

Agile Software-Entwicklung: Überblick und Techniken

Prof. Dr. Stefan Kowalewski
Dr. Carsten Weise

Kapitel I

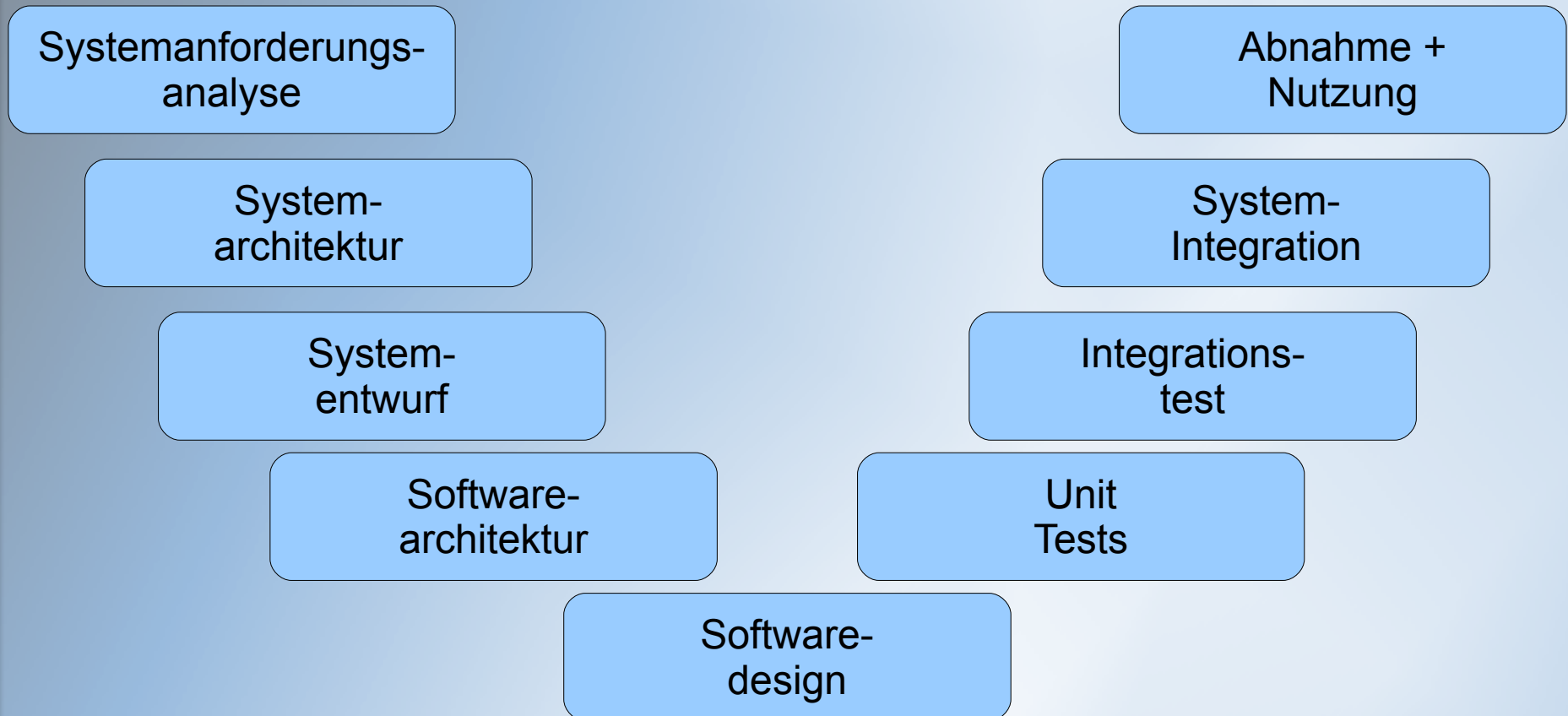
Der agile Ansatz

Agilität

- > agil = flink, beweglich
- > geringer bürokratischer Aufwand
- > wenige Regeln

