

# Agile Dokumentation verändert den Dokumentationsprozess

Marlis Friedl,  
Comet Computer GmbH

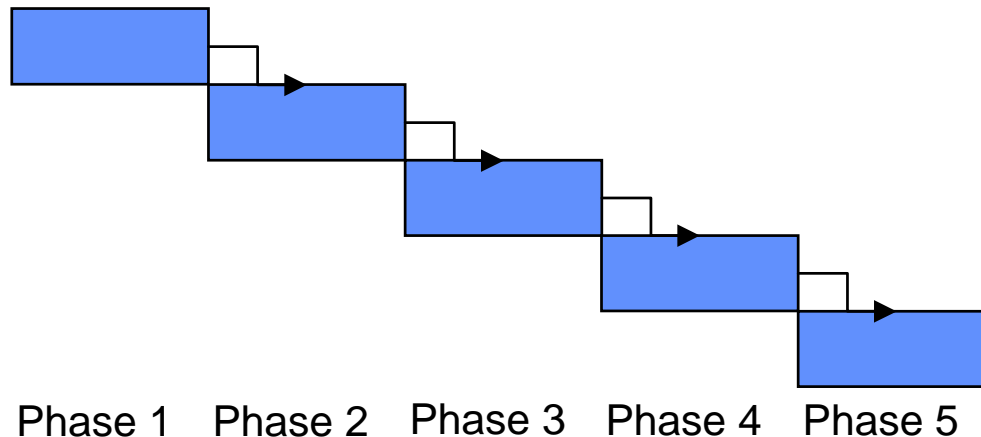
# Übersicht

- Dokumentationsprozesse in der Software-Entwicklung
- Bedingungen und Herausforderungen bei agiler Software-Entwicklung
- Lösungen der agilen Software-Entwicklung
- Diskussion der agilen Dokumentation

# Einführung in die agile Software-Entwicklung

- Agile Software-Entwicklung als Überbegriff für verschiedene Methodiken
- Februar 2001: Agiles Manifest
- Als Antwort auf die Probleme bisheriger Prozesse in der Software-Entwicklung
- Beispiele: eXtreme Programming, Scrum, Agile Modeling, Feature-driven Development
- Begriffe: Daily Scrum, Stand-Up-Meeting, Sprint, Iteration, Backlog, Pair Programming

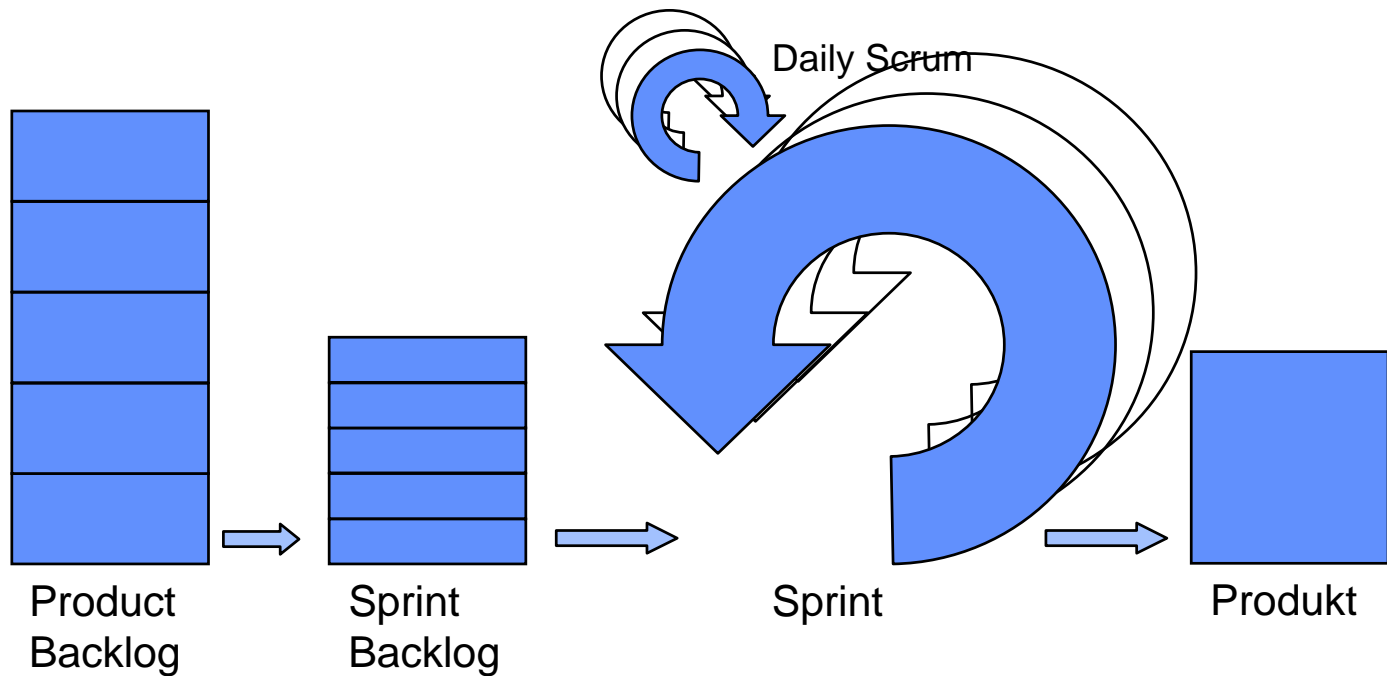
# Entwicklungsbegleitende Dokumentation – nicht agil



