



Leichtgewichtige Traceability im agilen Entwicklungsprozess am Beispiel von Scrum

29. Nov. 2012 FG-Treffen

Elke Bouillon¹, Baris Güldali², Andrea Herrmann³,
Thorsten Keuler⁴, Daniel Moldt⁵, Matthias Riebisch⁶

¹Technische Universität Ilmenau, elke.bouillon@tu-ilmenau.de

²s-lab – Software Quality Lab/Universität Paderborn, bguldali@s-lab.upb.de

³Freie Software Engineering Trainerin und Forscherin, herrmann@herrmann-ehrich.de

⁴Fraunhofer IESE, Thorsten.keuler@iese.fraunhofer.de

⁵Universität Hamburg, moldt@informatik.uni-hamburg.de

⁶Universität Hamburg, riebisch@informatik.uni-hamburg.de

Zu den Autoren - Forschungsvorschau

Gesellschaft
für Informatik



- Arbeitskreis Traceability / Evolution
- gegründet 2006 der GI Fachgruppe Architekturen
- ➔ ein Teil der Autoren kommt also aus dem Themenkreis der Softwarearchitekturen
- ➔ Letztes Arbeitstreffen im Juli in Paderborn im Rahmen der Jahrestagung der Fachgruppe
- ➔ Frage aufgekommen, welche Konzepte für Traceability in agilen Umfeld existieren
- ➔ Offener Kreis – Mitarbeit möglich

Motivation

Traceability ist nützlich für...

- Überprüfung der Abdeckung (beispielsweise der User Stories durch Code und Testfälle)
- Konservieren von Entscheidungen
- Unterstützen von Entscheidungsfindungen
- Weitergabe von Wissen, z.B. bei der Einarbeitung neuer Teammitglieder

Motivation

Agile Entwicklung

- Leichtgewichtig
- Trend: größere Projekte + verteilte Teams
- Traceability??

