

Scrum

Done Criteria

FDA

Product Backlog

EC 62304
Medizinische Software

Agile

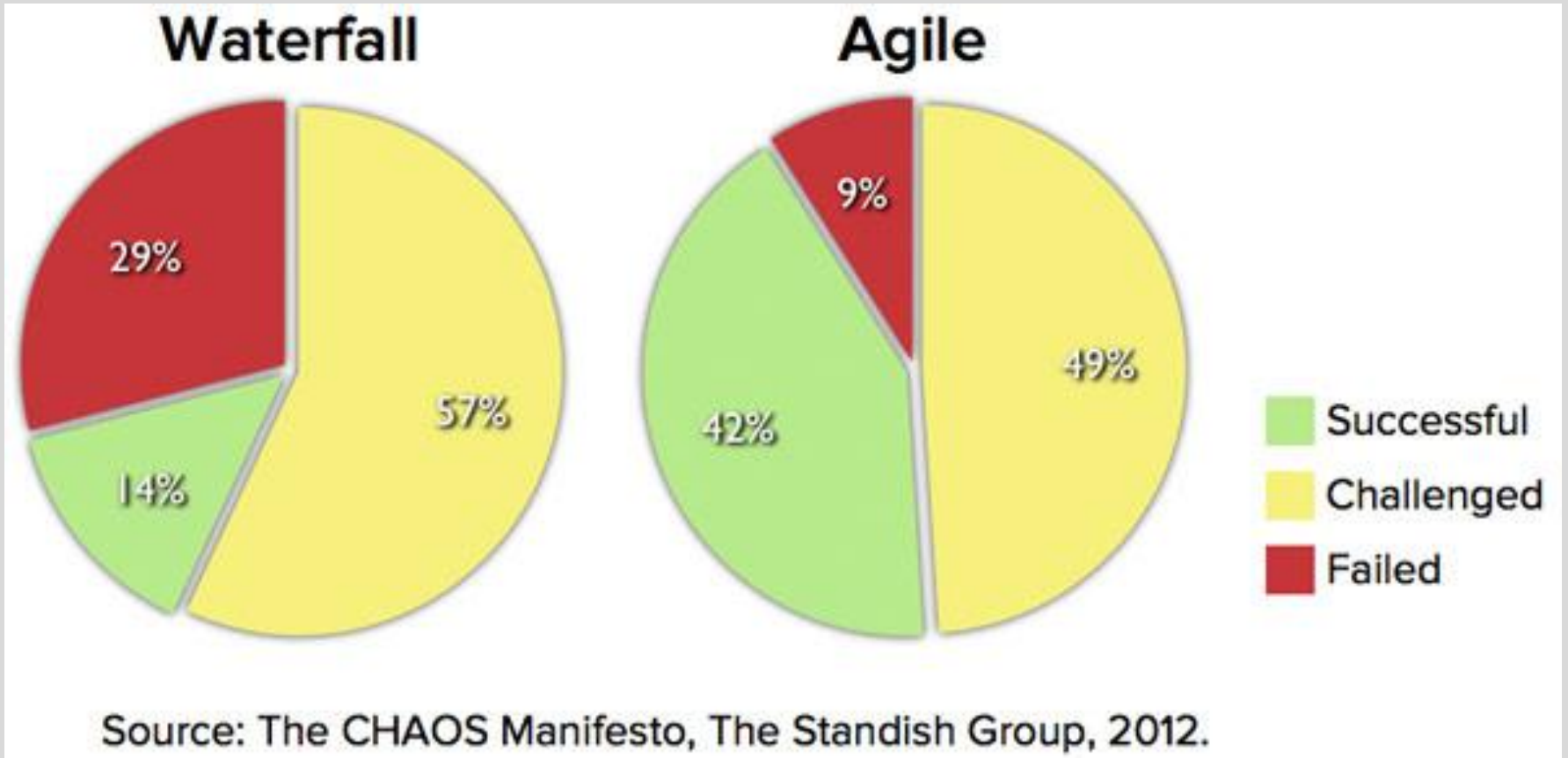
User Story

Water-Scrum-Fall

Ein Entwicklungsprozess mit Zukunft?

