

## B.I.10

### Algorithmen – Unterrichtseinheit

# Lineare und agile Softwareentwicklung mit Wasserfallmodell und Scrum

Ein Beitrag von Johann-Georg Vogelhuber



© Eva Katarini/E+

Scrum ist ein Vorgehensmodell aus dem agilen Projektmanagement bzw. der agilen Softwareentwicklung. Es wurde in der Softwaretechnik zur Produktivitätssteigerung entwickelt, wird aber mittlerweile auch in vielen anderen Bereichen eingesetzt. Mit dieser Unterrichtseinheit erarbeiten sich die Schülerinnen und Schüler handlungsorientiert und spielerisch die Grundlagen des Scrum-Frameworks und die Unterschiede zwischen linearen (nicht iterativen) Vorgehensmodellen aus dem traditionellen Projektmanagement, wie dem Wasserfallmodell.

#### KOMPETENZPROFIL – UNTERRICHTSEINHEIT

**Klassenstufe:** 10 + Sek. II

**Dauer:** 8–16 Unterrichtsstunden

**Lernziele:** Die Lernenden kommunizieren und kooperieren, indem sie ein Softwareprojekt mit dem Scrum-Framework planen (und umsetzen).

**Kompetenzen:** Kooperieren und kommunizieren, Implementieren, Analysieren und Reflektieren

**Themenbereiche:** Scrum, Wasserfallmodell, agiles Projektmanagement, lineare Vorgehensmodelle, agile Vorgehensmodelle, agile Softwareentwicklung

Java

## Auf einen Blick

### Benötigt

- Tablet/Laptop/PC pro Schüler/in oder pro Schülerpaar
- Internetzugang



### Einstieg

**Thema:** Vorgehensmodelle in der Softwareentwicklung

**M 1** Projektplanung zu einem neuen Kundenauftrag

### Erarbeitung

**Thema:** Wasserfallmodell und Scrum als lineares und agiles Vorgehensmodell

**M 2** Softwareentwicklung mit einem linearen Vorgehensmodell – Das Wasserfallmodell

**M 3** Schritte innerhalb der Phasen des Wasserfallmodells

**M 4** Durchführung der Wasserfallmodellphasen Anforderungen und Entwurf

**M 5** Erprobung des agilen Vorgehensmodells Scrum als Alternative zum Wasserfallmodell

**Thema:** Agile Softwareentwicklung mit Scrum

**M 6** Agile Softwareentwicklung mit dem Vorgehensmodell Scrum

**Benötigt:** Ggf. *LearningSnack*: <https://raabe.click/LearningSnack-Scrum>



**M 7** Scrum-Training mit dem *Paper Plane Game*

**M 8** User Stories – Formulierung von Anforderungen in Scrum

**M 8a** Vorlage für Story Cards

### Sicherung

**Thema:** Lernfortschrittskontrolle zur Unterrichtseinheit

**M 9** Wissenstest zu linearen und agilen Vorgehensmodellen – Wasserfallmodell und Scrum

**Benötigt:** Ggf. *LearningApp* <https://raabe.click/LA-Scrum>



Ggf. **M 10 Kahoot!-Quiz**: <https://raabe.click/Kahoot-Scrum>

### Benötigte Dateien



Lösung zu M 8 Aufgabe 3: *Java\_Aufgabenmanager.zip*

# M 1

## Projektplanung zu einem neuen Kundenauftrag

### Situationsbeschreibung

Jakob hat vor Kurzem eine Ausbildung zum Fachinformatiker Anwendungsentwicklung begonnen und arbeitet derzeit in der Abteilung ‚Mobile Development‘ der MeViTo GmbH. In dieser Abteilung werden Apps für Smartphones oder Tablets nach Kundenwünschen entwickelt. Die Abteilungsleiterin Frau Schmidt erläutert Jakob den neuesten Kundenauftrag:



© Nitat Termproment

Frau Schmidt: „Der Kunde ist mit den bekannten ToDo-Apps nicht zufrieden und hätte gerne eine Lösung, die ganz auf den eigenen Betrieb ausgerichtet ist.“

