

Agile Software-Entwicklung mit Scrum

Die Lösung für die Probleme der Software-Industrie?

Gastvortrag im Rahmen der SWT II-Vorlesung

Dr. Joachim Melcher

4. November 2013

Firmenprofil

Gründung: 1995

Rechtsform: Eigentümer geführte Aktiengesellschaft

Standorte: Karlsruhe (Hauptsitz), Frankfurt

Mitarbeiter: ca. 90 Mitarbeiter (Tendenz wachsend)

Kernkompetenz: agile Softwareentwicklung

Kunden u.a.: 1&1, EnBW, L-Bank, SAP, Siemens, SWR

Leistungsportfolio

Beratung /
Training



APM
ARE
ASE



Projekt-
unterstützung



APM ARE ASE

EasyMock EclEmma Eclipse EclipseRCP Energiewirtschaft
 Facets Find Bugs FIT Groovy GWT Hibernate HTML Integrationtests iText J2EE
 Java JavaScript JDBC JFace JIRA JMS JodaTime JPA JSF JUnit Karban
 Log4J Maven MS SQL Server NHibernate Oracle OSGi Scrum SOAP Solars
 Spring SQL Subversion SVN Swing SWT TDD Tomcat UML UnitTests Unity/Unix/Linux
 WCF WebServices Windows WPF XML xpath XSD

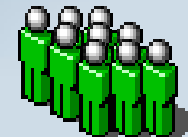


Lösungen



APM ARE ASE

EasyMock EclEmma Eclipse EclipseRCP Energiewirtschaft
 Facets Find Bugs FIT Groovy GWT Hibernate HTML Integrationtests iText J2EE
 Java JavaScript JDBC JFace JIRA JMS JodaTime JPA JSF JUnit Karban
 Log4J Maven MS SQL Server NHibernate Oracle OSGi Scrum SOAP Solars
 Spring SQL Subversion SVN Swing SWT TDD Tomcat UML UnitTests Unity/Unix/Linux
 WCF WebServices Windows WPF XML xpath XSD



Motivation für Agilität

- (Zunächst) fehlgeschlagene Softwareprojekte
- Klassische Vorgehensmodelle als Ursache?



**Welche großen
Softwareprojekte
fallen Ihnen ein?**

**Waren diese
erfolgreich?**

(Zunächst) fehlgeschlagene Softwareprojekte

Beispiele:

- INPOL-neu (Informationssystem der Polizei) [1]
- Toll Collect (Lkw-Mautsystem) [2]
- A2LL (Arbeitslosengeld II – „Hartz IV“) [3]
- DoSV (zentrale Hochschulzulassung) [4][5][6]

Klassische Vorgehensmodelle als Ursache?

Software**projekt**

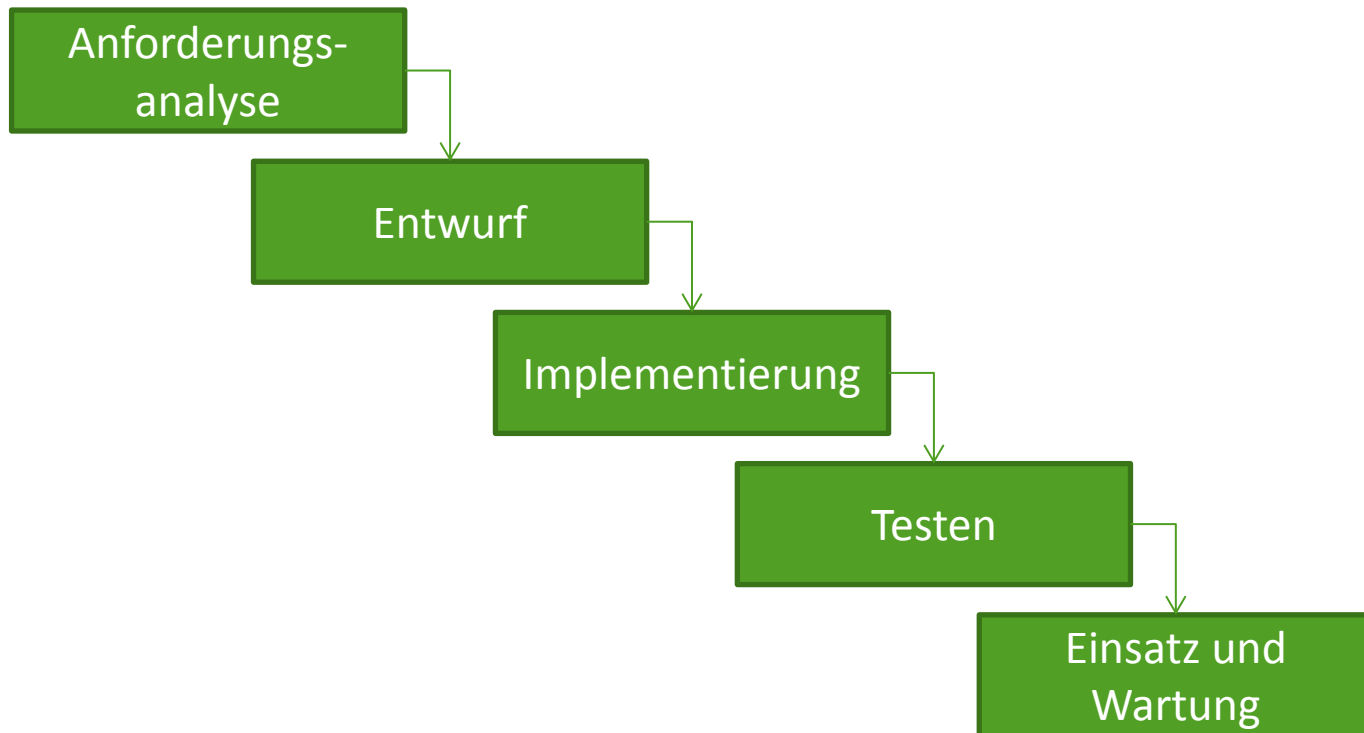
- Ziel
- einmalig
- Anfang/Ende
- Ressourcen



