

Wasserfall, «Death March», Scrum und agile Methoden



30.August 2011 Embedded Computing Conference 2011
Urs Böhm



Übersicht

- Entwicklungsprozess
- Warum Projektmanagement?
- Gängige SW Entwicklungsprozesse
 - Wasserfall
 - V-Modell
 - „Death March“
 - Iteratives / Imperatives Modell
 - Scrum
- Vergleichstabelle / Übersicht /
- Wie wähle ich das richtige SW Entwicklungsmodell?
- Diskussion



Was ist Projektmanagement?

- Erreichen eines Ziels
 - Beschränkte Ressourcen
 - On Time
 - On Budget

- Projektmanagement ist nicht Ressourcenmanagement





Toolunterstützung



Prozesse





Projektmanagement

- Software wird heute oft in einem oder mehreren Projekt(en) hergestellt
- Projektmanagement ist sinnvoll – gerade auch für Software
- Software ist jedoch – irgendwie – anders
- Vervielfältigungskosten sind vernachlässigbar
- Kein Verschleiss aber "Software Alterung"
- Lizenz-Chaos (Rechtliche Situation ist bei Software ein Drama)
- Konfigurierbar
- Leicht und schnell Änderbar
- Software enthält Fehler. Das ist akzeptiert und Fehlerfreiheit ist nicht zu erreichen



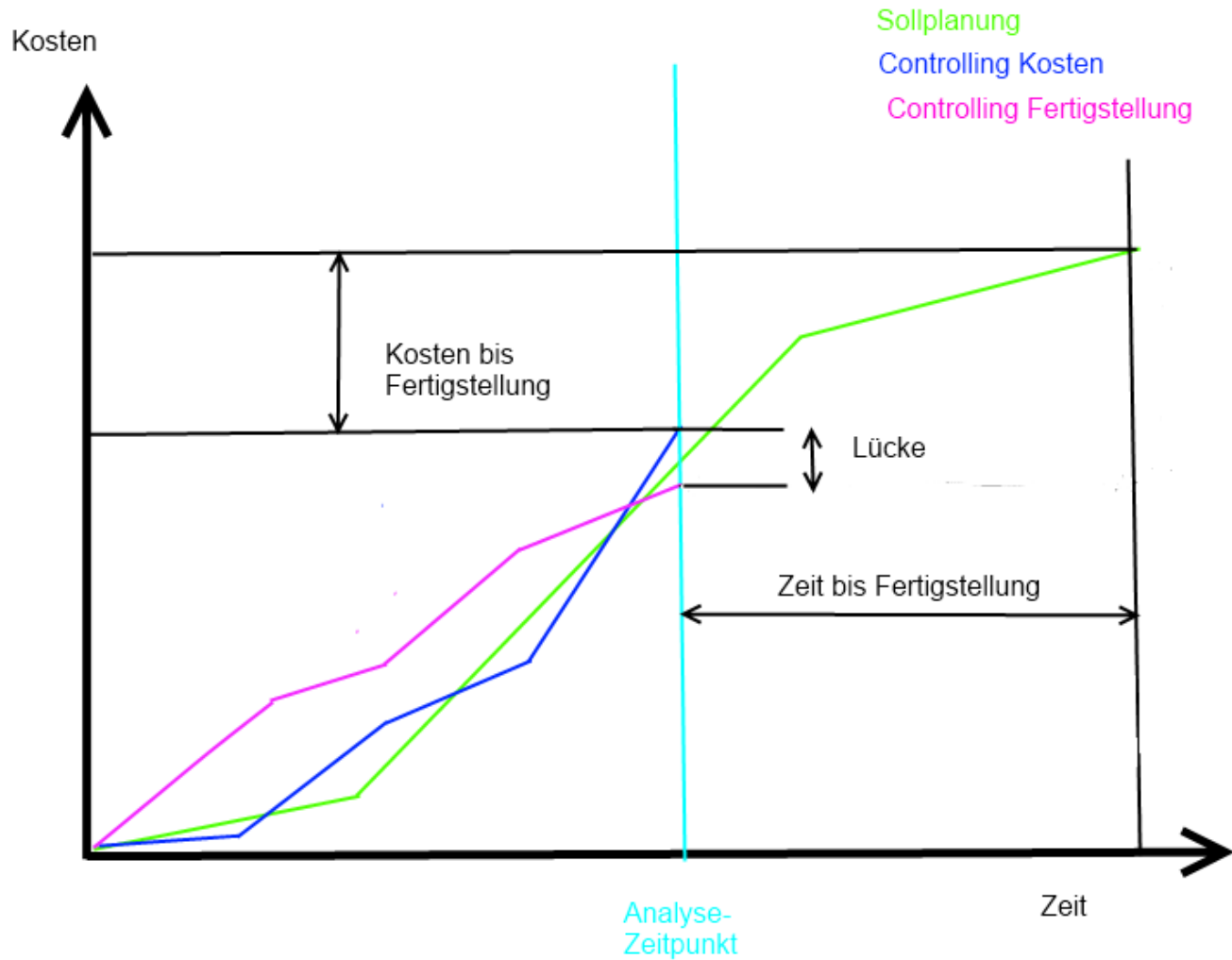
Prozesse erleichtern das Steuern und Regeln!

- Positionsbestimmung!
 - Wo bin ich fachlich im Projekt? Was ist getan und was muss noch getan werden?
 - Welchen Aufwand hatte das Projekt bisher?
 - Welchen Aufwand erwarte ich noch?
- Weiteres Vorgehen bestimmen:
 - Ist mein Projekt auf Kurs?
 - Muss regelnd / korrigierend eingegriffen werden?
 - Kann es wie geplant weitergehen?
 - Welche Risiken liegen im Projekt vor mir?

Ziemlich viele ?