Jakob: „Und welche Anforderungen sind das genau?“

Frau Schmidt: „Es soll möglich sein, in der App neue Aufgaben zu erstellen und erledigt zu markieren. Diese Aufgaben müssen eine Beschreibung, eine Deadline und den geschätzten Aufwand in Arbeitstagen haben.“

Jakob: „Das kenne ich so ungefähr auch von anderen ToDo-Apps. Der geschätzte Aufwand ist eher ungewöhnlich. Warum hat der Kunde das haben?“

Frau Schmidt: „Die Aufgaben sollen sortiert und angezeigt werden, dass oben in der Liste immer die Aufgaben stehen, die als Erstes begonnen werden müssen. Der Benutzer darf auch immer nur die oberste Aufgabe als erledigt markieren.“

Jakob: „Ich verstehe. Wir müssten also die Deadline die geplante Anzahl Arbeitstage abziehen und dann danach sortieren.“

Frau Schmidt: „Richtig. Dazu können wir als erstes die notwendigen Klassen in Java entwickeln und eine einfache Konsolenanwendung als Prototypen erstellen.“

Jakob: „Alles klar. Dann kann ich direkt anfangen zu programmieren.“

Frau Schmidt: „Besser nicht. Schau dir bitte erst unseren Entwicklungsprozess an und mach dir einen Plan, welche Schritte du durchführen musst.“

### Analyse

Fasse die Situation kurz mit eigenen Worten zusammen.

---

---

Welche Aufgabe muss Jakob bewältigen?

---

---

Wie könnte er zur Erledigung dieser Aufgabe vorgehen?

---

---

# M 3 Schritte innerhalb der Phasen des Wasserfallmodells

## Situationsbeschreibung

Nachdem sich Jakob die Dokumentation genauer durchgelesen hat, bittet ihn Frau Schmidt darum, einen Zeitplan für die Umsetzung der Aufgabe zu erstellen. Jakob überlegt sich, dazu eine Tabelle aufzustellen, in der er für jede Phase notiert, welche Dokumente/Produkte er erstellen muss, und wie lange er wohl dafür brauchen wird. Dazu möchte er für jede Phase eine kurze Checkliste erstellen, damit er keinen wichtigen Arbeitsschritt vergisst.



© Maryna Pletschk/Moment

## Aufgabe 1

Vervollständigt die Tabelle so, dass Jakob die notwendigen Arbeitsschritte für die Umsetzung der Kundenanforderung (M 1) nach dem Wasserfallmodell durchführen kann.



Phase	Welches Dokument/Produkt muss erstellt werden?	An was muss bei der Erstellung gedacht werden? (Checkliste)
Anforderungen		
Entwurf		
Implementation		
Überprüfung		
Wartung		

## Aufgabe 2

Notiert für jede Phase den ungefähren Zeitbedarf in Stunden und notiert eure Schätzung in der folgenden Tabelle. Gebt dazu für jede Phase auch eine kurze Begründung an.



Phase	Schätzung	Begründung
Anforderungen		
Entwurf		
Implementation		
Überprüfung		

## Scrum-Training mit dem *Paper Plane Game*

M 7

Das Spiel wird in Teams zu je 5–6 Mitgliedern gespielt. Die Teams müssen dabei möglichst gleich groß sein.

### Ziel des Spiels

Jedes Team stellt so viele qualitativ hochwertige und getestete Papierflieger wie möglich her. Die Papierflieger müssen dabei bis zum anderen Ende des Klassenraums fliegen können.



© mrs/Moment

### Spielregeln Durchführung

Das Spiel wird in drei Sprints durchgeführt. Ein Sprint dauert 9 Minuten:

- 3 Minuten für die Planung
- 3 Minuten zum Produzieren und Testen der Flieger
- 3 Minuten für Review und Retrospektive

In der Planungsphase legt jedes Team fest, wie viele Flieger es produzieren will. Während der Produktion darf jeder Spieler immer nur einen Falz machen und muss dann den Flieger an den nächsten Spieler weitergeben. An der Produktion eines Fliegers müssen alle Spieler beteiligt sein.

