



Technische
Universität
Braunschweig

Institut für
Flugführung



Prozessmodelle

Einführung in das Projektmanagement bei Softwareprojekten

Prof. Dr.-Ing. Peter Hecker, Dipl.-Ing. Paul Frost, 18. April 2017

Agenda

- 04. April Kick-Off
- 11. April Projektmanagement
- 18. April Prozessmodelle**
- 25. April Versionsverwaltung und Entwicklungsumgebungen
- 02. Mai Einführung Arduino/Funduino
- 09. Mai Entwicklungsumgebungen und Debugging
- 16. Mai Dokumentation und Testing
- 23. Mai Dateieingabe und -ausgabe
- 30. Mai GUI-Erstellung mit Qt
- 06. Juni Exkursionswoche
- 13. Juni Bibliotheken
- 20. Juni Netzwerke
- 27. Juni Projektarbeit
- 04. Juli Projektarbeit
- 11. Juli Vorbereitung der Abgabe

Teil I

Wiederholung

Motivation

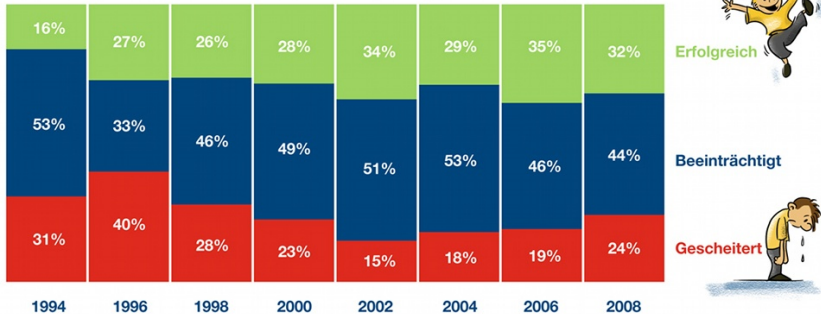


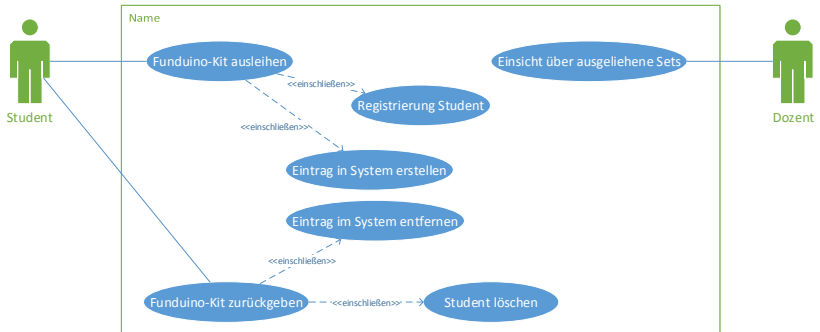
Abbildung 1: Chaos-Report

Lastenheft

- Jedes Arduino-Kit soll durch ein RFID-Tag eine eindeutige ID erhalten.
- Das Lesegerät soll mit einem Computer verbunden sein.
- Die IDs der Arduino-Kits sollen tabellarisch auf dem Computer gespeichert sein.
- Beim Lesen eines RFID-Tags sollen die tabellarisch gespeicherten Daten des Kits mit der entsprechenden ID angezeigt werden.

Lastenheft

- Über den Computer sollen Informationen in der Tabelle eingetragen/geändert werden können. Dazu zählen:
 - Gruppennummer/-name
 - Vollständige Namen und Matrikelnummer der Studenten, an die das Arduino Kit ausgeliehen wurde
 - Eine Übersicht der Bauteile, die seit einem bestimmten Datum fehlen oder defekt sind, mit Angabe von weiteren Hinweisen
- In der Tabelle sollen neben der Kit-Nummer weitere, unveränderliche Informationen angezeigt werden.
- Über die Eingabe der Kit-Nummer sollen die Daten des entsprechenden Kits auch ohne das Lesen des RFID-Tags angezeigt und bearbeitet werden können.