Traditionelle Entwicklung

- Dokumentenlastig
- einschließlich Traceability

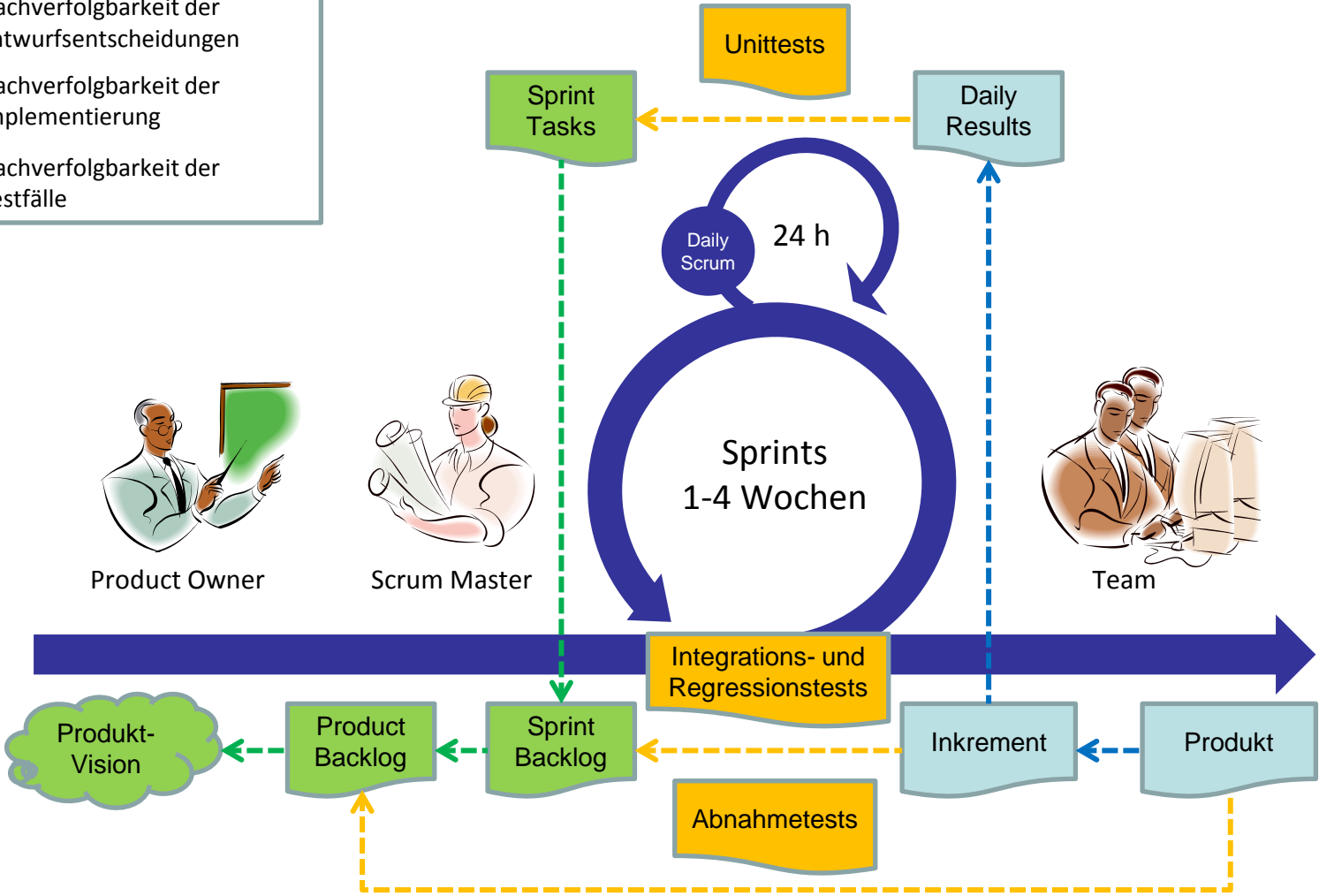
Unser Ziel

- Ansatz für Traceability in der agilen Entwicklung
- leichtgewichtig
- keine / möglichst wenige neue Artefakte, sondern die vorhandenen Artefakte nutzen und verbinden
- unabhängig von einem konkreten agilen Vorgehen

Vorgeschlagener Ansatz

Legende

- ← - - - Nachverfolgbarkeit der Entwurfsentscheidungen
- ← - - - Nachverfolgbarkeit der Implementierung
- ← - - - Nachverfolgbarkeit der Testfälle



[Ken Schwaber, Jeff Sutherland: ScrumGuide, www.scrum.org]

Traceability-Informationen in Modellen

Agile Prozesse fordern: möglichst geringer Overhead

Benötigte Traceability-Informationen z.B.:

- Quelle und Ziel – Verbindung mit einfachem (Hyper-)Link
- Letzte Änderung – um Bezug zu Sprint herzustellen
- Sicherheit – gemeint ist confidence – benötigt bei Reengineering

Unterstützung durch CASE-Werkzeug

Benötigte Funktionalität:

- (automatisches) Erkennen von Traceability-Beziehungen zwischen existierenden Artefakten
- Erstellen + Verwalten von Traceability-Informationen
- Behandeln von Änderungen
 - Erkennen von obsoleten Traceability-Informationen
 - Nachführen von Änderungen
- Auswerten von Traceability-Information
 - Verständnis und Überblick
 - Verfolgen von Entscheidungen
 - Impact Analyse

Nächste Schritte

1. Analyse agiler Werkzeuge: Unterstützung der vorgeschlagenen Artefakt-Verknüpfungen sowie Aktivitäten?
 2. Vorschlag eines praktischen Vorgehens und dessen Werkzeugunterstützung
 3. Diskussion des Ansatzes mit agilen Praktikern
 4. Praktische Evaluierung des Ansatzes in einem oder mehreren Projekten: Aufwand und Nutzen?
- Beteiligung erwünscht!
 - Nähere Informationen zum Arbeitskreis:
<http://www.theoinf.tu-ilmenau.de/~riebisch/traceability>

Vielen Dank für Ihre Aufmerksamkeit!

Fragen?