### Abrechnung

Es zählen nur Papierflieger, die bis zum anderen Ende des Klassenraums fliegen können. Jedes Team muss vor Beginn des Sprints eine Schätzung für die Anzahl der zu produzierenden Papierflieger abgeben. Von der Anzahl der erfolgreich geflogenen Papierflieger muss die Anzahl der nicht getesteten oder unfertigen Flieger abgezogen werden.

### Weitere Regeln

In der Retrospektive muss jedes Team eine Verbesserungsidee entwickeln und aufschreiben. Ein Spieler in der Gruppe ist verantwortlich für das Zählen und Notieren der erfolgreichen Papierflieger. Ganz wichtig: Papier zu einem Ball wüllen und Wurfen ist verboten.

### Aufgaben zur Nachbesprechung

Diskutiert in der Gruppe folgende Punkte:

- Was würdet ihr in festem Zeitrahmen am ehesten passieren lassen?
- Welche Unterschiede ergeben sich beim Einsatz des Wasserfallmodells?
- Was hat zu einer Verbesserung während der drei Sprints geführt?

Halte eure Ergebnisse stichpunktartig fest.

## M 8

## User Stories – Formulierung von Anforderungen in Scrum

**Situationsbeschreibung**

Die MeViTo GmbH möchte den Prototypen für die Aufgabenverwaltung (M 1) mithilfe des Scrum-Prozesses umsetzen. Dazu muss der Product Owner Frau Schmidt zunächst die Anforderungen in Form von User Stories formulieren. Dazu hat sie eine Vorlage erstellt, mit der sie die User Stories in einem einheitlichen Format ausdrucken und auf ihrem Scrum Board verwalten kann.



© ljubaphoto/E+

**User Stories**

Zur Formulierung der Anforderungen verwendet man in Scrum sogenannte User Stories. Eine User Story ist eine in **Alltagssprache** formulierte Software-Anforderung, die in der Regel nicht mehr als zwei Sätze umfasst. User Stories können entweder formlos oder unter Verwendung einer Vorlage angelegt werden. Eine typische Vorlage lautet:

„Als <Rolle> möchte ich <Ziel/Wunsch>, um <Nutzen>“.

Manchmal verwendet man auch sogenannte Story Cards, um weitere Informationen erfasst werden können.

**Aufgabe 1**

Begib dich in die Rolle des Product Owners und formuliere für die Anforderungen aus der Situationsbeschreibung **M 1** einen vollständigen Satz an User Stories. Verwende dazu die Vorlage **M 8a** für die Story Cards.

**Tipp:** Über diesen Link <https://raabe.click/Learn/Scrum-UserStories> kannst du einige Beispiele von User Stories als Anregung anschauen.

**Aufgabe 2**

Bringe die einzelnen User Stories in eine priorisierte Reihenfolge. Notiere dir jeweils eine kurze stichpunktartige Begründung.

**Aufgabe 3**

Die einzelnen User Stories sollen entsprechend des Scrum-Prozesses umgesetzt werden:

• Der Sprint hat die Dauer einer Unterrichtswoche.

• Zu Beginn der Woche findet ein Planning-Meeting statt. Dieses Meeting darf nicht länger als 10 Minuten dauern. In diesem Meeting werden die zu bearbeitenden Stories entsprechend der Priorität ausgewählt und auf das Team aufgeteilt.