Teil II

Prozessmodelle



Abbildung 2: Einwurf des Balls durch Luke Burgess in ein Gedränge¹

¹By PierreSelim - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=17336884>

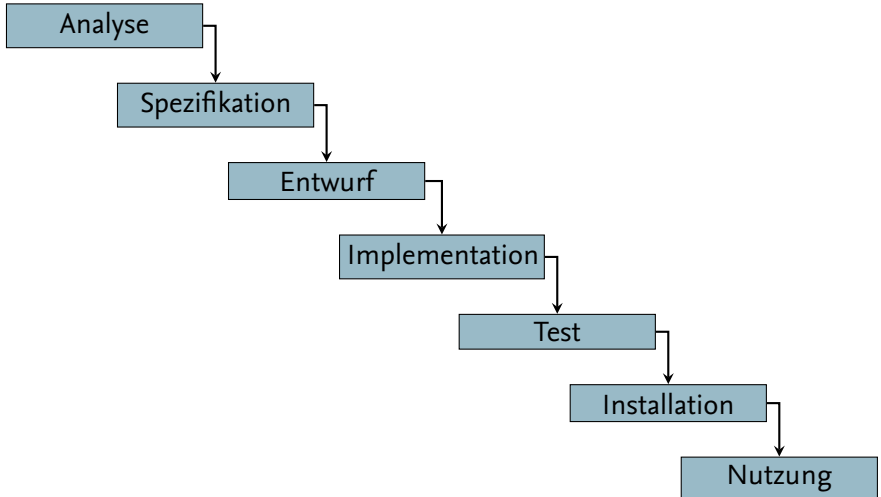
Ein simples Modell

1. Quellcode schreiben
2. Fehler beheben

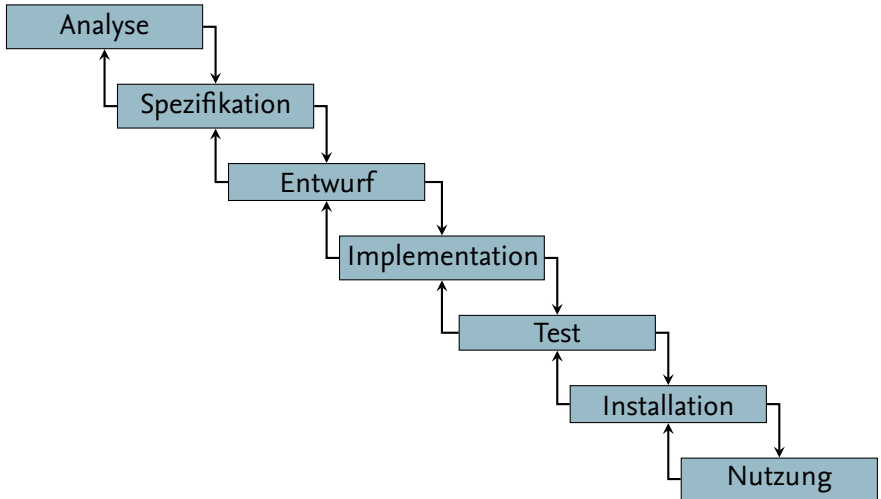
Nachteile:

- Fehlerbehebungen führen meistens zu Umstrukturierungen
⇒ Weitere Behebungen werden aufwendiger
- Wenig Akzeptanz des Produkts beim Endnutzer
- Fehleridentifikation ist sehr schwierig, weil Tests nur unzureichend vorbereitet wurden