Vorgehensmodell

Wasserfallmodell (1)

- Vorgehensmodell mit linear angeordneten Phasen
- nicht iterativ



Wasserfallmodell (2)

Eigenschaften:

- starr:
 - kein „Zurück“, nur eine Richtung
- „schwergewichtig“
 - viele Dokumente
 - erst spät lauffähige Software

Voraussetzungen:

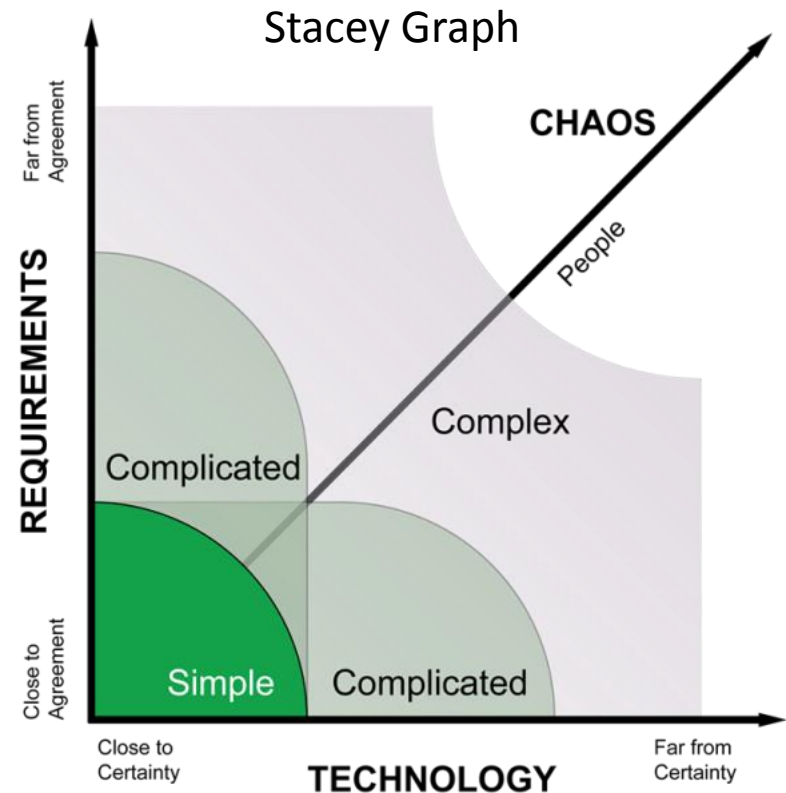
- Anforderungen stabil
- Technologie bekannt
- keine „Überraschungen“



**Wie realistisch
sind diese
Voraussetzungen?**

Realität in Softwareprojekten

- Anforderungen ändern sich
 - Gesetzesänderungen
 - Konkurrenz
 - Geschäftsstrategie
 - Zeit/Budget
- Unsicherheiten bei Technologie
 - wechselnde Technologie
 - fehlendes (internes) Know-how
 - unausgereifte Technologie