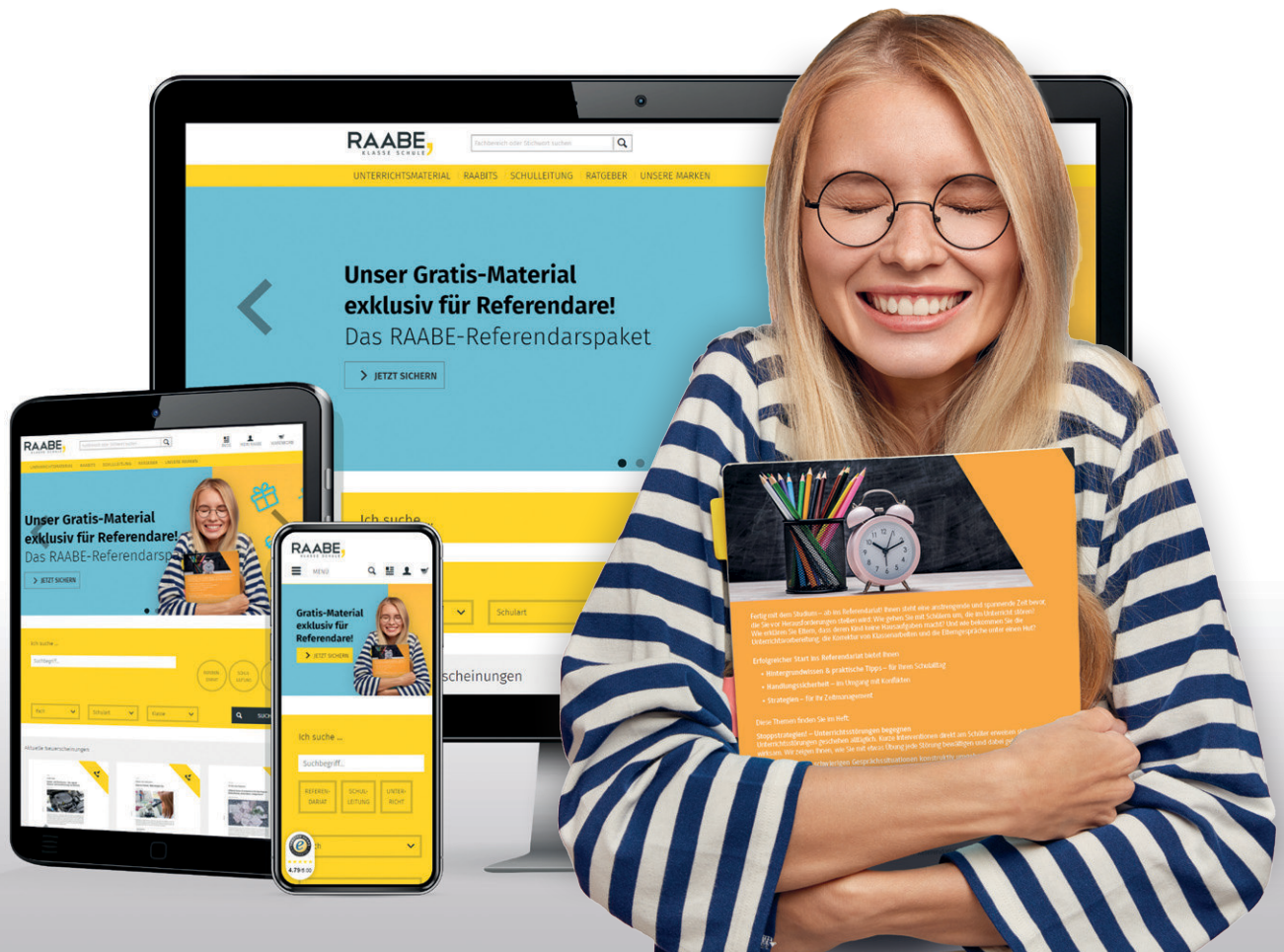
• Zu Beginn jeder Unterrichtswoche wird ein Daily Scrum mit der Dauer von 5 Minuten durchgeführt.

• In der letzten Unterrichtswoche werden Review und Retrospektive mit der gesamten Klasse durchgeführt. Für beide Meetings sind jeweils 15 Minuten vorgesehen.

• Für die Umsetzung des Projekts sind insgesamt zwei Sprints vorgesehen.

# Sie wollen mehr für Ihr Fach?

## Bekommen Sie: Ganz einfach zum Download im RAABE Webshop.



✓ **Über 5.000 Unterrichtseinheiten**  
sofort zum Download verfügbar

✓ **Webinare und Videos**  
für Ihre fachliche und  
persönliche Weiterbildung

✓ **Attraktive Vergünstigungen**  
für Referendar:innen  
mit bis zu 15% Rabatt

✓ **Käuferschutz**  
mit Trusted Shops

Jetzt entdecken:  
**www.raabe.de**