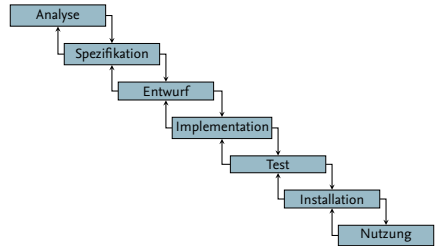
Das Wasserfallmodell



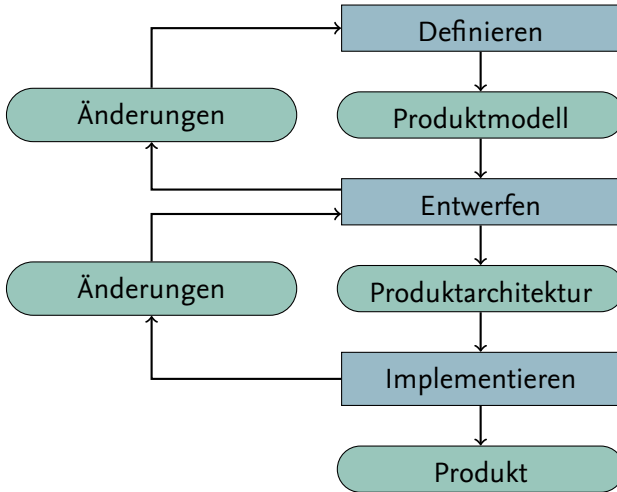
Das Wasserfallmodell



- Jede Phase wird mit einem fertigen Dokument abgeschlossen
 - Sequentielle Entwicklung
 - Am Top-Down-Vorgehen orientiert
- Es wird immer konkreter



²softtech.



Vorteile:

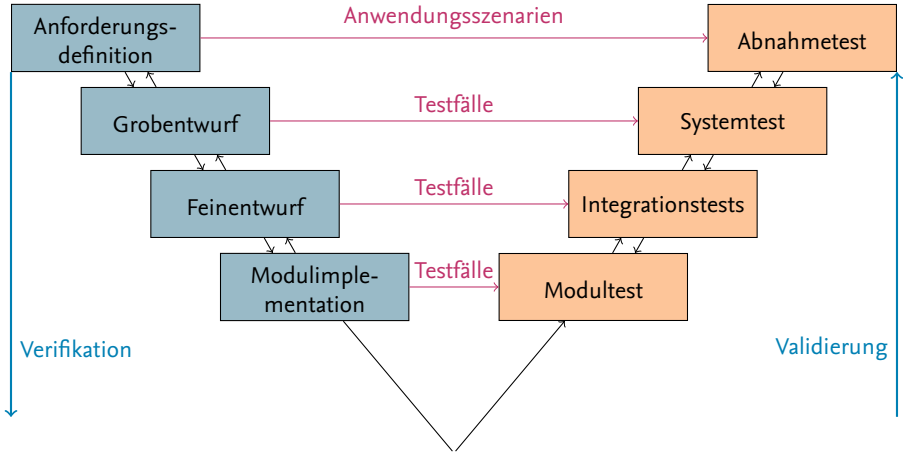
- Einfach
- Geringer Managementaufwand

Nachteile:

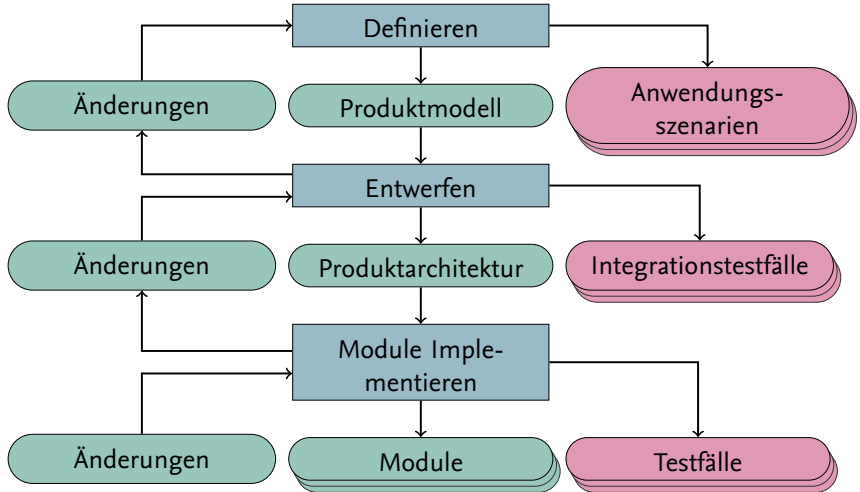
- Nicht immer sinnvoll Phasen komplett abzuschließen
- Nicht immer sinnvoll alle Phasen sequentiell abzuarbeiten
- Dokumente haben zum Teil eine höhere Priorität als das Produkt
- Risiken könnten vernachlässigt werden, da der festgelegte Ablauf auch in der Form durchgeführt wird