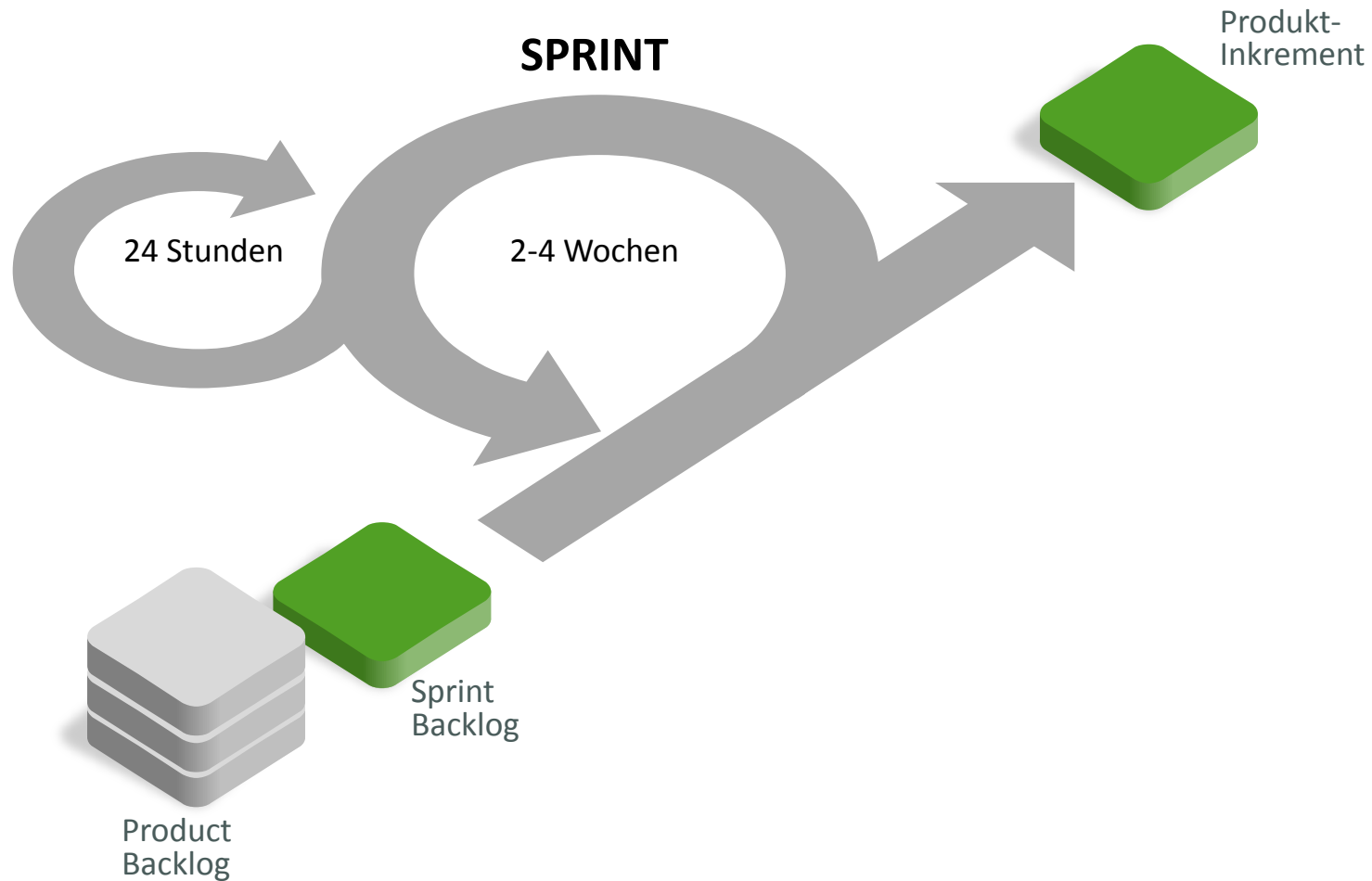
Scrum

- Grundidee
- Rollen, Artefakte, Aktivitäten

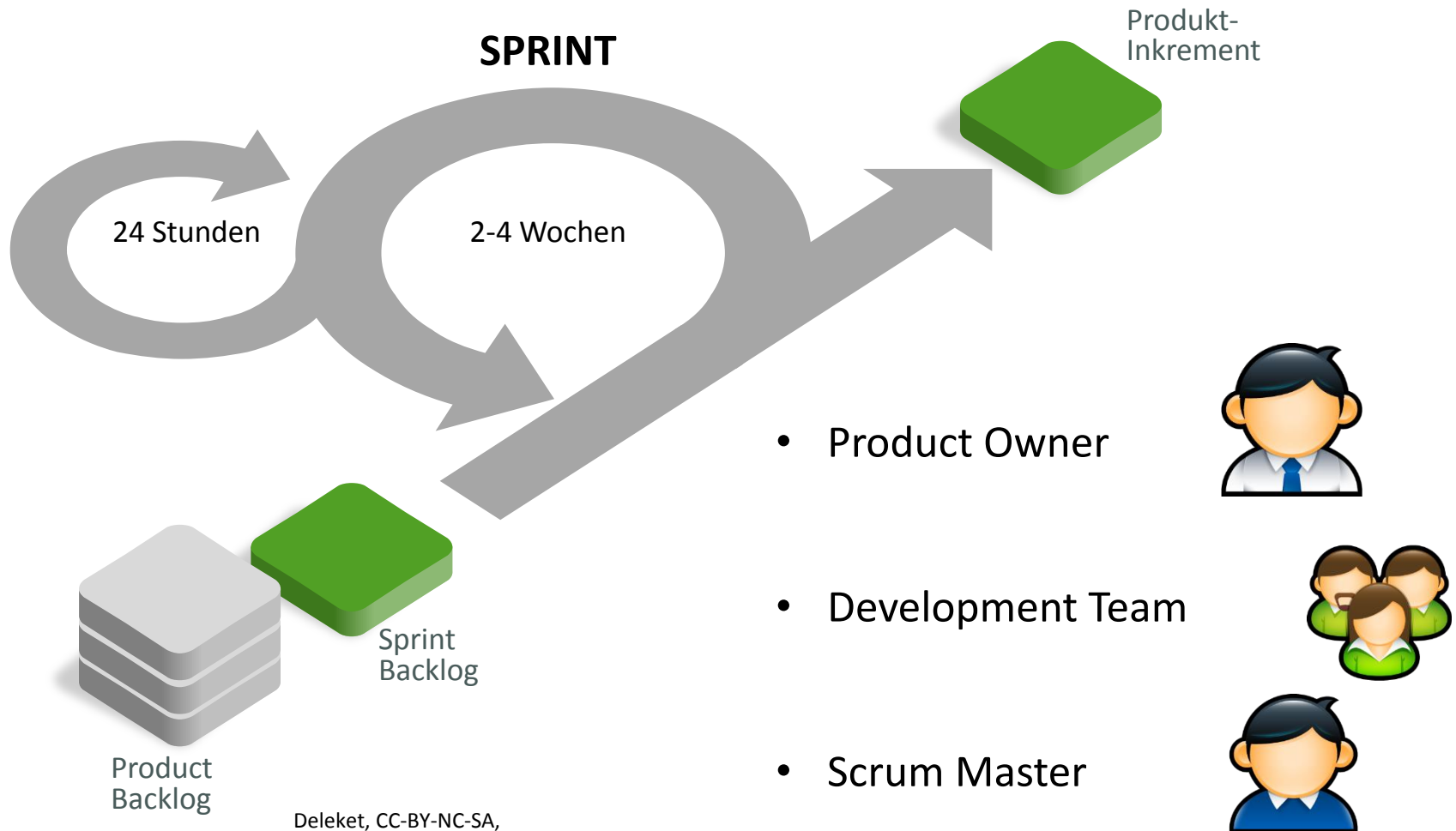
Grundidee (1)