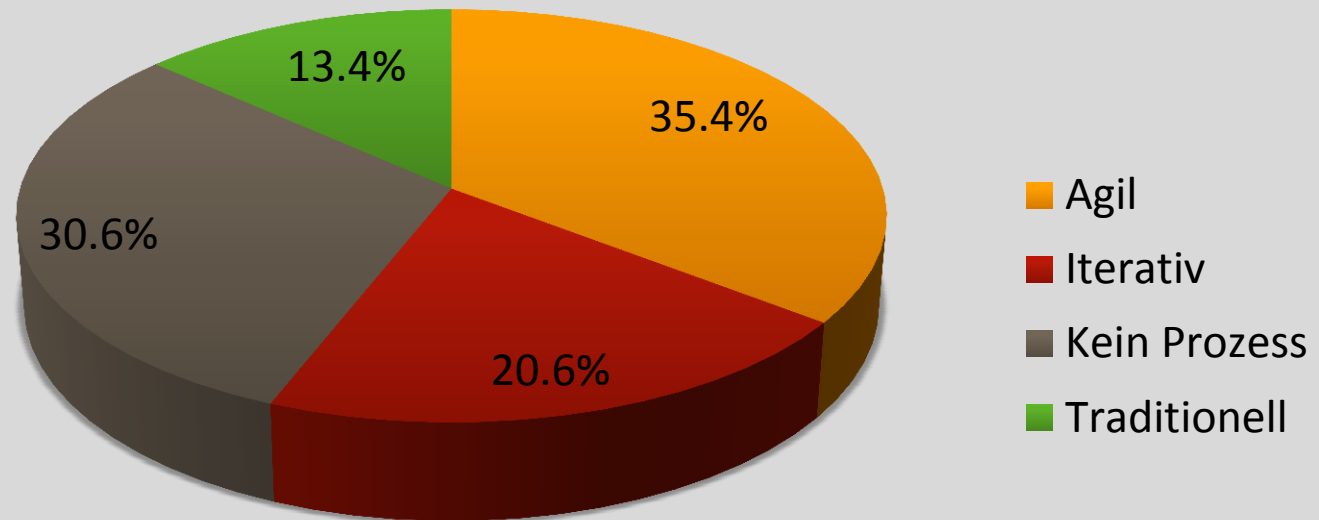
Bernhard Fischer

Waterfall vs. Agile: Eine Erfolgsstory



Umsetzung agiler Prinzipien

Entwicklungsprozess 2009



Basis: Forrester, 2010

SCRUM

Anteil > 10%



Gründe für den Einsatz agiler Methoden

- Probleme Projektmanagement: Entwicklungsprojekte können zu Projektbeginn detailliert geplant werden.
- Probleme im Anforderungsmanagement: Anforderungen können und sollen zu Projekt-beginn vollständig erhoben werden
- Probleme bei der Entwicklung und dem Messen des Entwicklungsfortschritts: Phasenweise Entwicklung bereitet Probleme beim Messen des Entwicklungsfortschritts
- ...

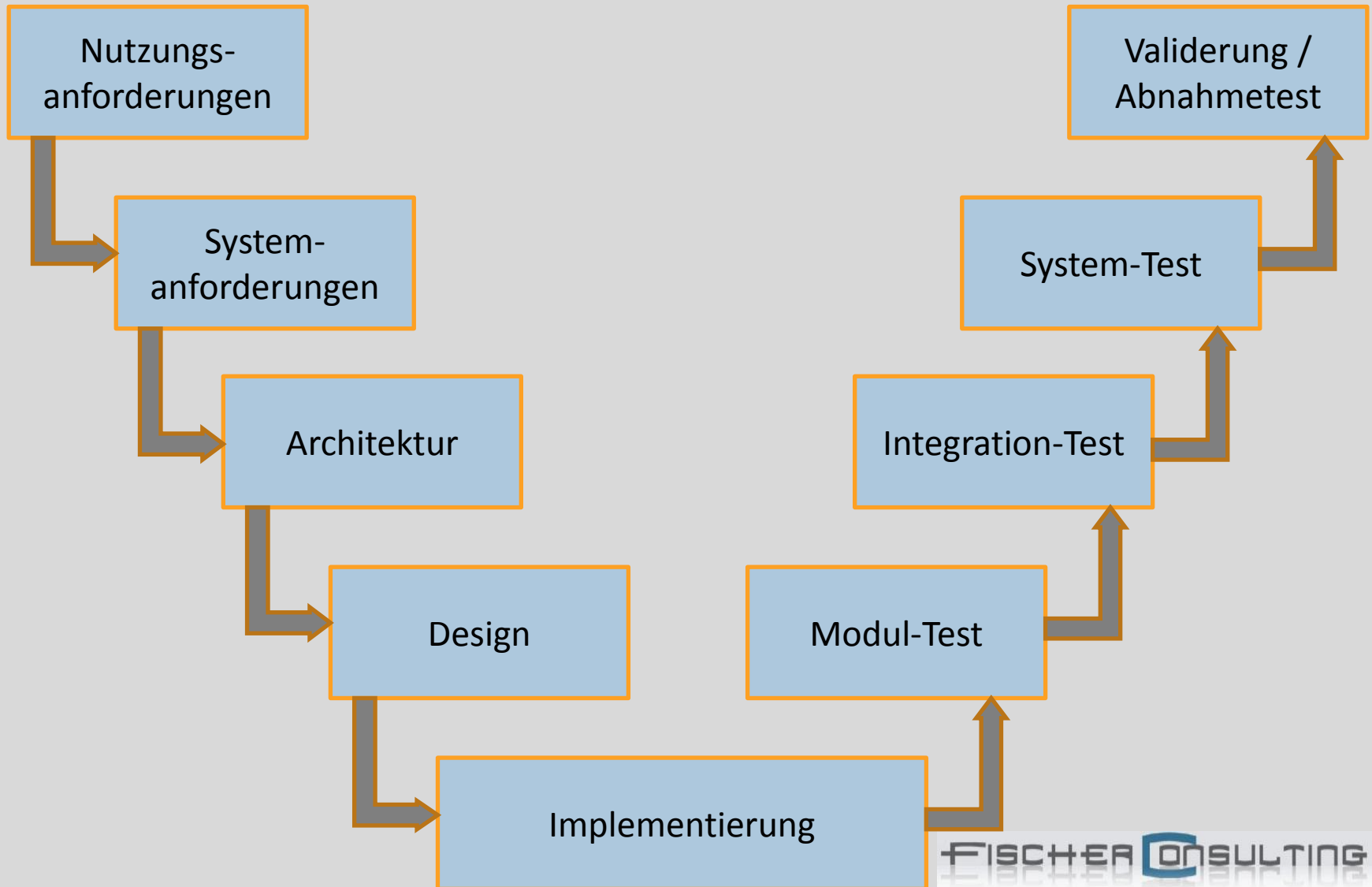
Und das Ergebnis?