Das schwache, flexible überlebt (Tarkowskij: Stalker)

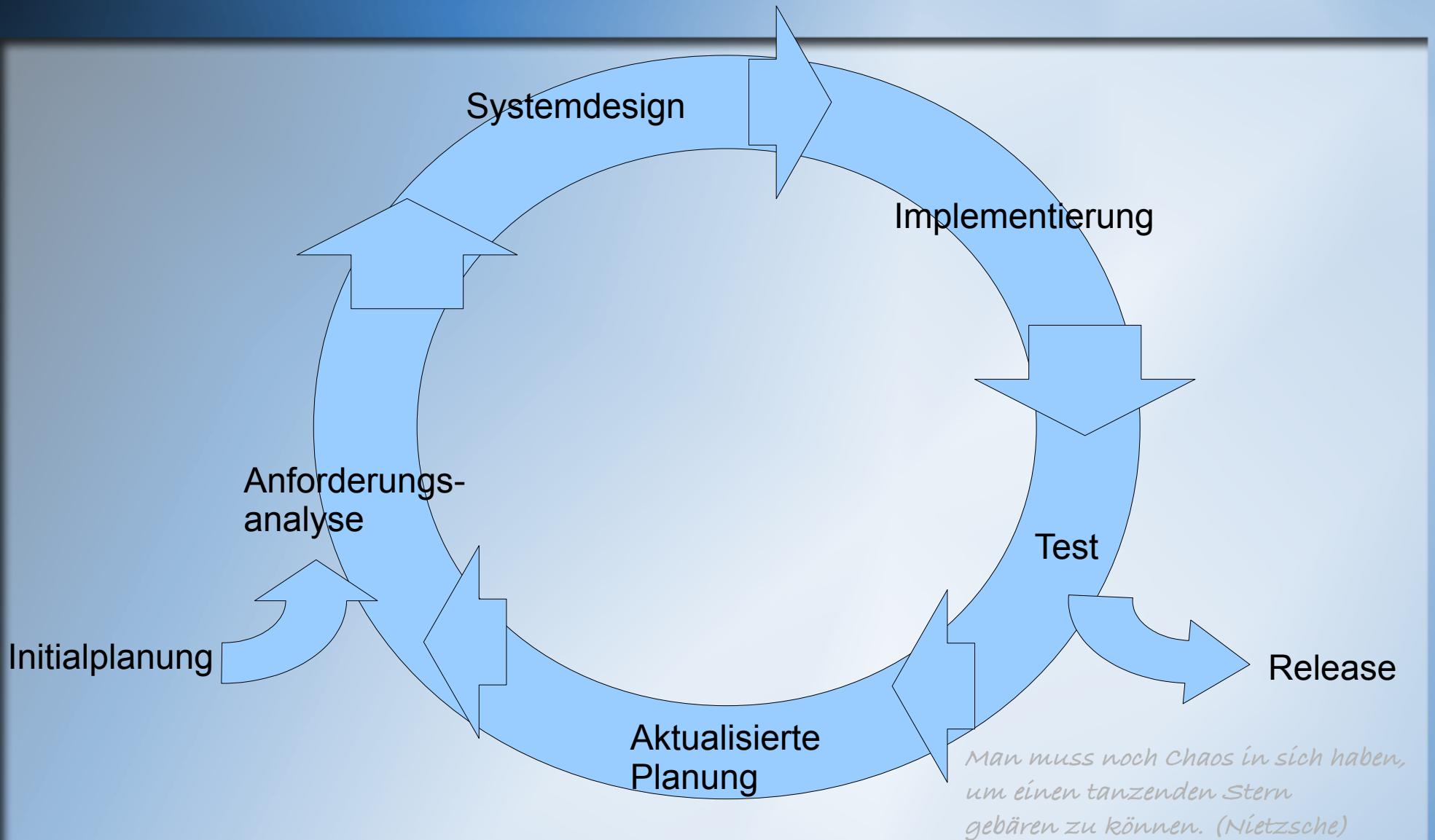
Klassisch: Wasserfallmodell



Kann man so Software entwickeln?