- **iterative** Entwicklung in **Zyklen** (sog. Sprints)
- nach jedem Zyklus **auslieferbare** (Teil-)Funktionalität (sog. Produkt-Inkrement)
 - Umgehen des „Wir sind zu 90 % fertig“-Problems
- nach jedem Zyklus werden die **wichtigsten** Funktionalitäten für den **nächsten** Zyklus bestimmt
 - dadurch Reaktion auf Änderungen möglich

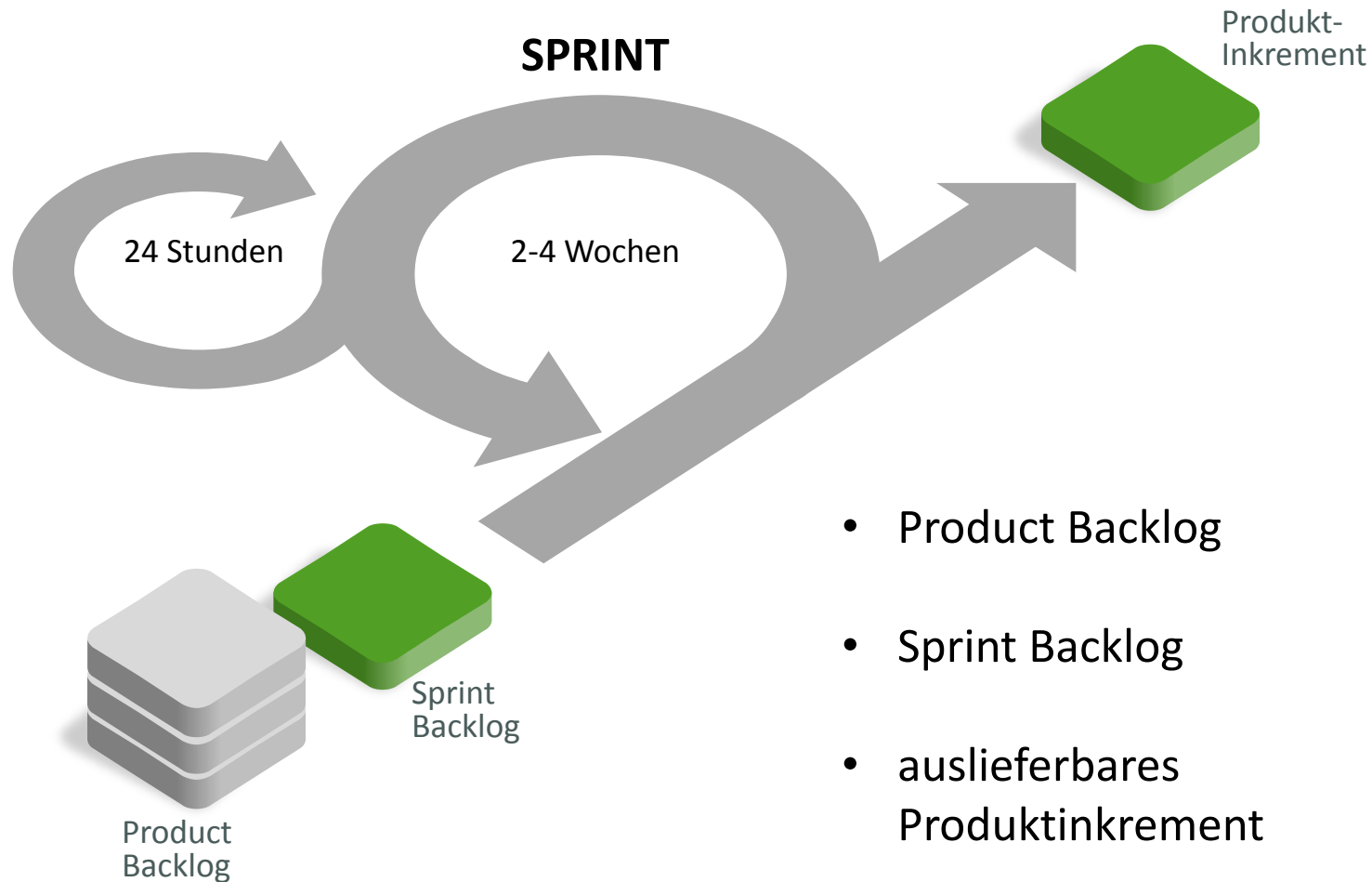
Grundidee (2)