- Agile Vorgehen wird als Sammlung isolierter Methoden betrachtet, die konkrete Probleme lösen
- Agile Methoden sollen nur isolierte Schwachstellen beseitigen, damit werden nur Symptome gelindert
- Agile Methoden werden daher häufig in traditionelle wasserfallartige Vorgehensmodelle eingearbeitet
- Angst für eine nicht regulatorisch konformen Lösung schreckt ab

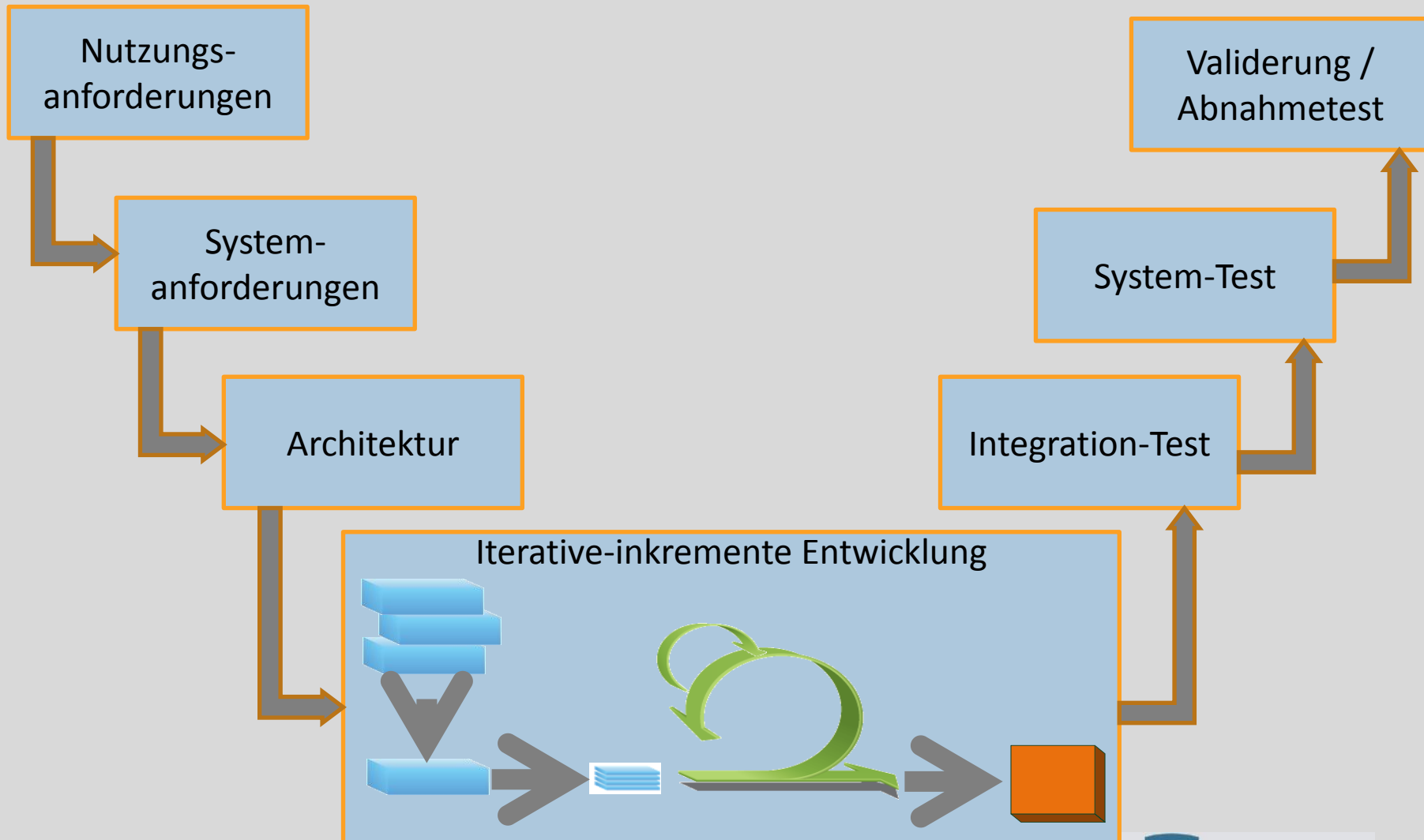
... und wir sehen nur ScrumButs ...(1)