- > viele Softwareprojekte scheitern
- > Software ist inkrementell
- > Software Designer denken inkrementell
- > Software hat viel mehr Freiheitsgrade als Hardware

Der agile Blick



Agiles Manifest

- > Individuen und Interaktionen sind wichtiger als Prozesse und Werkzeuge.
- > Funktionierende Programme sind wichtiger als ausführliche Dokumentation.
- > Die stetige Abstimmung mit dem Kunden ist wichtiger als die ursprüngliche Leistungsbeschreibung in Verträgen.
- > Der Mut zu und die Offenheit für Änderungen stehen über dem Befolgen eines festgelegten Plans.

Wie sieht das in der Praxis aus?

- > Kommunikationslastig
 - > innerhalb des Teams
 - > zwischen Team und Kunde
- > Veränderungslastig
- > Kurze, wiederkehrende Arbeitszykel

Werte und Prinzipien

- > Fortlaufende Iterationen, Entwicklung in kleinen Schritten.
- > Am Ende jeder Iteration existiert fertige und testbare Software.
- > Feedback: Fehlentwicklungen, Mängel an der Benutzeroberfläche, mangelnde Stabilität oder Performance zeitnah erkennen und in nachfolgenden Iterationen korrigieren.
- > Anforderungen sind unklar: der Kunde weiß nicht, was er will, hat widersprüchliche und unvollständige Anforderungen, kein Projektbeteiligter hat den Gesamtüberblick.
- > Mini-Design und Refactoring.
- > KISS (Keep It Stupid And Simple).
- > YAGNI (You Aren't Gonna Need It).

Methoden

- > Extreme Programming (XP).
- > Nightly Builds.
- > Pair-Programming.
- > Prinzip der gemeinsamen Verantwortung (collective code ownership).
- > Testgetriebene Entwicklung (Test Driven Design).
- > regelmäßige Strukturverbesserungen (Refactoring).
- > gemeinsame Reflektion (Code-Reviews).

Der agile Entwickler

- > **Aktivität:** Initiative ergreifen, handeln statt abwarten, Verantwortung für die Projektziele übernehmen.
- > **Disziplin:** vereinbarte Grundregeln einhalten.
- > **Gemeinnützigkeit:** Teamgeist, keine Starallüren.
- > **Kommunikation:** Offen, klar und sachlich Meinungen vertreten, Fehler konstruktiv angehen, Informationen breit streuen, Know-how austauschen.
- > **Dynamik:** Veränderungen und Verbesserungen mittragen und fördern. Neue Wege ausprobieren.
- > **Selbstkritik:** Eigene Fehler zugeben, Hilfe anderer Teammitglieder annehmen. Vertrauens- und respektvolle Atmosphäre im Team.
- > **Courage:** Probleme die morgen auftreten, auch morgen lösen.

Testgetriebene Entwicklung

- > Testfälle zuerst
- > ändere den bestehenden Code mit geringstmöglichem Aufwand um die Testfälle zu bestehen
- > Aufräumen: Refactoring, Stabilisierung
- > wiederhole

Agile Dokumentation

- > Der Code ist die Dokumentation?
- > Prozess-Dokumentation, insbesondere Dokumentation der sich ändernden Anforderungen, muss zeitnah erfolgen
- > Software-Dokumentation im Code, aber mit Werkzeug-Unterstützung
- > Benutzer-Dokumentation entsteht aus Anforderungen: Use-Cases und Testfälle