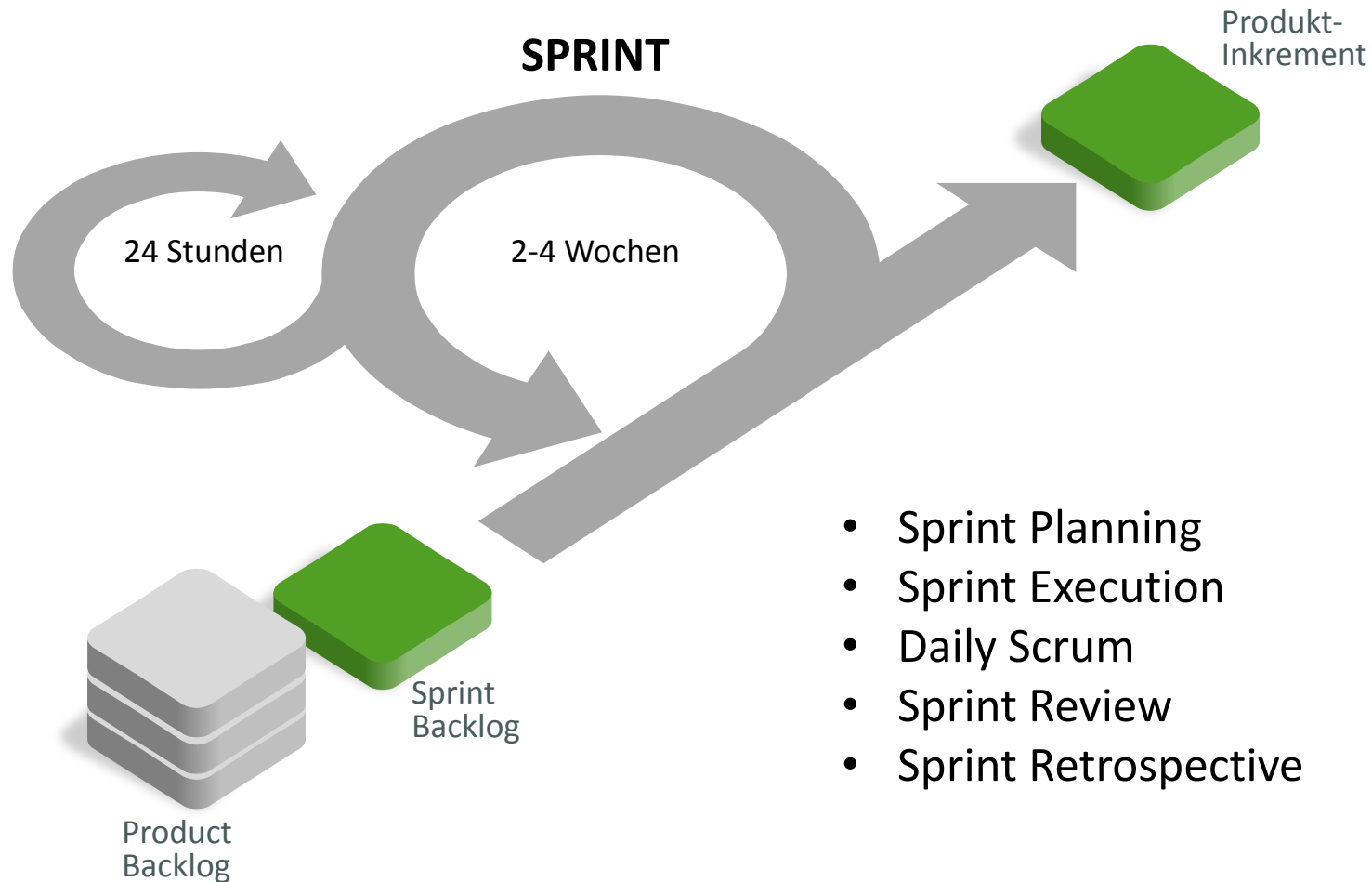
Rollen



Artefakte



Aktivitäten



Details und Übungen

- User Stories
- Priorisierung von User Stories
- Aufwandsschätzung von User Stories
- Monitoring des Arbeitsfortschritts während Sprint

User Stories (1)

- Beschreibung gewünschter Funktionalität
- Aufbau:
 - „Als ... (Rolle, Akteur)
will ich ... (Funktion, Ziel).“
 - ggf. weitere Details zur Programmlogik, Vorbedingungen, Nachbedingungen
 - Akzeptanztest(s)
- Beispiel:

„Als Bankkunde
möchte ich am Bankautomaten meinen
Kontostand abrufen können.“

User Stories (2)

- Anforderungen an User Stories (nach Bill Wake):
 - unabhängig
 - nützlich
 - schätzbar
 - klein
 - testbar