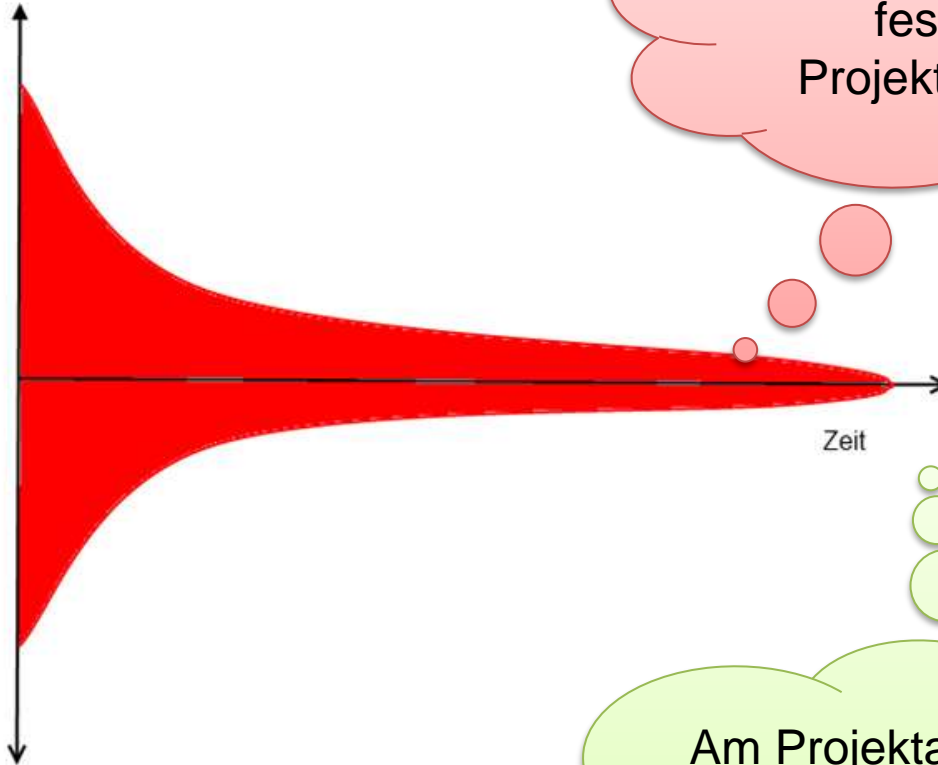
Projekt Überwachung





Schätzungen verfeinern sich

Fehler der Schätzung



Achtung: Bitte nicht an falschen Schätzungen festhalten und die Projektrealität ignorieren!

Am Projektanfang ist die Schätzung sehr ungenau.
Am Projektende weiss man's dann besser!



Software Entwicklungsprozess

- Wir haben in der Embedded SW-Entwicklung einige „Challenges“
- Grössere und leistungsfähigere Hardware als jemals zuvor
- Riesige Codebasis gerade auch für Embedded Systeme

- Teamgrössen und Zusammensetzung
- Qualität
- Testabdeckung
- Produktivität
- Effizienz und Effektivität

- Funktionale Sicherheit wird mehr und mehr auf Software ausgedehnt EN ISO 61508 und abgeleitete Normen