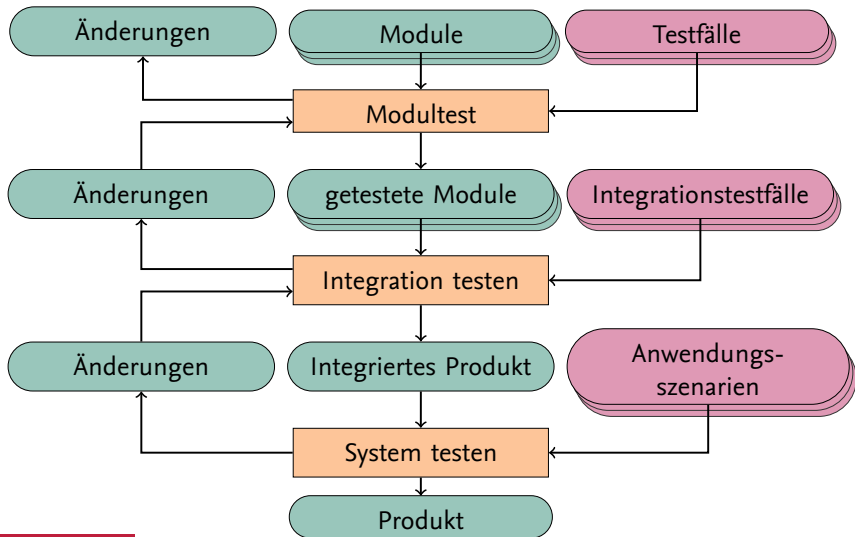
³softtech.

Vorgehensmodell (V-Modell)



- Diente als Vorlage für die Vorgehensmodelle bei der Bundeswehr und Behörden
- Dokumentengetriebenes Modell
- Erweiterung des Wasserfallmodells um Qualitätssicherung
- Das V-Modell ist unterteilt in
 - Systemerstellung
 - Qualitätssicherung
 - Konfigurationsmanagement
 - Projektmanagement





Vorteile:

- Generisches Modell, das angepasst werden kann
- Ermöglicht standardisiertes Vorgehen
- Gut geeignet für große Projekte

Nachteile:

- Sehr aufwendig
- Sehr bürokratisch
- Ohne Computer Aided Software Engineering (CASE)-Unterstützung nicht handhabbar
- Sehr viele Rollen erforderlich

Prototypen-Modell

- Anforderungen herausfinden
- Diskussionsbasis und Hilfe bei Entscheidungen
- Sammlung von praktischen Erfahrungen

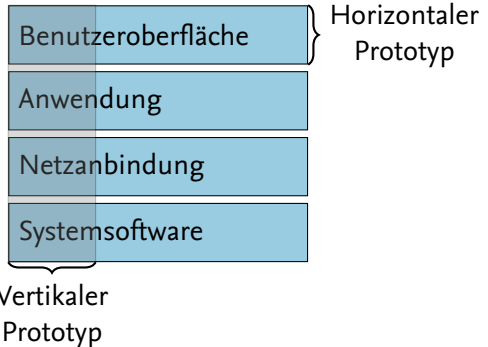
Unterscheidung von Prototypen

- Prototyp zur Klärung von Fragen
- Prototyp zur Erstellung der Produktdefinition
- Prototyp zur inkrementellen Weiterentwicklung

Prototypen-Modell

Horizontaler Prototyp

- Realisiert spezifische Ebenen
- Ebene wird vollständig realisiert



Vertikaler Prototyp

- Umfasst alle Ebenen des Systems
- Ebenen werden partiell realisiert

Vorteile:

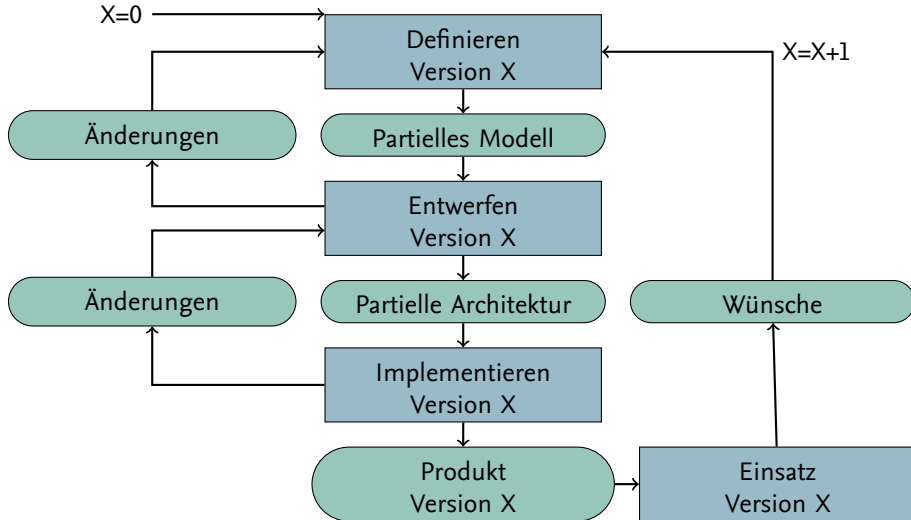
- Geringeres Entwicklungsrisiko
- Bessere Planbarkeit
- Unterstützung von anderen Prozessmodellen
- Benutzer können besser einbezogen werden

Nachteile:

- Mehr Aufwand
- Ein unfertiger Prototyp könnte zum Endprodukt werden
- Oft nicht eingegrenzt

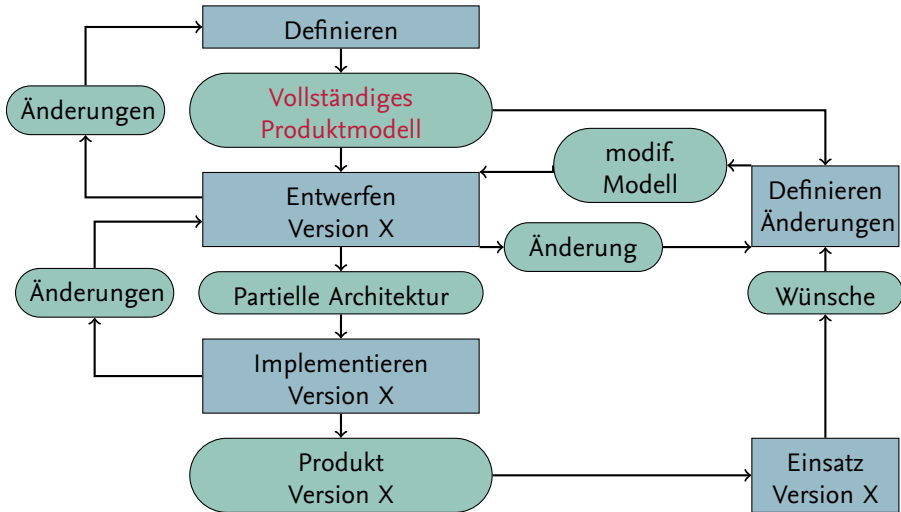
Evolutionäres Modell

- Stufenweise Entwicklung
- Gut für nicht komplett generierbare Anforderungen
- Konzentration auf lauffähige Teilprodukte



Inkrementelles Modell

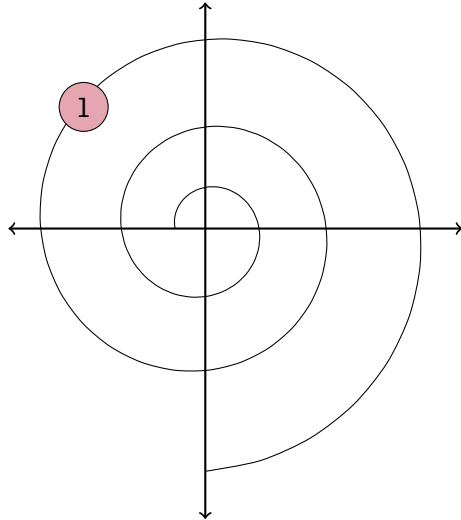
- Aufgebaut wie das evolutionäre Modell
aber
- Anforderungen werden vollständig erfasst und modelliert
- Nur ein Teil der Anforderungen wird umgesetzt