- Aus Business Analysten werden Product Owner
- Teammitglieder arbeiten an mehr als einem Projekt
- Team werden nach Projektende auseinandergerissen
- Die Scrum-Rollen werden nicht besetzt
- Done- und Ready-Kriterien für Stories werden nicht berücksichtigt
- Der Prozess / QM fordert sinnlose Dokumente

Das V-Modell



Wasserfallmodell mit agilen Aspekten



Anforderungen u. Architektur im V-Modell

- Nutzungsanforderungen werden nicht systematisch erhoben
- Einbeziehen von Anwendern in die Entwicklung erfolgt nicht
- Systemanforderungen werden unnötig detailliert erhoben und dokumentiert
- Die Architektur wird nicht vom Team festgelegt, ist aber festgeschrieben
- Das Entwicklungsteam ist nicht an der Erstellung der Anforderungen beteiligt

Iterative Entwicklung im unteren V

- Das Entwicklungsteam ist nicht funktionsübergreifend besetzt, insbesondere fehlen Tester im Team
- Die Entwickler arbeiten nicht als Team und das Entwicklungsteam ist nicht selbst-organisierend
- Scrum-Rollen werden nicht besetzt oder die Rollen sind nicht befähigt
- Mikro-Management, kein Vertrauen in das Team
- Überflüssige Dokumentation und unzureichende Dokumentation

Verifizierung und Validierung im V-Modell

- Inkremente erfüllen nicht die Done-Kriterien, oder es werden gar keine Done-Kriterien definiert
- Fehler, die im V&V festgestellt werden führen zu Verzögerungen
- Die Entwicklung dauert länger / ist teurer als erwartet.
- Technische Schulden machen Änderungen teuer / schwieriger

Gesamtergebnis:

Verschenkte Chancen!

Was können wir daraus machen?

- Sie können die Vorteile inkrementellen und iterativen Vorgehens konkret belegen.
- Sie haben bewiesen, dass der Fokus auf lauffähige Software tragfähig ist
- Sie haben gezeigt, dass es sinnvoll die Entwicklung ist die Hände eines entsprechend besetzten Entwicklungsteams zu legen
- Sie haben konkret gezeigt dass die Entwicklung medizinischer Software mit agilen Methoden möglich ist!

Konsequenz

Nutzen Sie die bisherigen Erfahrungen mit agilen Methoden um bei der Umsetzen eines agilen Vorgehensmodell weiterzukommen!

Agiles Projektmanagement

- Entwicklung als empirischer Prozess
- Iterative und inkrementelle Entwicklung
- Größen-Schätzungen durch das Team
- Zweistufige Schätzung (Release vs. Iteration)
- Fokus auf Ergebnissen: Überprüfung der Fertigstellung nach jeder Iteration
- Fokus auf Nützlichkeit : Orientierung an der Fertigstellung von Teilfunktionalitäten