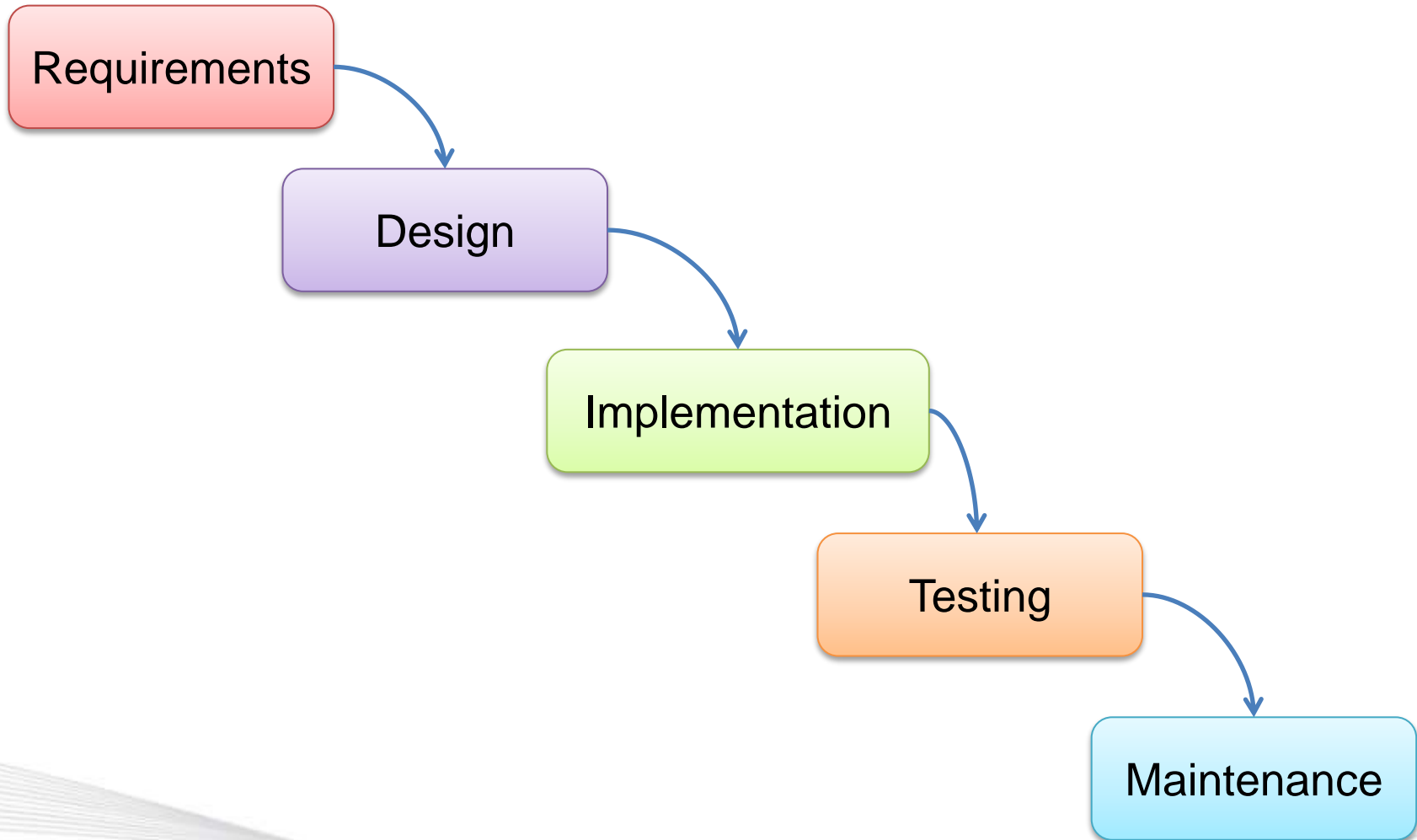
Beispiele von SW-Prozessen

- Waterfall
- „Death March“
- V-Modell

- Agile Methoden
 - Code and fix
 - Iteratives Vorgehen
 - Scrum



Waterfall





Und in der Original Darstellung

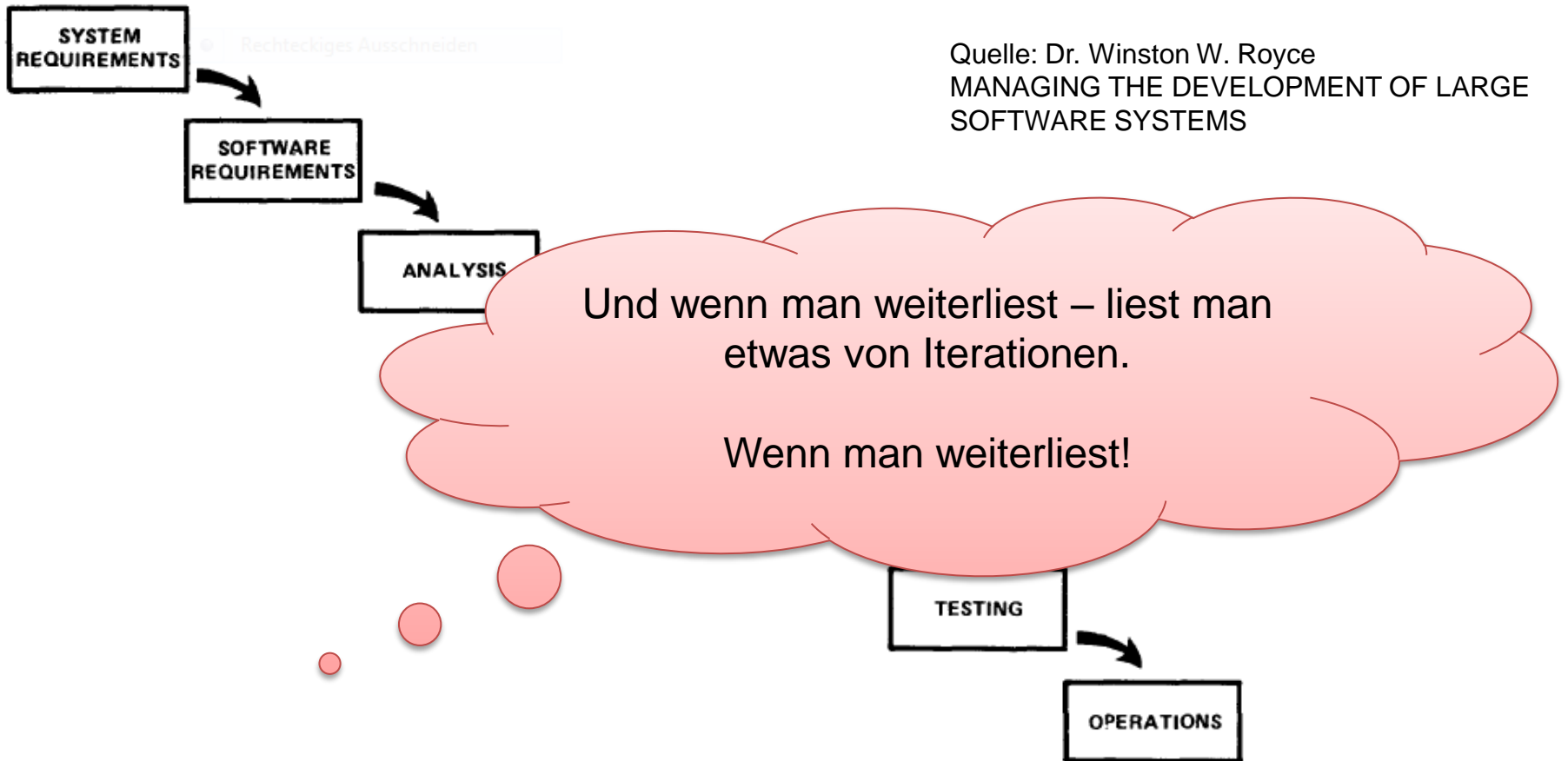


Figure 2. Implementation steps to develop a large computer program for delivery to a customer.