Spiralmodell

Schritt 1:

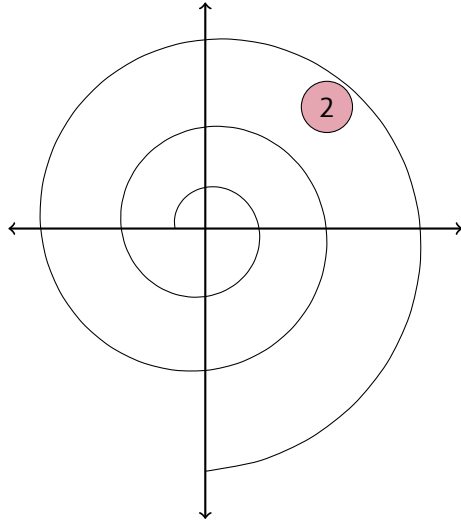
- Identifikation der Ziele
- Alternative Möglichkeiten
 - Entwurf A, Entwurf B
 - Wiederverwendung
 - Kauf
- Randbedingungen ausarbeiten
 - Kosten
 - Zeit
 - Schnittstellen



Spiralmodell

Schritt 2:

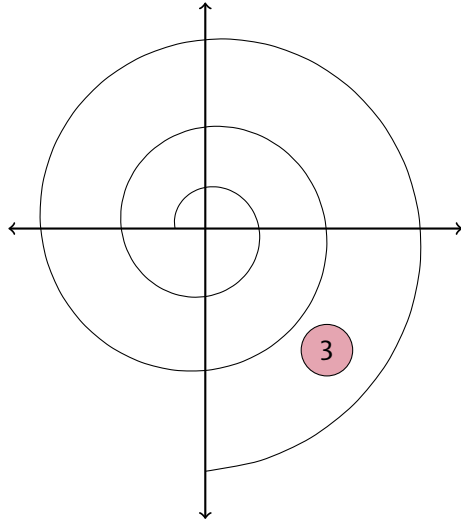
- Evaluierung der Alternativen
- Bei Risiken: Entwicklung einer Strategie
 - Prototypen
 - Simulationen
 - Benutzerbefragungen



Spiralmodell

Schritt 3:

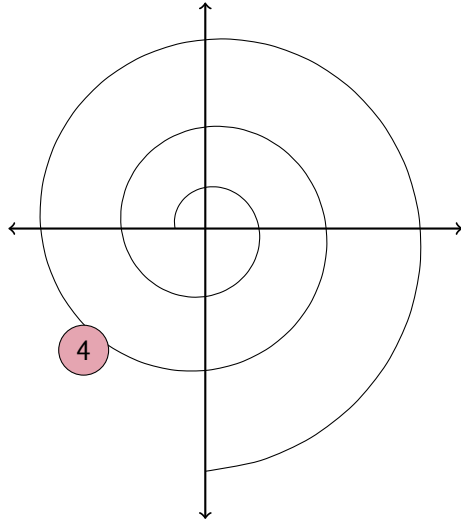
- Risiken \Rightarrow Wahl eines Prozessmodells
 - evolutionäres Modell
 - Prototypenmodell
 - Wasserfall-Modell
 - ...
- Kombinationen denkbar



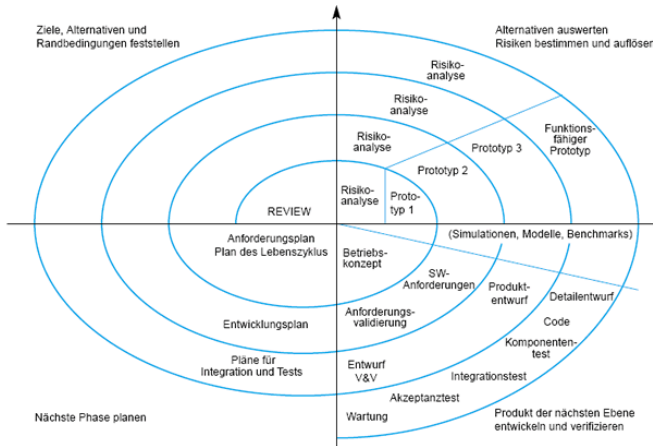
Spiralmodell

Schritt 4:

- Planung des nächsten Zyklus
 - Ressourcen
 - Aufteilung in Komponenten
- Review der Schritte 1-3
- Einverständnis über den nächsten Schritt



Spiralmodell



Vorteile:

- Periodische Überprüfung kann ein Abdriften von Zielen und Risiken verhindern
- Je nach Zyklus kann ein geeignetes Prozessmodell ausgewählt werden
- Flexibel
- Erfahrungen können im nächsten Zyklus eingebracht werden

Nachteile:

- Hoher Managementaufwand für kleine und mittlere Projekte
- Risiken müssen identifizierbar sein

Scrum

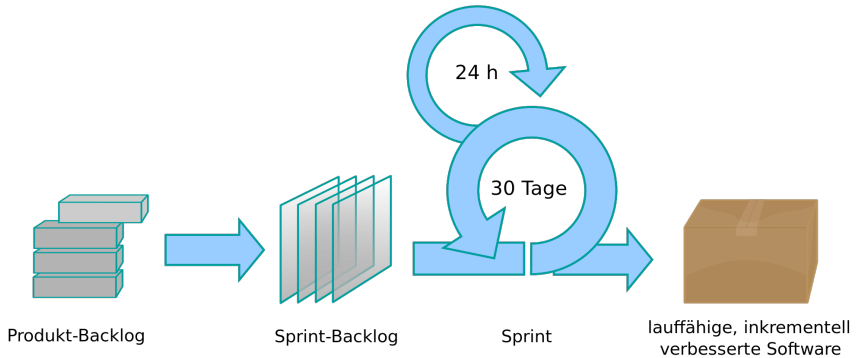
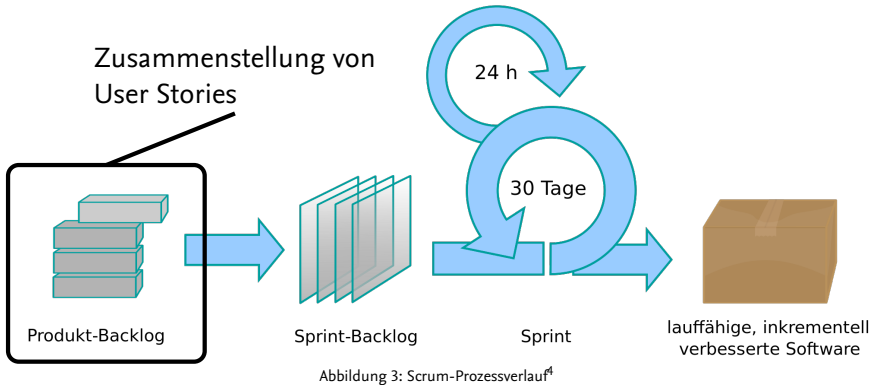


Abbildung 3: Scrum-Prozessverlauf⁴

⁴Von Scrum_process.svg: Lakeworksderivative work: Sebastian Wallroth (talk) - Scrum_process.svg, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=10772971>

Scrum



⁴Von Scrum_process.svg: Lakeworksderivative work: Sebastian Wallroth (talk) - Scrum_process.svg, CC BY-SA 3.0, commons.wikimedia.org/w/index.php?curid=10772971

Scrum

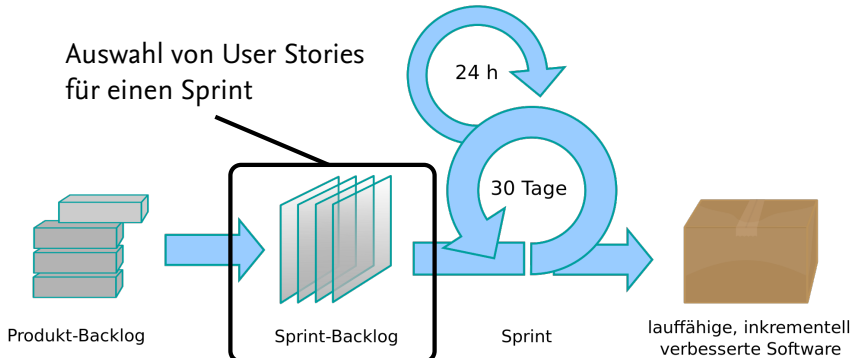
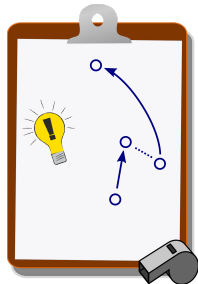