**Formulieren Sie
User Stories
für die
Software einer
Bibliothek!**



**Wie werden
User Stories
priorisiert?**

Priorisierung von User Stories

- Anwender-, Kunden- bzw. Geschäftsnutzen
- Häufigkeit der Verwendung
- evtl. Konkurrenzprodukt
- (technologisches) Risiko

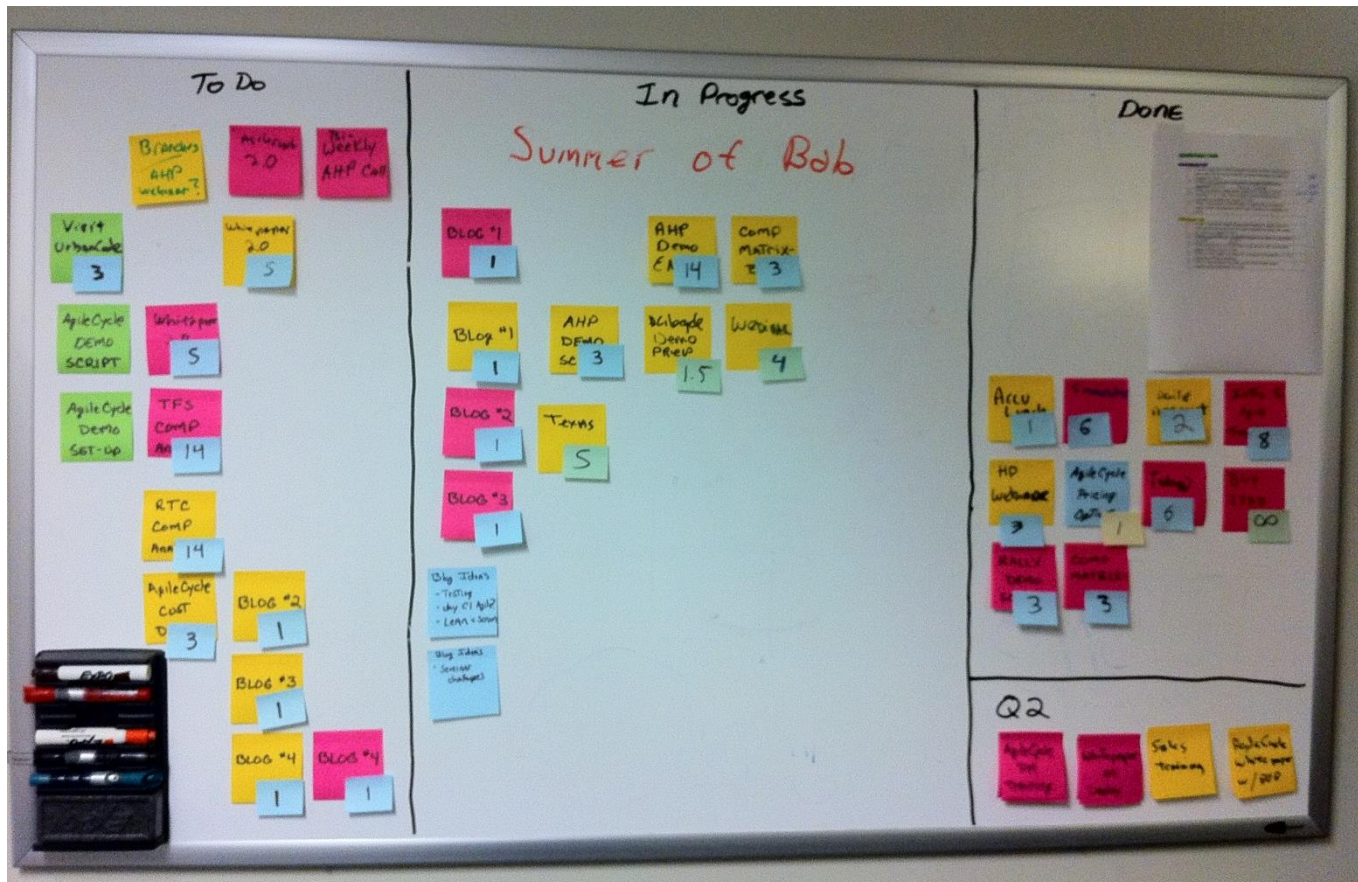
Aufwandsschätzung von User Stories

- Schätzung in „Story Points“
 - individuelles Maß des Development Teams
 - nicht identisch mit Personenstunden
 - relatives Maß für den Aufwand einer User Story
- Planning Poker
 - Aufwandsschätzung im Team
 - Jedes Teammitglied wählt verdeckt eine Karte
 - Karten werden aufgedeckt
 - bei großem Unterschied: Diskussion und „neue“ Runde