Big Design Up Front

Fehlervermeidung in frühen Phasen ist günstiger

Aufwändiger Umgang mit unbekanntem oder fehlenden Requirements

Starkes Requirement Engineering

Späte Sichtbarkeit des Ergebnisses

Klare Phasen zum Fortschrittstracking

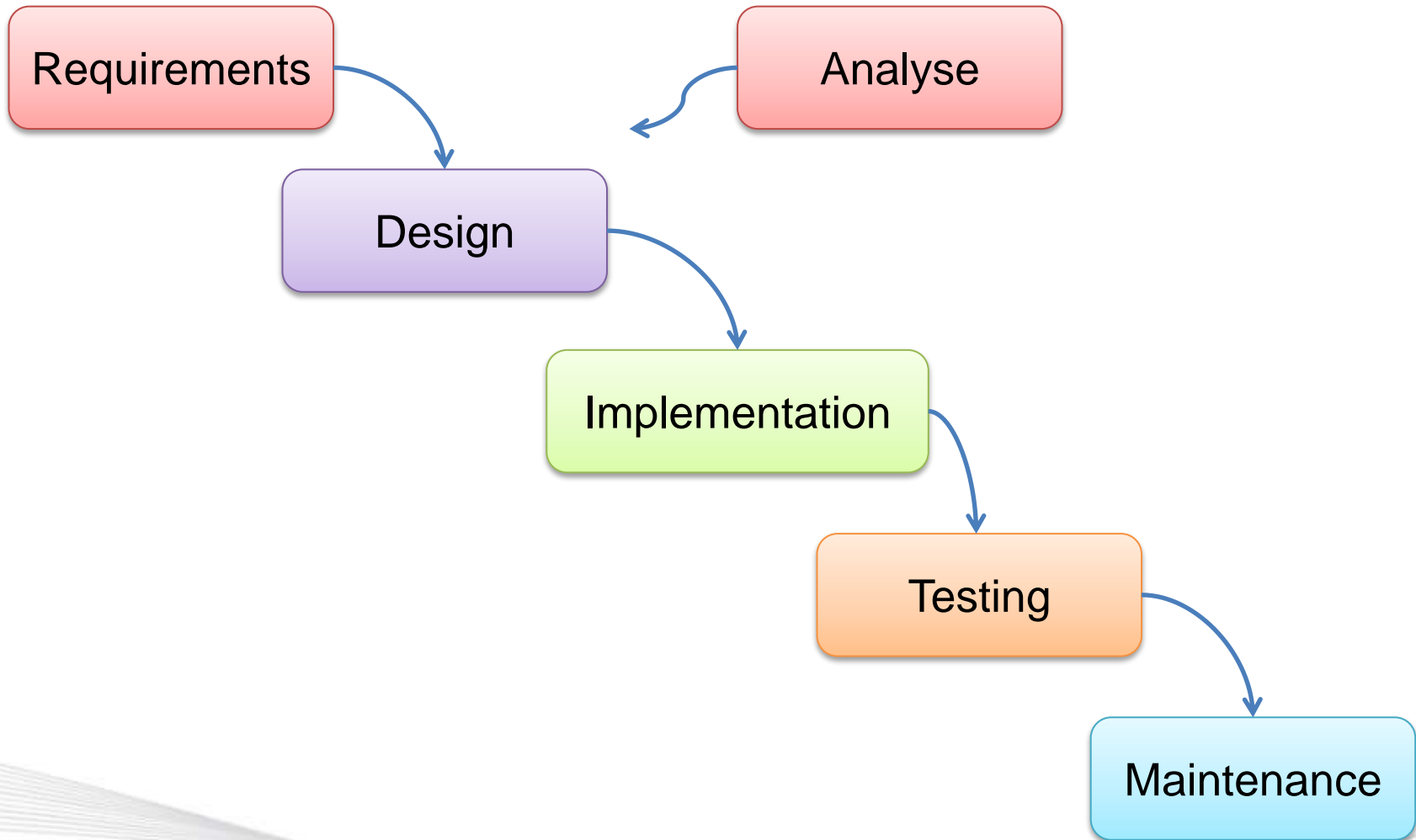
Keine Projektabbruchstrategie

Passende SW-Architektur und SW-Design

Analysis Paralysis

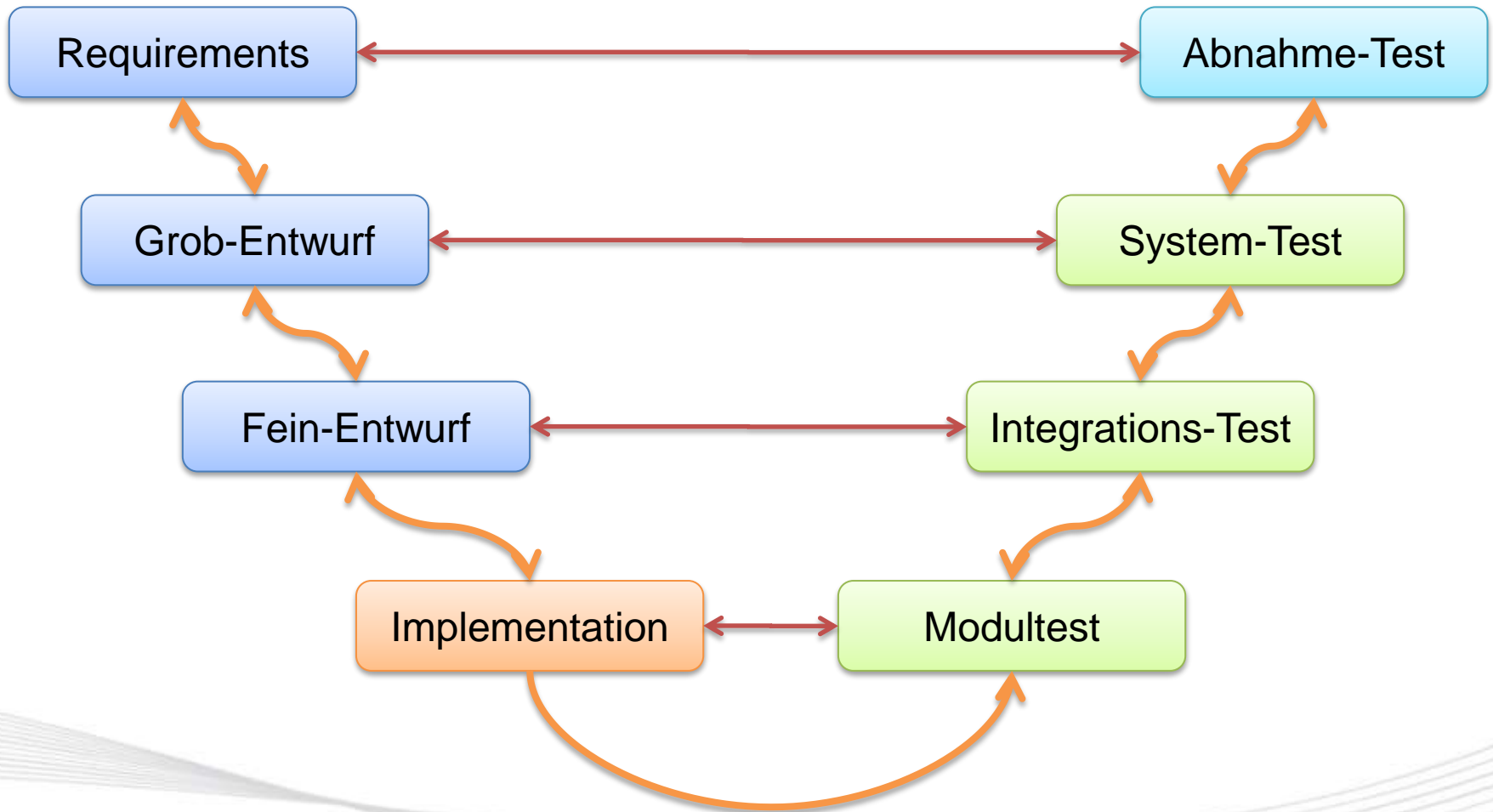


V-Modell basiert auf dem Wasserfallmodell





V Modell





„Death March“ Model in Comic

Licensed from www.dilbert.com

In der Präsentation wurde ein Comic gezeigt. Aus lizenzrechtlichen Gründen wird dieser Comic nur in der Live Präsentation auf der ECC 2011 gezeigt:

Quelle:

<http://search.dilbert.com/search?w=death+march>



„Death March“ in Project Management

Unrealistische Termin-Planung

Unrealistischer SW-Umfang

Ungenügendes Training

Das Projekt ist zum Scheitern verurteilt!

Aber die Beteiligten sind zum Erfolg verdammt!

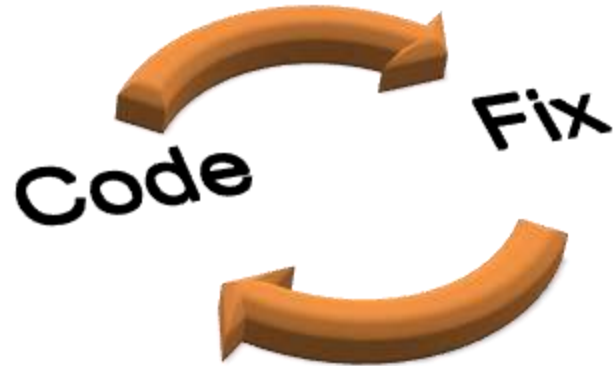


Nachteile von „Death March“-Projekten

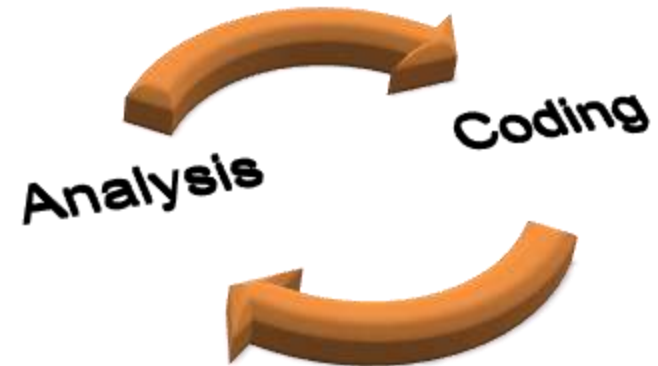
- Projektmanagement Methode der „Steuerung durch Verknappung einer Ressource.“
- Unerwünschte Begleitergebnisse:
 - Abstriche an der Qualität
 - Abstriche am Funktionsumfang
 - Geringere Testtiefe
 - Mehr SW-Bugs
 - Burnout / Fluktuation / Innere Kündigungen
- Auf einer solchen SW kann ohne massives Refactoring nicht weiter aufgebaut werden
- Grosse Gefahr des Scheiterns



