = 48$ h insgesamt)

Rolle: Der ScrumMaster ... (1)

... **sorgt** während des gesamten Prozesses dafür, dass Regeln eingehalten werden und der Status aller Tasks im Sprint Backlog von den jeweils zuständigen Team-Mitgliedern täglich aktualisiert wird.

- ... macht den Projektfortschritt transparent durch einen geeigneten Reporting-Mechanismus: die Veröffentlichung sog. **Burndown Charts**, welche den Fortschritt für den aktuellen Sprint bzw. für das gesamte Projekt jeweils in Form einer Kurve visualisieren.
- ... ist dafür verantwortlich, dass Scrum gelingt.
- arbeitet mit dem Entwicklungsteam zusammen, gehört aber selbst meist nicht dazu. Er führt die Scrum-Regeln ein, überprüft deren Einhaltung und kümmert sich um die Behebung von Störungen und Hindernissen.

Der ScrumMaster ... (2)

- ... ist gegenüber dem Entwicklungsteam eine **dienende Führungskraft**.
- ist als Coach für den Prozess und die Beseitigung von Hindernissen verantwortlich. Unterschiedliche Teams und Situationen erfordern vom Scrum Master ein situatives Führen.
- Zu Beginn einer Scrum-Implementierung ist der Scrum Master eine Vollzeitstelle. Er bildet das Team in Scrum aus.
- Später kann der Scrum Master seine Rolle als Change-Manager wahrnehmen.

Rolle: Der Product Owner ... (1)

- ... ist für die Eigenschaften und den wirtschaftlichen Erfolg des Produkts verantwortlich.
- ... gestaltet das Produkt mit dem Ziel, seinen Nutzen zu maximieren.
- ... erstellt, priorisiert und erläutert die zu entwickelnden Produkteigenschaften, und er **urteilt** darüber, welche Eigenschaften am Ende eines Sprints fertiggestellt wurden.
- ... ist eine Person, **kein** Komitee. Ihm allein obliegt die Entscheidung über das Produkt, seine Eigenschaften und die Reihenfolge der Implementierung.

Der Product Owner ... (2)

- Zur Festlegung der Produkteigenschaften benutzt er das *Product Backlog*. Darin trägt er in Zusammenarbeit mit dem *Entwicklungsteam* und den *Stakeholdern* die Anforderungen an das Produkt ein.
- Der Product Owner ordnet, detailliert und aktualisiert das Product Backlog regelmäßig im *Product Backlog Refinement*.
- Als Produktverantwortlicher hält der Product Owner regelmäßig Rücksprache mit den Stakeholdern, um deren Bedürfnisse und Wünsche zu verstehen.

Rolle: Das Entwicklungsteam ... (1)

- liefert die Produktfunktionalitäten. Es trägt die Verantwortung für die Einhaltung der vereinbarten Qualitätsstandards.
- Das Entwicklungsteam organisiert sich selbst. Es lässt sich **von niemandem vorschreiben**, wie es Backlogeinträge umzusetzen hat.
- Eine interdisziplinäre Besetzung des Entwicklungsteams ist wichtig, z. B. mit Architekt, Entwickler, Tester, Dokumentationsexperte und Datenbankexperte.
- Ein Entwicklungsteam besteht aus mindestens drei, höchstens neun Mitgliedern.

Das Entwicklungsteam ... (2)

... hat weitere Aufgaben:

- Schätzung des Umfangs der Einträge im Product Backlog (im *Product Backlog Refinement*).
- unterteilt in der *Sprint Planung* die für einen Sprint ausgewählten Einträge aus dem Product Backlog in Arbeitsschritte, sogenannte Tasks (deren Bearbeitung in der Regel nicht länger als einen Tag dauern soll). Das Ergebnis ist das *Sprint Backlog*.

Artefakt: Das Product Backlog...

... ist eine geordnete Auflistung der Anforderungen an das Produkt. Eintragungen mit der höchsten Priorität werden als erste im Sprint umgesetzt.

... ist dynamisch und wird ständig weiterentwickelt.

- Alle Arbeit, die das Entwicklungsteam erledigt, muss ihren Ursprung im Product Backlog haben.
- Der **Product Owner** ist für die Pflege des Product Backlogs verantwortlich. Er verantwortet die Reihenfolge bzw. Priorisierung der Einträge.
- Anforderungen im Product Backlog sollten nicht technisch, sondern fachlich und anwenderorientiert sein → **User Stories**.




Artefakt: Das Sprint Backlog...

... ist der aktuelle Arbeitsplan (die *Tasks* (Aufgaben)) für diesen Sprint.