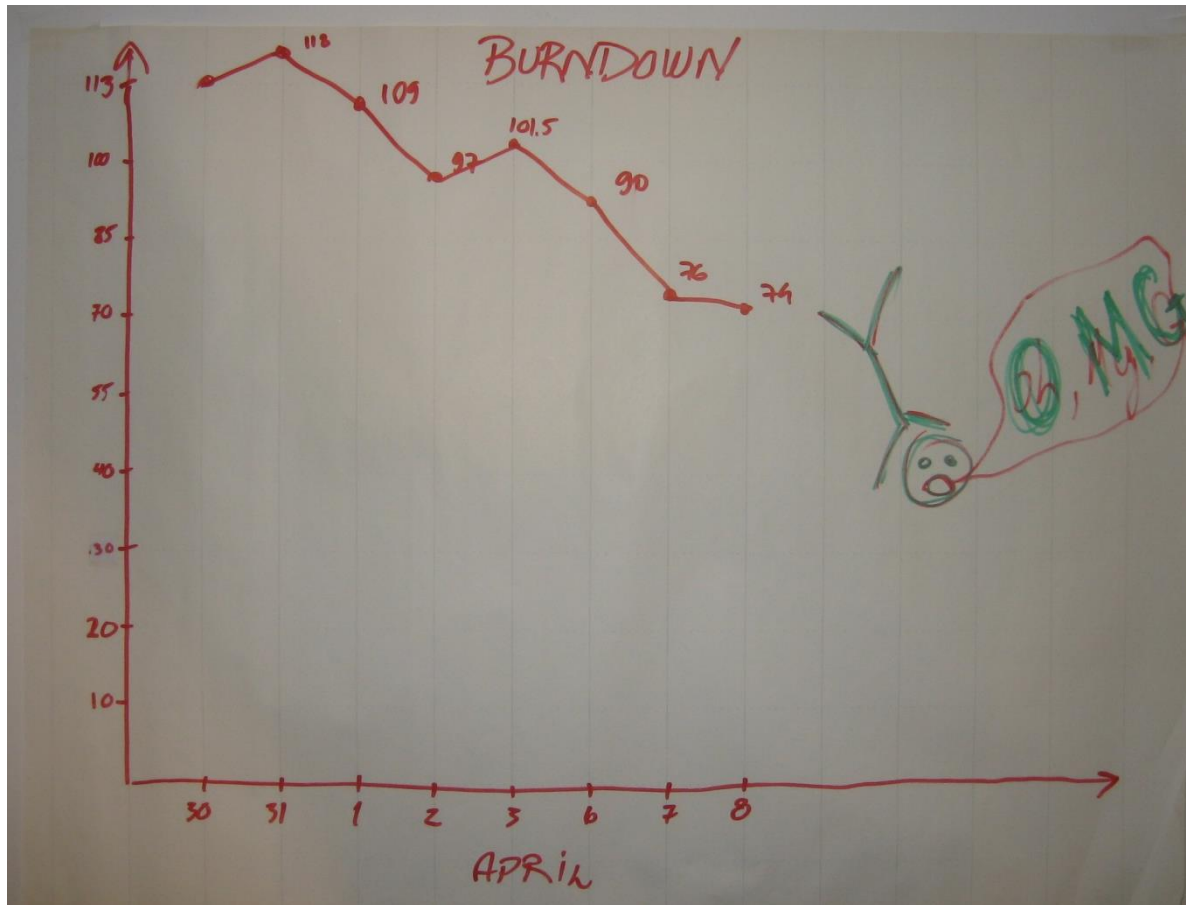
Monitoring des Arbeitsfortschritts während Sprint (1)

- Scrum Board



Monitoring des Arbeitsfortschritts während Sprint (2)

- Burndown-Chart



Schwierigkeiten bei Einführung bzw. Einsatz von Scrum

- Change Management
- Aufgabenänderungen der Beteiligten
- Missbrauchsmöglichkeit der Transparenz
- Prozess-Framework Scrum vs. Vorgehen beim Implementieren
- Vertragsgestaltung

Change Management

- Menschen „hassen“ Veränderungen
 - Angst vor Kontrollverlust
 - Angst vor Know-how-Verlust
 - evtl. Angst vor Arbeitsplatzverlust
- Konsequenz: Widerstände
- nach Einführung von Änderung zunächst Abfall der Produktivität
- erst nach gewisser Zeit (Investment!) wird (im besten Fall!) Änderung angenommen und ursprüngliche Produktivität übertroffen