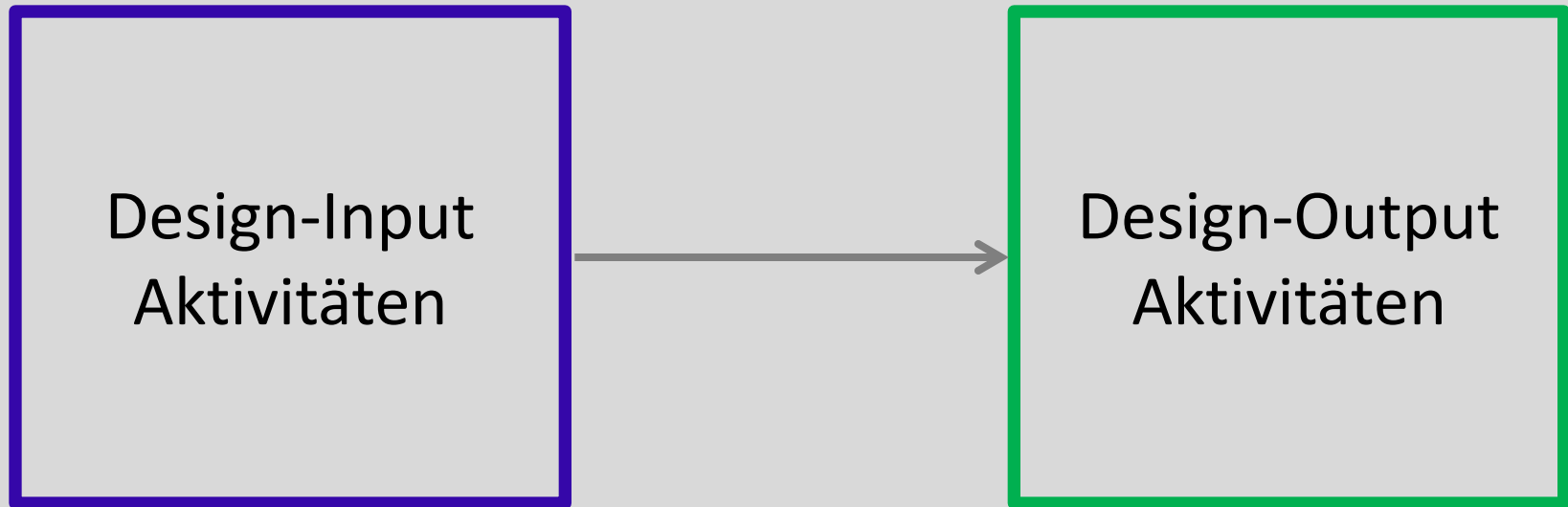
Agiles Requirement Engineering

- Einbeziehung von Kunden und Anwendern über die gesamte Projektlaufzeit hinweg
- Regelmäßiges Feedback über die erstellten Produktinkremente einholen
- Formulierung der Systemfunktionalität aus Kundensicht unter Berücksichtigung der Nutzungsanforderungen
- Priorisierung der Anforderungen nach Kundenwert
- Unterschiedliche Granularität der Anforderungen und Detaillierung nach Notwendigkeit