Es enthält:

- die ausgewählten Product-Backlog-Einträge,
- die dafür nötigen Tasks (z. B. Entwicklung, Test, Dokumentation)
- Den Stand der Bearbeitung der Aufgaben.

Das Sprint Backlog wird laufend nach der Erledigung einer (Teil-)Aufgabe von den Team-Mitgliedern aktualisiert.

Dies dient zur Übersicht des aktuellen Bearbeitungsstands (insbesondere im Daily Scrum).

Genutzt wird hierfür häufig ein **Taskboard** →

Das ist z.B. eine **Kanban-Tafel**. Quelle: Jeff.Iasovski CC; <https://de.wikipedia.org/wiki/Datel:Simple-kanban-board-.jpg>



55
Vorlesung PRG 1
Vorgehens- und Prozessmodelle
Prof. Dr. Detlef Krömer



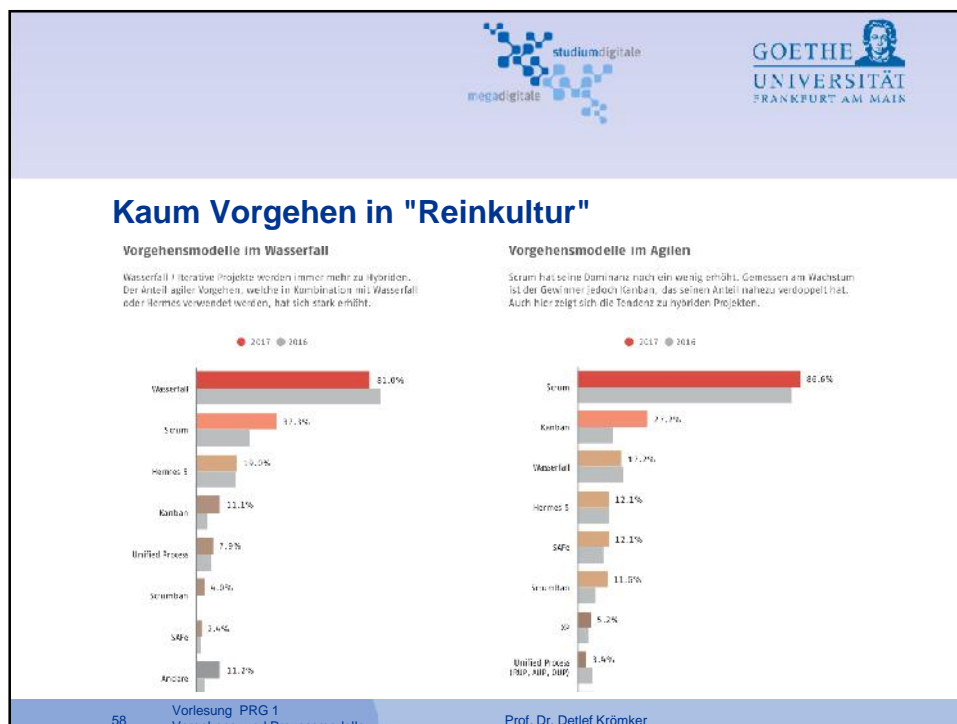
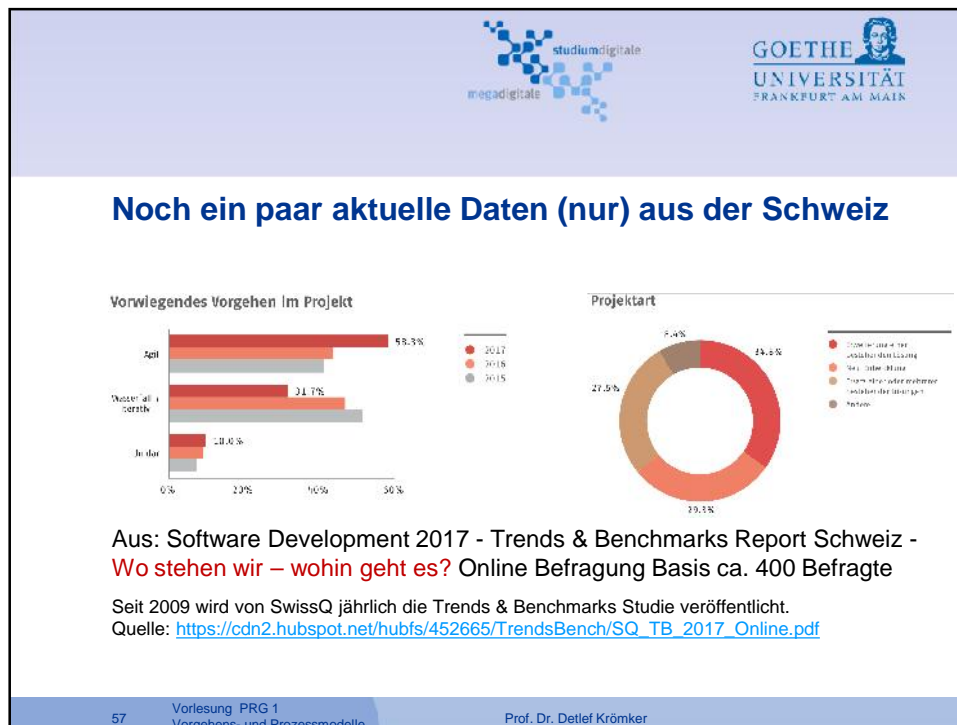