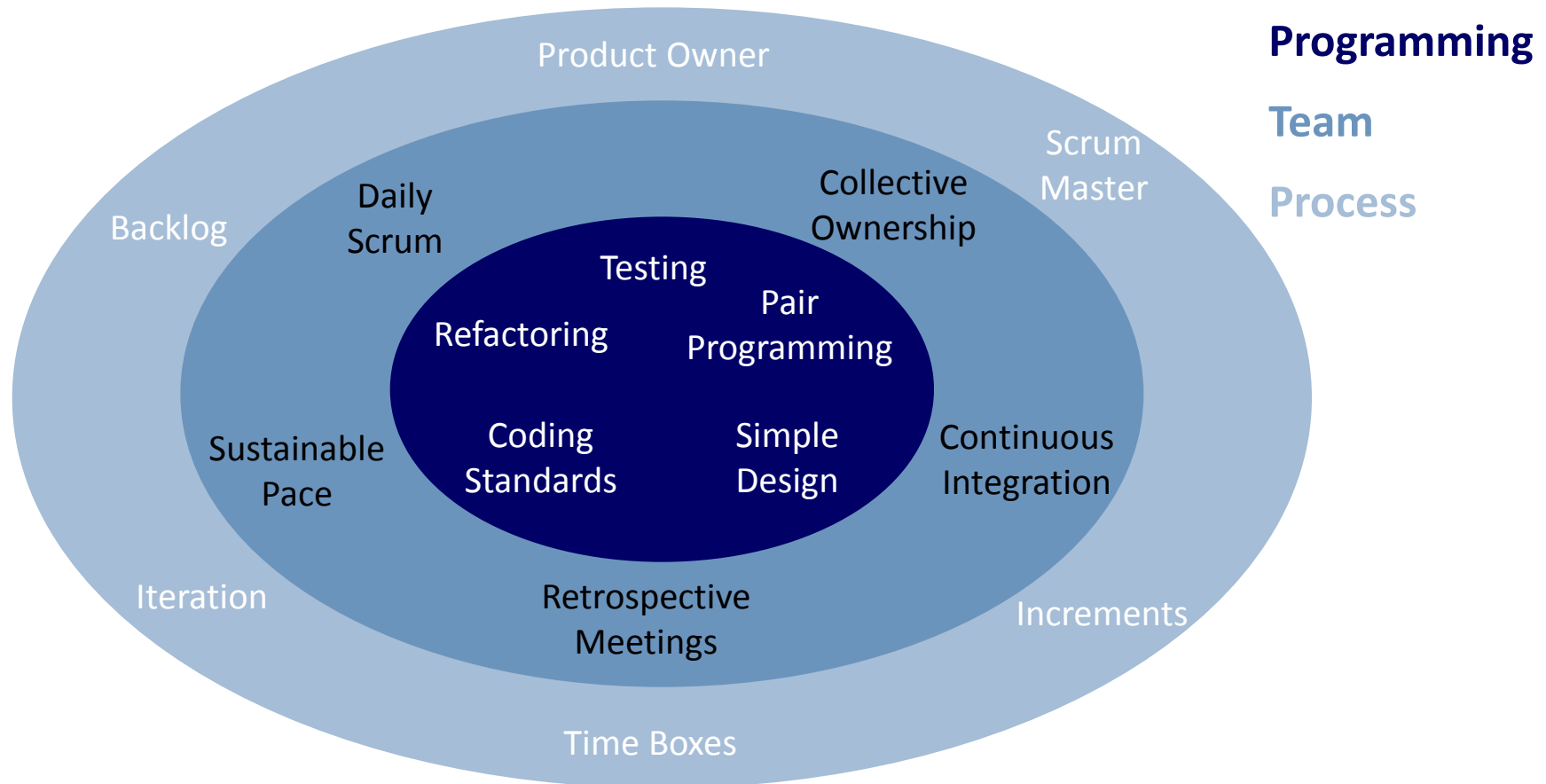
Aufgabenänderungen der Beteiligten

- Management
 - kein „Micro-Management“ mehr
 - viel Verantwortung nun beim Development Team
 - (verbliebene/neue) Aufgaben:
 - IT-Strategie (Was wird wann mit welchen Ressourcen umgesetzt?)
 - Teamzusammenstellung
 - Arbeitssituation des Development Teams „optimieren“
- Product Owner
 - ein Verantwortlicher für Aufgabenpriorisierung
 - Vertreter für ggf. viele Stakeholder mit unterschiedlichen Interessen
- Development Team
 - keine festen Rollen mehr (Architekt, Frontend, Backend, Tester)
 - stattdessen: „cross-functional“ und selbst organisierend

Missbrauchsmöglichkeit der Transparenz

- Transparenz (z.B. durch Daily Scrum, Burndown Chart usw.) kann zur individuellen Leistungskontrolle der Mitarbeiter missbraucht werden

Prozess-Framework Scrum vs. Vorgehen beim Implementieren



Vertragsgestaltung

- Kunden „lieben“ Festpreisprojekte
- Festpreis setzt vorher festgelegten Leistungsumfang voraus
- Widerspruch zum agilen Vorgehen
- schwierige Vertragsgestaltung

Weitere Informationsmöglichkeiten

- **JobTalk der Informatik-Fakultät**

- Firmeninfos, Arbeitsalltag Informatiker, Einstiegsmöglichkeiten
- 09.12.2013, 17.00 Uhr
- Anmeldung erforderlich!
- <http://www.informatik.kit.edu/909.php>

JobTalk

- **ASE Student Edition**

- 5-tägiges Training/Praktikum agile Software-Entwicklung
- nächster Termin: 17.-21.02.2014
- Bewerbung erforderlich!
- <http://www.andrena.de/studentpage/ase-student-edition>



Literaturverweise

Literaturverweise

- [1] <http://de.wikipedia.org/wiki/INPOL-neu>
- [2] http://de.wikipedia.org/wiki/Toll_Collect
- [3] <http://de.wikipedia.org/wiki/A2LL>
- [4] <http://www.heise.de/newsticker/meldung/Software-fuer-zentrale-Hochschulzulassung-wird-teurer-1280097.html>
- [5] <http://www.heise.de/newsticker/meldung/Zentrale-Hochschulzulassung-erneut-gescheitert-1397010.html>
- [6] <http://www.heise.de/newsticker/meldung/Hoffnungsschimmer-fuer-zentrale-Hochschulzulassung-1417163.html>