Kapitel II

Agilität in der Praxis

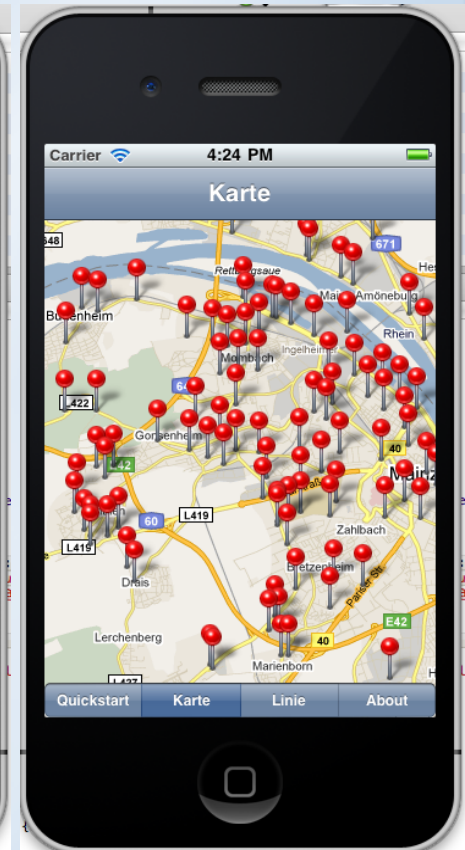
Success Story I (a)

- Webapp Leitstand + iPad App Leitstand + iPhone App Fahrgastinformation

Linie	L/F	Standort	Ziel	Abweichung	Fahrzeug Nr.	Wendezeit	Telezeit	
<input checked="" type="checkbox"/>	Linie 33	Bushof	Vaals	+1,5	415	+0,5	16:27:23	
<input checked="" type="checkbox"/>	Linie 17	Juttastrasse	Marienhospital	+2,0	300	+0,5	16:27:23	
<input checked="" type="checkbox"/>	Linie 11	Linie 20	Juttastrasse	Marienhospital	+1,0	301	+0,5	16:27:23
<input checked="" type="checkbox"/>	Linie 51	Linie 40	Juttastrasse	Marienhospital	+0,0	303	+0,5	16:27:23
<input checked="" type="checkbox"/>	Linie 30	Linie 30	Juttastrasse	Marienhospital	+0,0	308	+0,5	16:27:23
<input checked="" type="checkbox"/>	Linie 31	Linie 30	Juttastrasse	Marienhospital	+0,0	309	+0,5	16:27:23
<input checked="" type="checkbox"/>	Linie 20	Linie 31	Juttastrasse	Marienhospital	-2,0	400	+0,5	16:27:23
<input checked="" type="checkbox"/>	Linie 40	Linie 30	Juttastrasse	Marienhospital	-1,0	410	+0,5	16:27:23
	Linie 30	Juttastrasse	Marienhospital	+0,0	417	+0,5	16:27:23	
	Linie 30	Juttastrasse	Marienhospital	+0,0	299	+0,5	16:27:23	
	Linie 30	Juttastrasse	Marienhospital	+0,0	188	+0,5	16:27:23	
	Linie 30	Juttastrasse	Marienhospital	+0,0	117	+0,5	16:27:23	

Linienauswahl: 33, 17, 11, 51, 30, 31, 20, 40 | Fahrzeugauswahl: 03. Apr 2011 16:27:52

Linie 30	Juttastrasse	Marienhospital	+0,0	417	+0,5	16:26:24
Linie 30	Juttastrasse	Marienhospital	+0,0	299	+0,5	16:26:24
Linie 30	Juttastrasse	Marienhospital	+0,0	188	+0,5	16:26:24
Linie 30	Juttastrasse	Marienhospital	+0,0	117	+0,5	16:26:24



Success Story I (b)

- > kurze Beschreibung der Funktionalität (13 Seiten) und der Schnittstelle (11 Seiten)
- > Storyboard für das User-Interface (5 Seiten)
- > Aufteilen der Funktionalität
- > iterative Implementierung
- > iterative Verfeinerung und Verbesserung
- > verzahnte Spezifikation (5 Versionen) und Implementierung