Artefakt: Product Increment

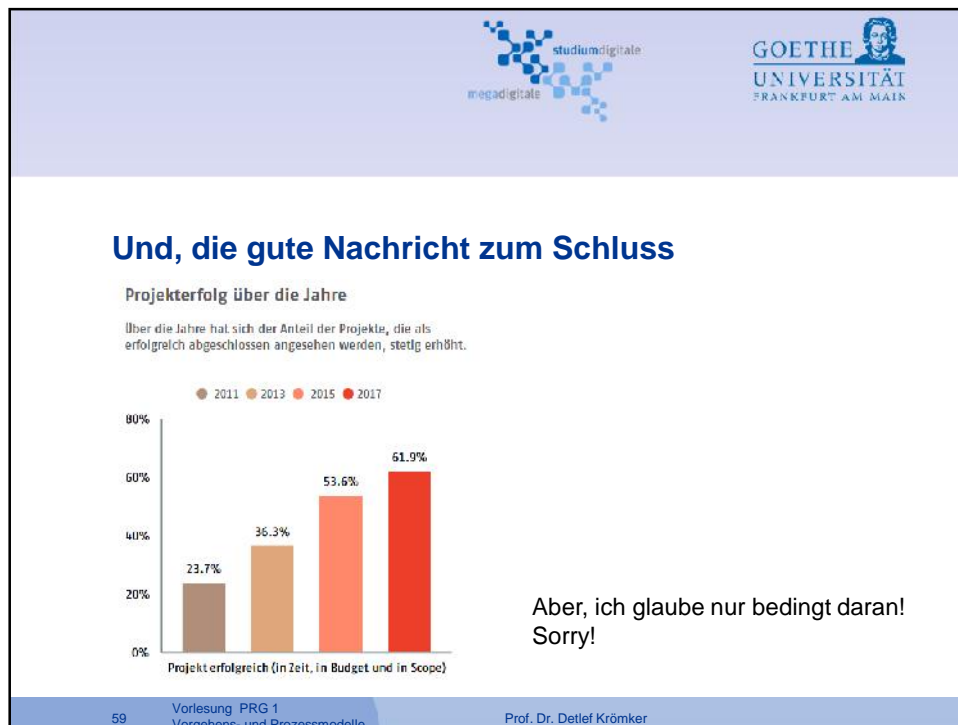
- Das Inkrement (= Meilenstein) ist das Ergebnis (ein Stück lauffähige Software) aus allen in einem Sprint fertiggestellten Product-Backlog-Einträgen **und** schon zugefügten
- Resultaten der Inkremente aller früheren Sprints.

Am Ende eines Sprints muss das neue **Inkrement in einem nutzbaren Zustand sein** und der *Definition of Done* entsprechen.

In der Art, dass der Product Owner **die Option wahrnehmen** kann, das Increment innerhalb eines Zwischen-Sprints produktiv zu setzen.

56
Vorlesung PRG 1
Vorgehens- und Prozessmodelle
Prof. Dr. Detlef Krömer





studiumdigitale
megadigitale

GOETHE
UNIVERSITÄT
FRANKFURT AM MAIN

Fragen

Fragen ?

und (hoffentlich) Antworten

60 Vorlesung PRG 1 Prof. Dr. Detlef Krömker
Grundlagen der Projekt- und Prozessmodelle



Ausblick ... kommenden Montag

**Algorithmenkonstruktion (Algorithmen(entwurfs-)muster) &
Rekursion vs. Iteration**

... und, danke für Ihre Aufmerksamkeit!

61

Vorlesung PRG 1
Vorgehens- und Prozessmodelle

Prof. Dr. Detlef Krömker