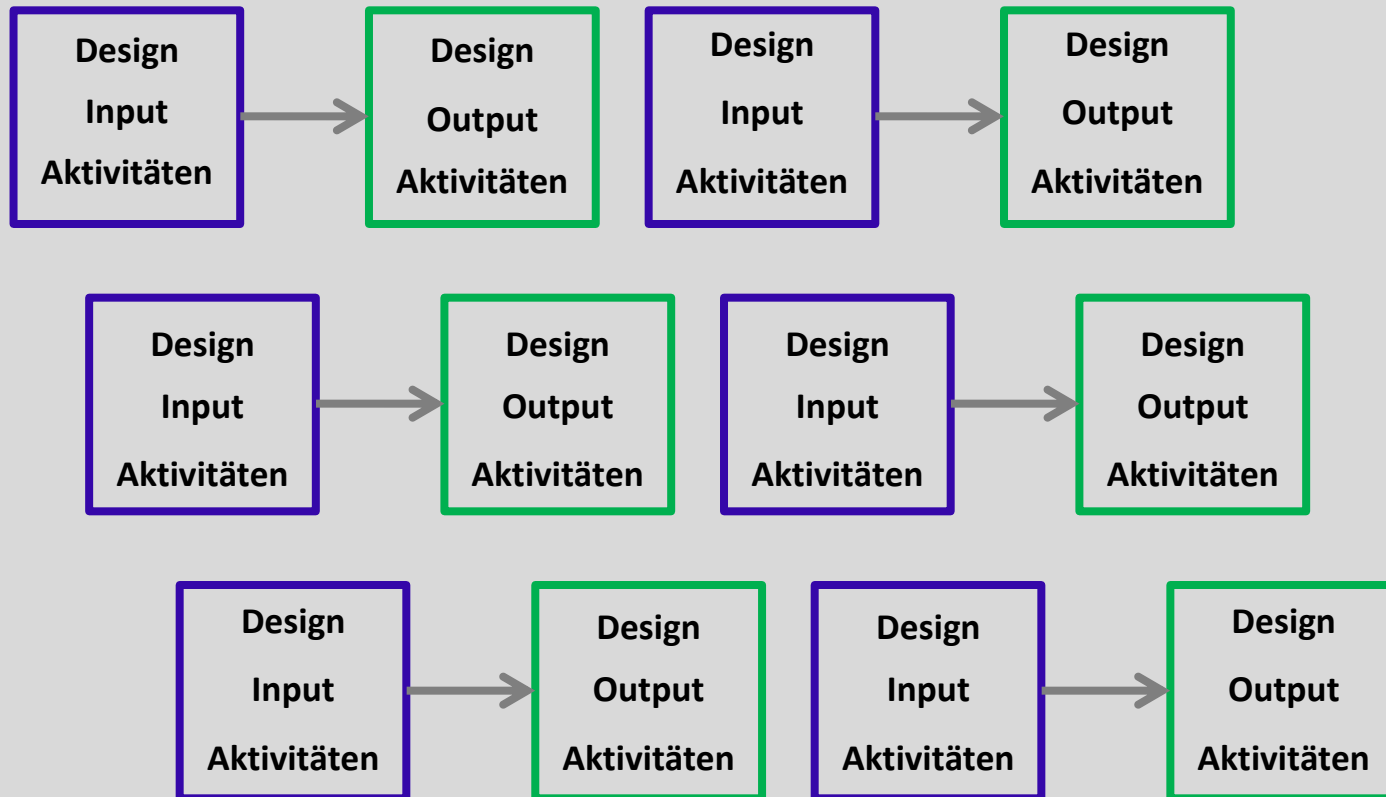
Agiles Entwickeln und Testen

- Iterative und inkrementelle Entwicklung
- Definition Done-Kriterien und Überprüfung der Artefakte auf Einhaltung der Done-Kriterien
- Fokussierung auf die Fertigstellung von Teilfunktionalitäten
- TDD auf Anforderungsebene und Codeebene
- Inkrementelles Design
- Testen ist in die Entwicklung integriert
- Funktionsübergreifend besetztes Team

Vom Design-Input zum Design-Output



Aufbrechen der Input und Output Aktivitäten



Entwicklung auf dem Story/Item-Level

5.1 Planung – der einzelnen Story

5.2 Requirements – Ermittlung der Story Details

5.3 Architektur – Emergent bei der Entwicklung

5.4 Design – inkrementell und iterativ

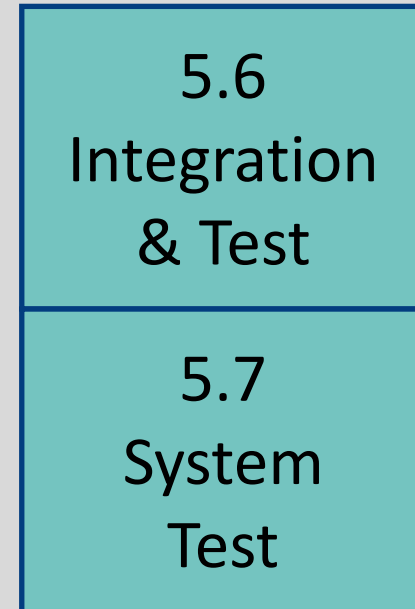
5.5 Implementierung und Verifikation der SW-Einheiten(n)