Code and fix



Früher mal so:



- Ist kein agiles Vorgehensmodell
- Geeignet aber trotzdem wenn Nutzer und Programmierer dieselbe Person sind
- Kleine Projekte / Kleine Software



Das Problem?

Billiger

Termingerechter

Budgetierter

Besser



Und gleichzeitig offen für Veränderungen ohne dass die obigen Eigenschaften berührt werden.





Manifest für Agile Softwareentwicklung

Wir erschließen bessere Wege, Software zu entwickeln,
indem wir es selbst tun und anderen dabei helfen.
Durch diese Tätigkeit haben wir diese Werte zu schätzen gelernt:

Individuen und Interaktionen mehr als Prozesse und Werkzeuge

Funktionierende Software mehr als umfassende Dokumentation

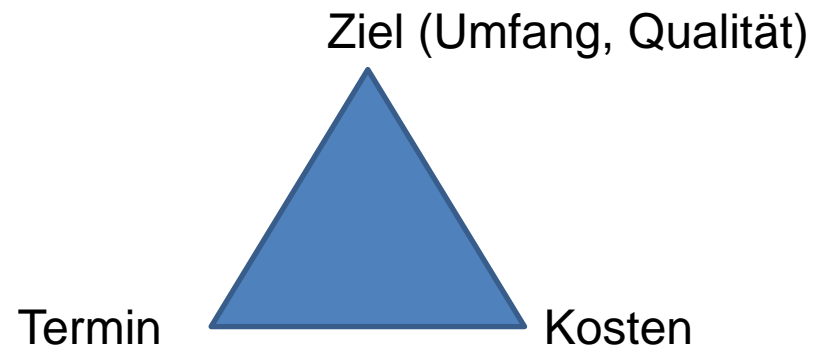
Zusammenarbeit mit dem Kunden mehr als Vertragsverhandlung

Reagieren auf Veränderung mehr als das Befolgen eines Plans

Das heißt, obwohl wir die Werte auf der rechten Seite wichtig finden,
schätzen wir die Werte auf der linken Seite höher ein.