Abbildung 3: Scrum-Prozessverlauf⁴

⁴Von Scrum_process.svg: Lakeworksderivative work: Sebastian Wallroth (talk) - Scrum_process.svg, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=10772971>

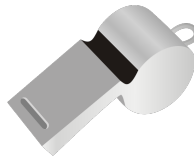
Product owner:

- Verkörpert die Projektidee
- Maximiert den Wert des Produktes
- Vermittelt die Vision an das Team
- Stellt das Team zusammen
- Erstellung eines Produkt-Backlogs
- Priorisierung des Produkt-Backlogs



Scrum Master:

- Treibt den Prozess
- Wahrung und Vermittlung der Scrum-Werte und -Regeln
- Schützt das Team vor Störungen
- Löst von Blockaden
- Vermittelt zwischen Team und Product Owner



Entwicklungsteam (3-9 Personen):

- Selbst organisierend
- Interdisziplinär
- Liefert Produkt
- Arbeitet eigenständig an den User Storys eines Sprints
- Implementiert und testet Anforderungen



Prozessmodell Überblick

Prozessmodell	Primäres Ziel	Antreibendes Moment
Wasserfall-Modell	minimaler Managementaufwand	Dokumente
V-Modell	maximale Qualität	Dokumente
Prototypen-Modell	Risikominimierung	Code
Evolutionäres-Modell	minimale Entwicklungszeit	Code
Inkrementelles-Modell	minimale Entwicklungszeit	Code
Objektorientiertes-Modell	Zeit- und Kostenminimierung	Wiederverwendung
Spiralmodell	Risikominimierung	Risiko

API-Prozessmodell

Ziel

- Risiko minimieren
- Minimale Entwicklungszeit
- Wenig Bürokratie
- Geeignet für Neueinsteiger

Auswahl

- Ein für API angepasstes Spiralmodell
- Software-Entwicklung soll evolutionär oder inkrementell erfolgen

Teil A:

- Ziele für den kommenden Zyklus formulieren

Teil B:

- Risiken formulieren (falls vorhanden)
- Maßnahmen ausarbeiten und ggf. Risiken einbeziehen
- Kleinere Prototypen entwickeln

Teil C:

- Ziele ausarbeiten
 - Feinentwurf und/oder
 - Programmieren und/oder
 - Prototypen entwickeln

Teil D:

- Review des aktuellen Zyklus (Ziele erreicht? Qualität?)
- Planung folgenden Zyklus

Ziele	Matrikel-Nr., Vorname und Name über Studentenausweis einlesen
Risiken	Daten sind nicht auslesbar
Maßnahmen	a) Prototyp: Testaufbau erstellen b) Auslesen des Studentenausweises testen c) ggf. andere Karte auslesen
Ergebnisse	a) Testaufbau fertiggestellt b) Studentenausweis kann nicht ausgelesen werden c) Auslesen der anderen Karten erfolgt problemlos
Review	<i>Inhalte hierfür folgen am 16.05.17</i>
Nächster Zyklus	Verknüpfung zwischen Student und Funduino-Kit

Teil III

Projektarbeit

Aufgabe 1

1. Erstellen Sie im Wiki Ihres Projekts die Seite **Prozessmodell**
2. Beginnen Sie den ersten von drei Zyklen zu planen und füllen Sie das API-Spiralmodell so weit es geht aus
3. Passen Sie Ihren Zeitplan so an, dass drei Zyklen erarbeitet werden können

GIT Installieren

Aufgabe 2

1. Git herunterladen
<https://git-scm.com/downloads>
2. Git installieren