5.6 SW-Integration und Integrationstest

5.7 Systemtest

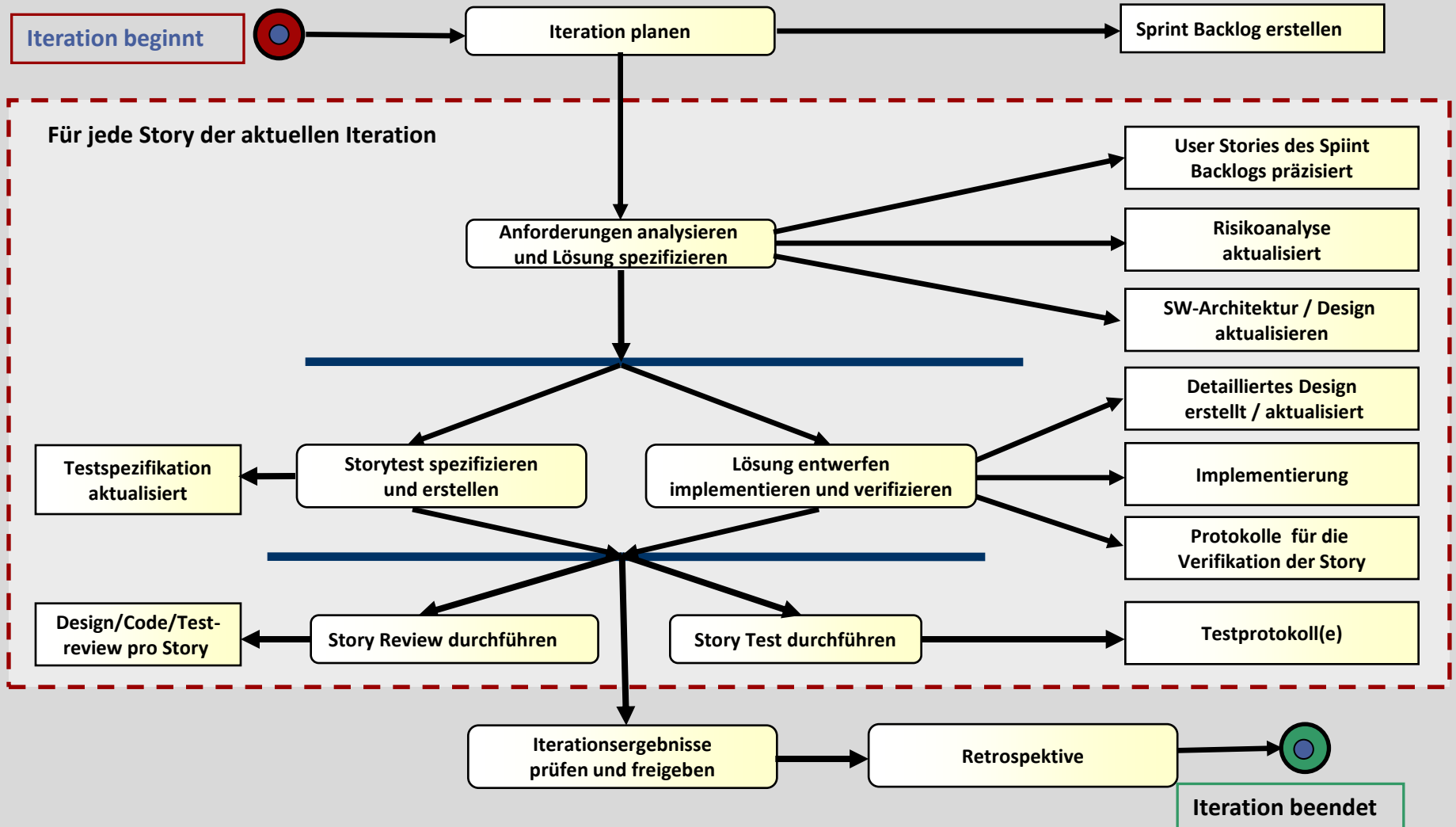
Für jede Story

Entwicklung auf dem Iterations-Level

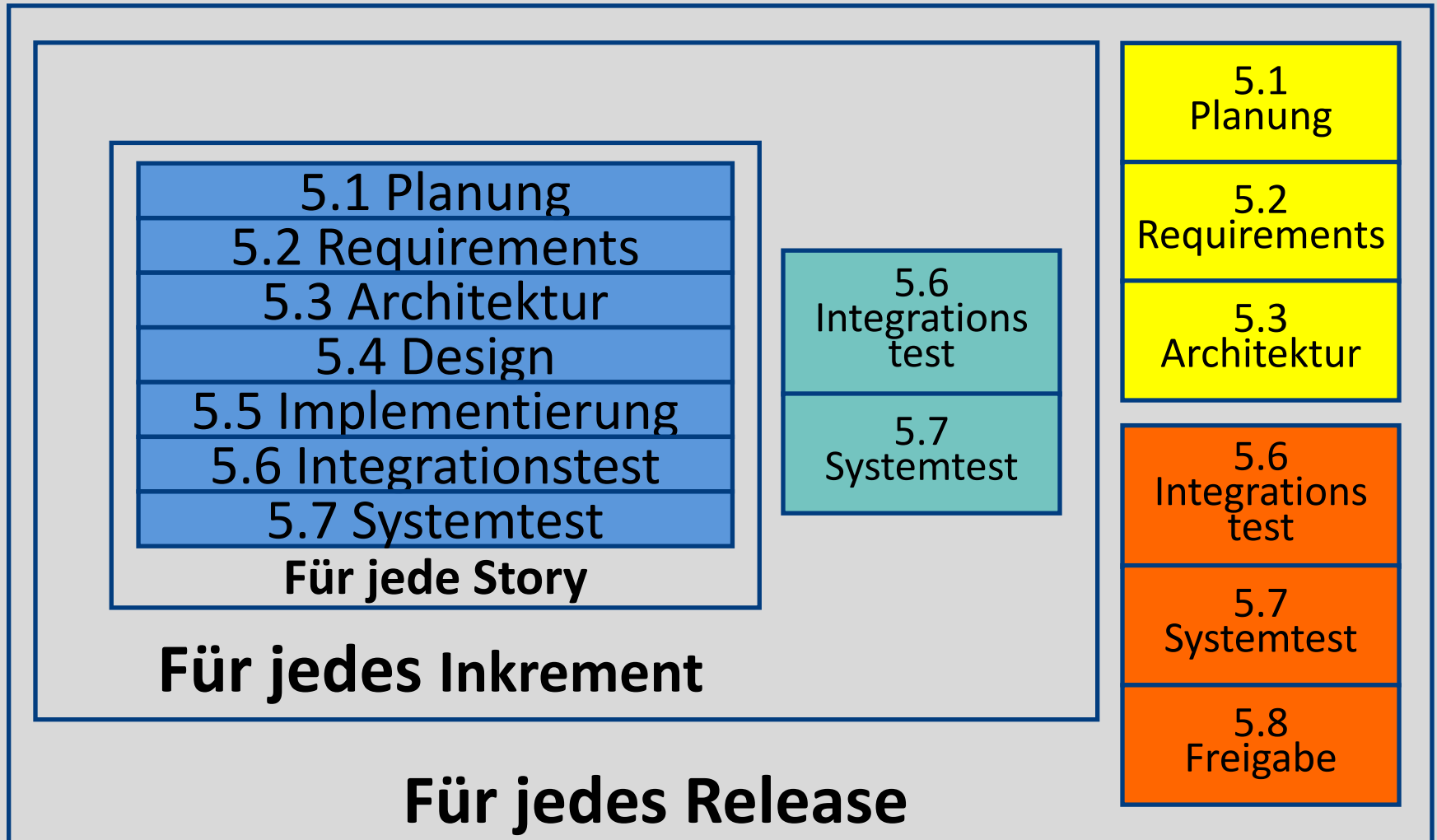


Für jede Iteration / jedes Inkrement

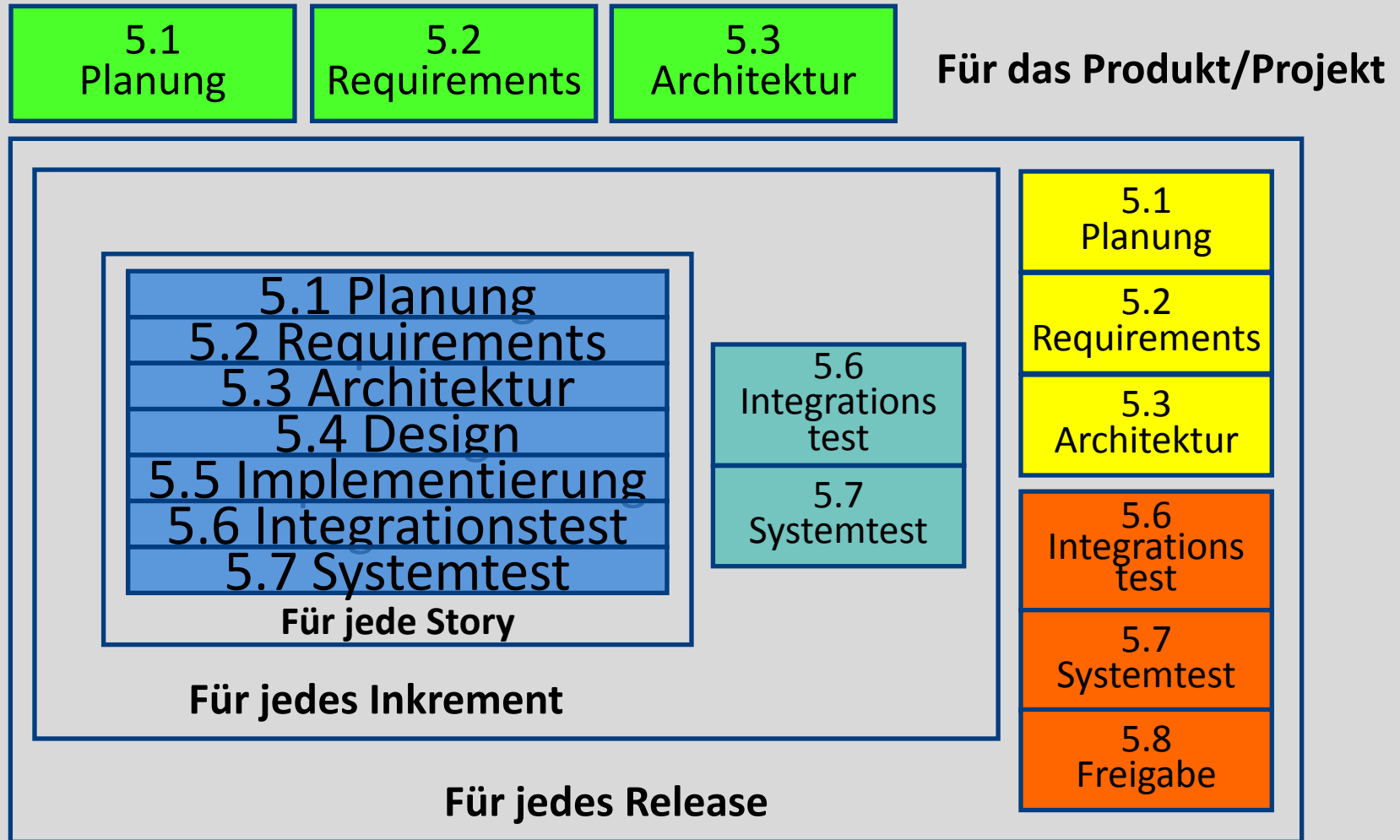
Ablauf einer Iteration: Beispiel



Entwicklung auf dem Release-Level



Entwicklung auf dem Produkt-Level



Fazit

- Water-Scrum-Fall kann bei der Einführung agiler Methoden ein Zwischenschritt sein, um internen Problemen zu begegnen.
- Häufig führen aber solchen Hybrid-Methoden dazu, dass die agile Umsetzung unvollständig bleibt
- Die Einführung agiler Methoden und Techniken führt nicht zwangsläufig agilen Denken.
- „Agil“ ist nicht eine Sammlung von Methoden und Techniken, sondern ein Wertesystem. Rechnen Sie mit mehreren Jahren für die Etablierung eines Wertesystems!

Thank you!

Fischer Consulting GmbH
Bernhard Fischer
Waldstr. 106
D-44869 Bochum

www.bfischer-consulting.de
info@bfischer-consulting.de