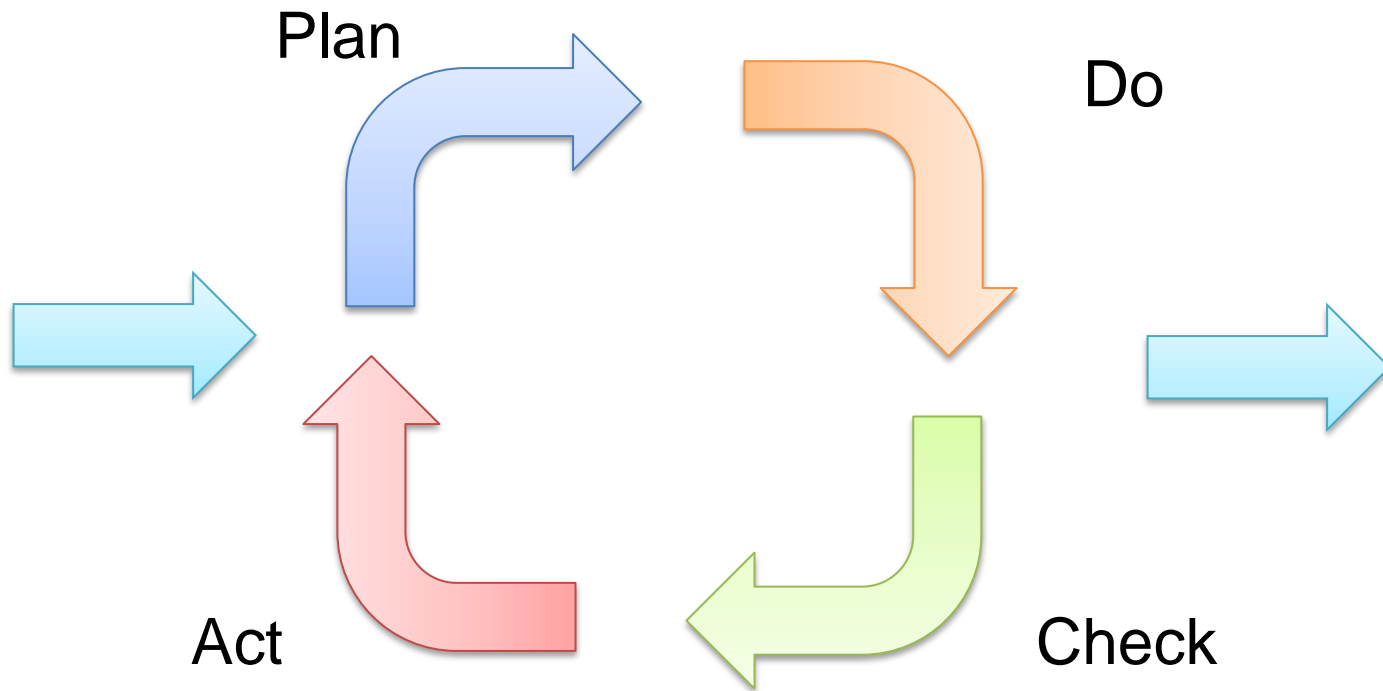


Magische Dreieck



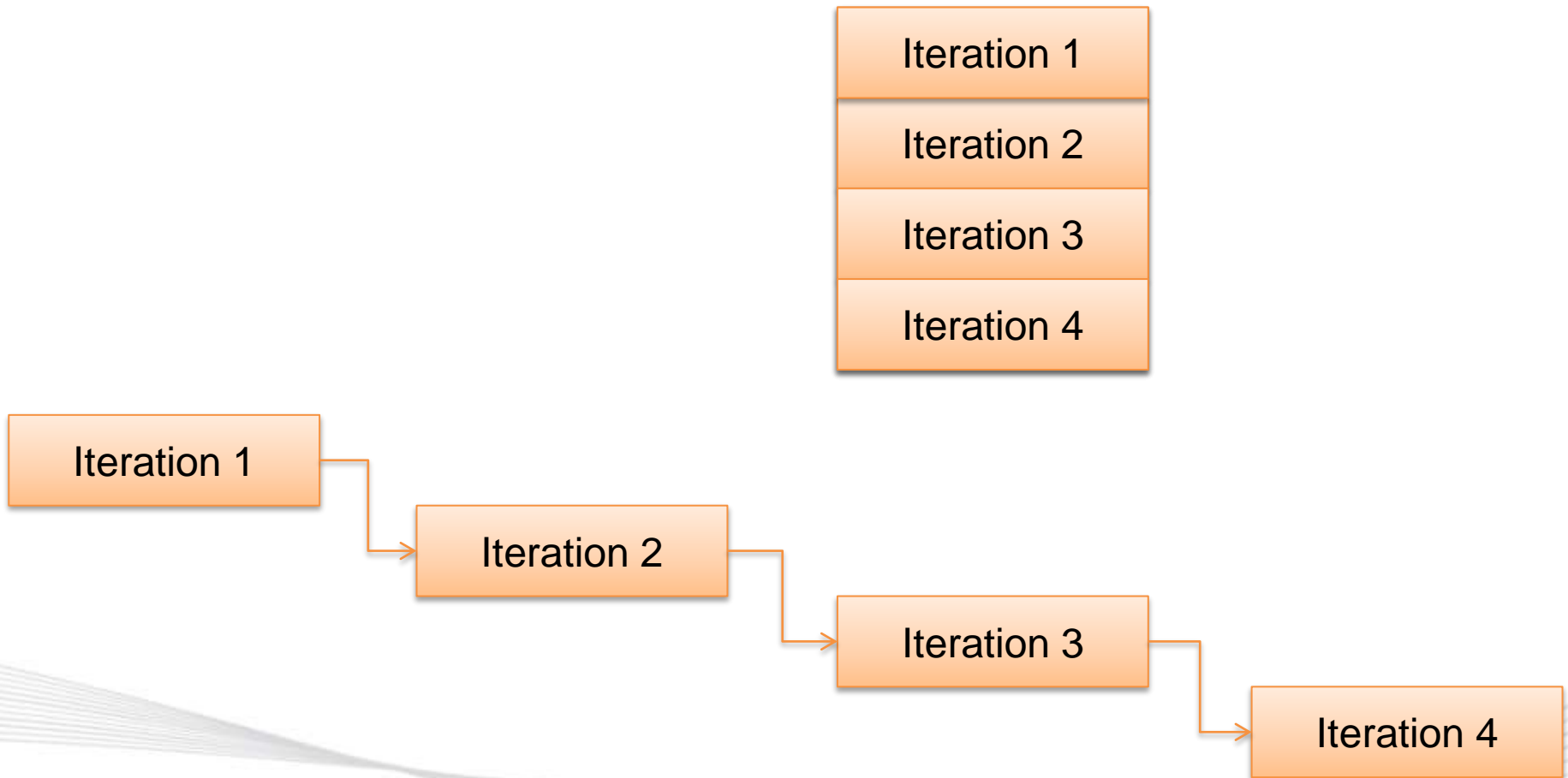


Inkrementelles Vorgehen





Iteratives Vorgehen „Divide Et Impera“





Scrum Schlagworte

- Scrum ist eine agile Projektmanagement Methode
- Besonders, aber nicht ausschliesslich für komplexe Software geeignet
- Setzt stark auf eigenverantwortliche Teams
- Bietet & verlangt grosse Transparenz
- Deklariert klare Rollen mit Rechten und Pflichten
- Greift tief in die Firmen und das Zusammenarbeiten miteinander ein – wenn komplett angewendet
- Guter Umgang mit ändernden Anforderungen
- Bei erfolgreichem Einsatz starke Fortschritte möglich.



Zertifizierungen im Scrum Prozess

Certified Scrum Master

Certified Scrum Product Owner

Certified Scrum Developer

Certified Scrum Professional

Certified Scrum Trainer

Certified Scrum Coach

Registered Education Provider



Kommunikation

Daily Scrum

Sprint Planning Meeting

Sprint Review Meeting

Sprint Retrospective

Besprechungen sind:

- Klar strukturiert
- Inhaltlich festgelegt
- Timeboxed
- Ergebnis





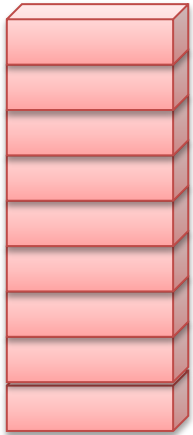
SCRUM Prozess

Burndown Chart

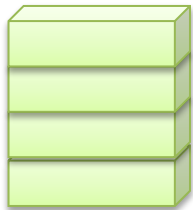
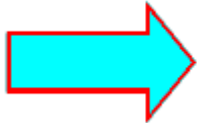
Sprint Review

Sprint Retrospektive

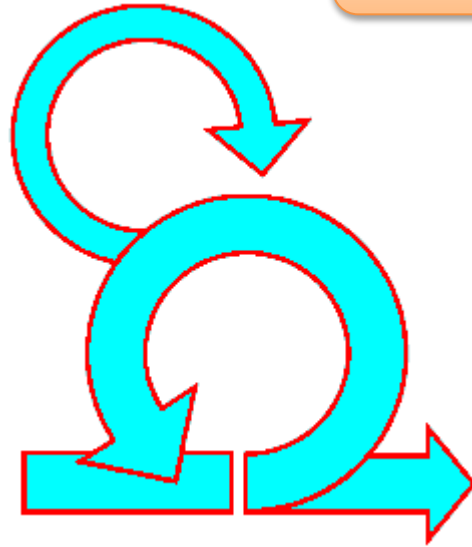
Burndown Chart



Produkt Backlog



Sprint Backlog



Potentiell Lieferbares Produkt-Inkrement



Scrum Rollen



Management



Kunden



User

Entwicklungs-
Team



Produkt Owner



Scrum Master



Verantwortlichkeiten

Produkt Owner:



- Auslieferungszeitpunkt
- Funktionalität
- Kosten

- Backlog Verantwortung
- Priorisierte User Stories

Entwicklungs-Team:



- Implementiert User Stories
- Garantiert die vereinbarte Qualität
- Entscheidet wie viel im Sprint entwickelt wird