Lineares Phasenmodell und Wasserfall-Modell für  
Software-Entwicklung und technische Dokumentation

# Prozesse in der agiler Software-Entwicklung

## Beispiel Scrum



# Veränderte Bedingungen durch agile Software-Entwicklung

- Input ist weniger schriftlich, stärker mündlich, fragmentierter, hat kürzere Geltungsdauer
- Prozesse mit kürzeren Planungs- und Entwicklungszyklen

# Neue Herausforderungen bei agiler Software-Entwicklung

- Dokumentationsprozesse sind kürzer getaktet
- Software-Funktionen werden modularer entwickelt  
„Päckchen“ oder „Häppchen“ für die Dokumentation
- Inputrecherche und Projektmanagement brauchen mehr Zeit, gewinnen an Bedeutung

# Lösungen der agilen Dokumentation

- Dokumentationsteam, Arbeitsweisen und Dokumente bleiben relativ unverändert, aber häufigere Abgaben
- Technische Redakteure sind Teil agiler Entwicklungsteams
- Dokumentationsteam ist intern agil
  - Agile Projektorganisation, Beispiel Planbarkeit
  - Agile Dokumente, Beispiel Ausgabeformate
  - Agile Arbeitsorganisation, Beispiel Backlog

# Probleme agiler Dokumentation

- Arbeitsaufwand für die Dokumentation ist ungleich über die Zeit verteilt
- Wiederverwendbarkeit der Texte ist schwieriger
- Änderungen in den bestehenden Überblickstexten müssen bewusster geplant werden

# Vorteile agiler Dokumentation

- Bekanntes nutzen
  - Redaktionssysteme
  - Modularisierung, Topicorientierung, Single Sourcing
- Neues aufnehmen
  - Transparenz der Aufgaben und Prozesse
  - Veränderungen sind im Prozess eingeplant
  - Rückkanäle und Gestaltungsmöglichkeiten
  - Iterative Texterstellung
- Synergien mit weiteren Trends in der technischen Dokumentation

**Vielen Dank!**

**Ihre Fragen?**

# Quellen

- **Agile Software-Entwicklung**
  - Dogs, Carsten; Klimmer, Timo (2005): Agile Software-Entwicklung kompakt. Bonn: mitp.
  - Eckstein, Jutta (2004): Agile Softwareentwicklung im Großen : Ein Eintauchen in die Tiefen erfolgreicher Projekte. Heidelberg: dpunkt.verlag.
  - Hüttermann, Michael (2008): Agile Java-Entwicklung in der Praxis. Beijing u.a.:O'Reilly
- **Grundlagen Software-Entwicklung und technische Dokumentation**
  - Hackos, JoAnn T. (2007): Information Development : Managing Your Documentation Projects, Portfolio, and People . Indianapolis, Indiana: Wiley Publishing
  - Spielmann, Heinz-Jürgen (2007): Softwaretechnik. In: Schneider, Uwe; Werner, Dieter (Hrsg.): Taschenbuch der Informatik. 6., neu bearbeitete Auflage, S. 220-259.
- **Bildnachweise**
  - Wasserfall-Modell nach Spielmann 2007: 225, 228.
  - Lineares Phasenmodell nach Hackos 2007: 554.
  - Prozess bei Scrum nach Scrum Alliance 2009 ([http://www.scrumalliance.org/pages/what\\_is\\_scrum](http://www.scrumalliance.org/pages/what_is_scrum)).