Success Story II

- > Transformationssoftware: Format X nach Y
- > Problem: völlig flache Struktur
- > Ansatz: Testfälle – immer komplexer werdende Eingabedateien, die schlussendlich alle Features von Format X abdecken
- > Resultat: sofortige, rasche Umsetzung des Konzepts und Kompletversion nach 8 Wochen

Success Story III

- > dahinplätscherndes Softwareprojekt ohne Fortschritt
- > Einführung von SCRUM in einer angepassten Form (Team 1 Senior + 4 Juniors)
- > Projekt wurde zielgerichtet und mit zunehmender Geschwindigkeit erfolgreich zu Ende gebracht

Probleme inkrementeller Entwicklung (a)

- > 1. Iteration:
lies Informationen von einer Webadresse, und
stelle sie geeignet dar
- 1 Webadresse hardcoded
- 2 hole Informationen
- 3 Information darstellen

Probleme inkrementeller Entwicklung (b)

- > 2. Iteration:
Webadresse konfigurierbar
- > Problem: erster Entwurf statisch, keine
Ereignisse
- > Neuer Entwurf ist ereignisgetrieben (update
Webadresse)

Agile Präsentationen

- > Das Problem ist nicht eine Präsentation agil zu halten...
- > aber wenig Werkzeuge um sie agil zu entwickeln...

Kapitel III

(Kleiner) Leitfaden und Zusammenfassung

Anwendbarkeit (Software-Team)

- > Für erfahrene Software-Teams ein Gewinn
- > Disziplin ist wichtig – Regeln einhalten, Termine einhalten
- > Auf Dokumentation achten
- > Änderungen nachvollziehbar dokumentieren
- > verteilte Versionskontrolle
- > Qualitätssicherung nicht vernachlässigen
- > Testgetriebene Entwicklung ersetzt keine Systemtests
- > Change Management

Anwendbarkeit (Kunde)

- > für kleine und mittlere Projekte
- > kein Festpreis, sondern Zwischenreleases vereinbaren und bezahlen
- > mindestens ein Mitarbeiter muss sich verantwortlich fühlen und Zeit investieren
- > alles braucht länger als man denkt und kostet mehr als man erwartet
- > wenig Vorteile für klar umrissene Routineaufgaben

Missverständnisse

- > agil bedeutet undiszipliniert.
- > agil bedeutet auf Planung zu verzichten.
- > agil bedeutet keine Dokumentation.
- > agil ist eine festgelegte Methode.

Vorteile

- > Schnellere Entwicklung
- > Geringere Kosten
- > zufriedenerere Entwickler
- > zufriedenerere Kunden

Probleme

- > große Projekte
- > passive Kunden
- > Festpreis-Verträge
- > Test-Verständnis
- > unerfahrene Entwickler

Let me repeat...

- > Agile Entwicklung ist kein festgelegter Prozess, sondern eine Arbeitsweise und -philosophie
- > Für den Erfolg muss man sie an das Problem anpassen
- > Das nehmen was passt – das andere weglassen
- > Grosse Projekte in agile und nicht-agile Anteile splitten (zB. sicherheitskritischer Kern nicht-agil, Benutzerführung agil)

Fazit

- > Agile Entwicklung kann dem Entwickler und seinem Kunden helfen
- > Beide Seiten müssen agile Prozesse verstehen und flexible umsetzen
- > Offene, zeitnahe Kommunikation zwischen Kunde und Entwickler ist wesentlich
- > Risiken kontinuierlich kontrollieren
- > Teamansatz mit hohem Einsatz und Eigenverantwortung

*Neue Fehler warten
Steine liegen auf dem Weg
Ich leg sie rüber in den Garten
(Dirk von Lowtzow/Tocotronix)*