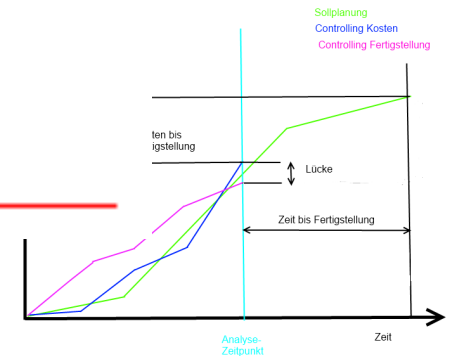
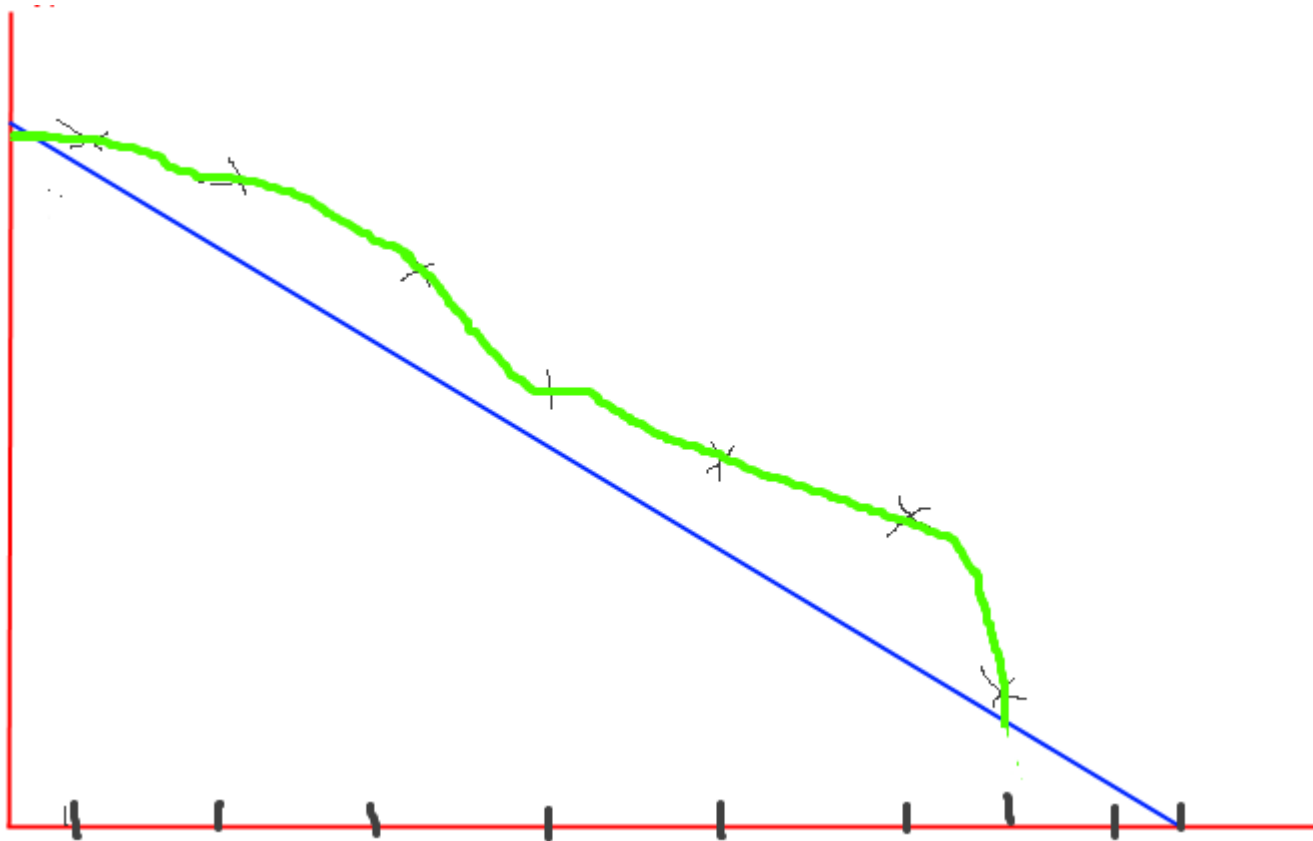
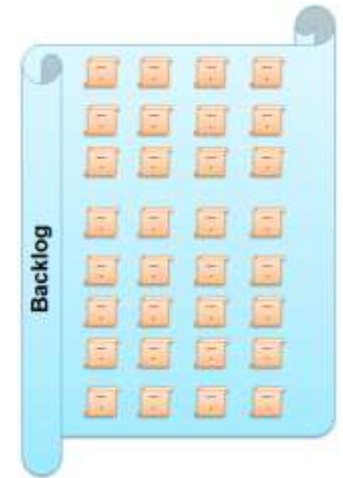
Scrum Master:



- Prozessverantwortung
- Eliminiert Störungen
- Impediments „Probleme“



Tracking des Projektfortschritts



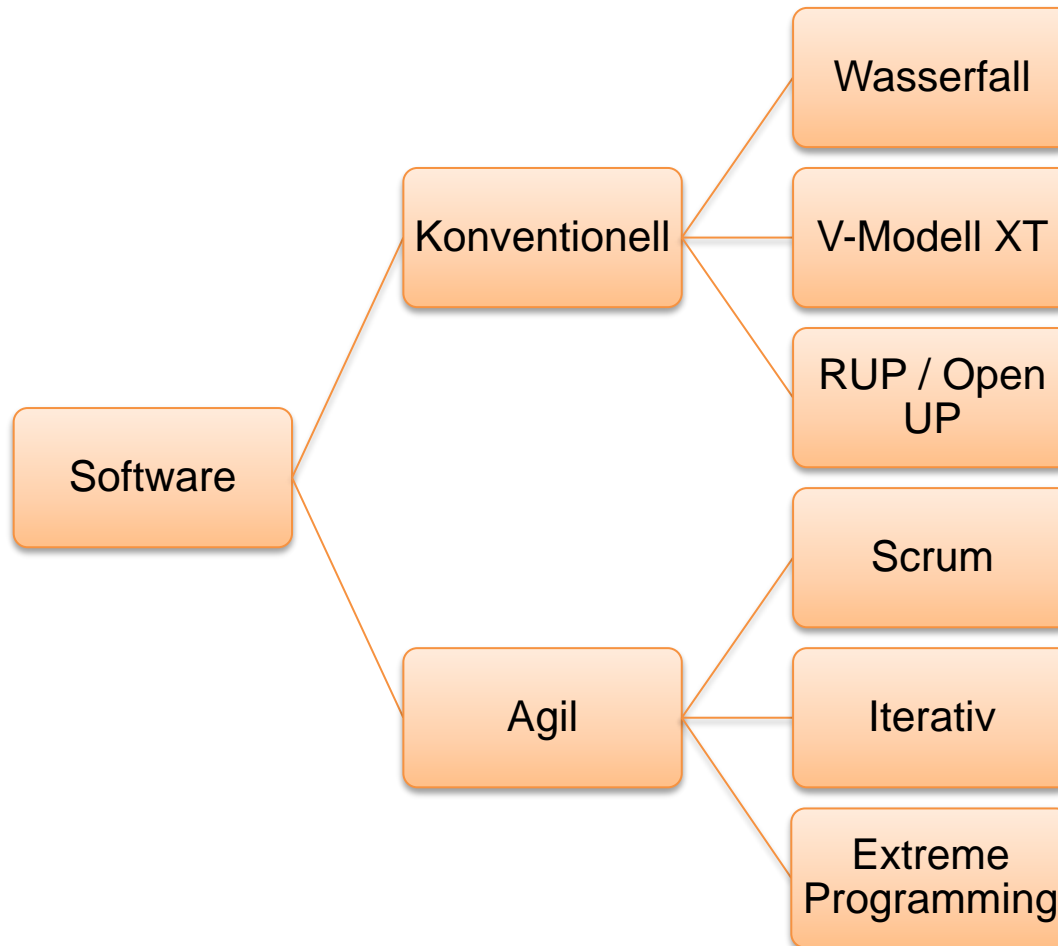


Vergleich

	Wasserfall	Death March	V-Modell	Code & Fix	Scrum
Flexibilität	--	--	-	++	++
Change	--	--	-	+	++
Planbarkeit	++	---	++	--	+
Abbruchmöglichkeiten	--	-	-	+	++
Quick and Dirty	--	++	--	++	--
Umgang mit formalen Anforderungen	++	-	++	--	-

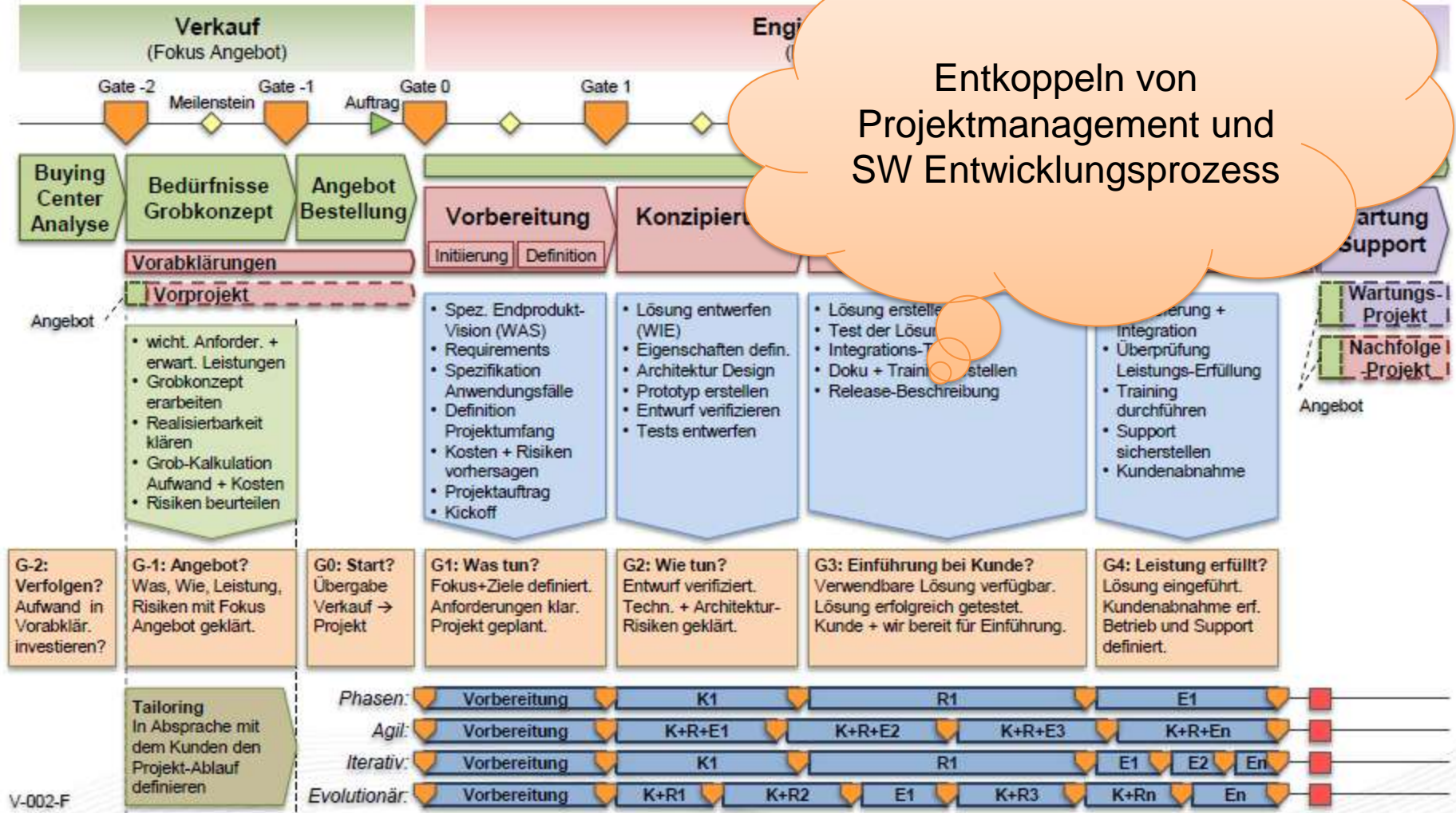


Entscheidungsgraph





NoEngPro



V-002-F



Links

- Ein paar interessante Links zum Thema
- Wikipedia & das passende Suchwort
 - http://de.wikipedia.org/wiki/Liste_von_Softwareentwicklungsprozessen
- Wasserfall
 - [Royce, 1970] "Managing the Development of Large Software Systems"
 - "The Rise And Fall Of Waterfall" (Youtube)
- V-Modell
 - <http://v-modell.iabg.de/>
- Scrum
 - „Scrum in under 10 minutes“ (Youtube)
 - <http://www.scrumalliance.org/>
 - <http://www.scrum.org/>
 - „State of Agile Development“ survey
- Hermes
 - http://www.hermes.admin.ch/welcome?set_language=de&cl=de



NOSERENGINEERING

WE KNOW HOW

Home | Über uns | Aktivitäten | Industrien | Tech Centers | Kunden | Academy | Newsroom | Showcase | Jobs & Karrieren | Kontakt



BERATUNG | LÖSUNGEN | PROJEKTE | BRAIN SOURCING

NOSER ENGINEERING AG
Talackerstrasse 99
CH-8404 Winterthur

+41 52 234 56 48 direct
+41 52 234 56 11 phone

urs.boehm@noser.com
www.noser.com

WE KNOW HOW