

Talk im Schloss

Zusammenbringen was zusammen gehört –

Der richtige Softwareentwicklungsprozess für
erfolgreiches Usability Engineering

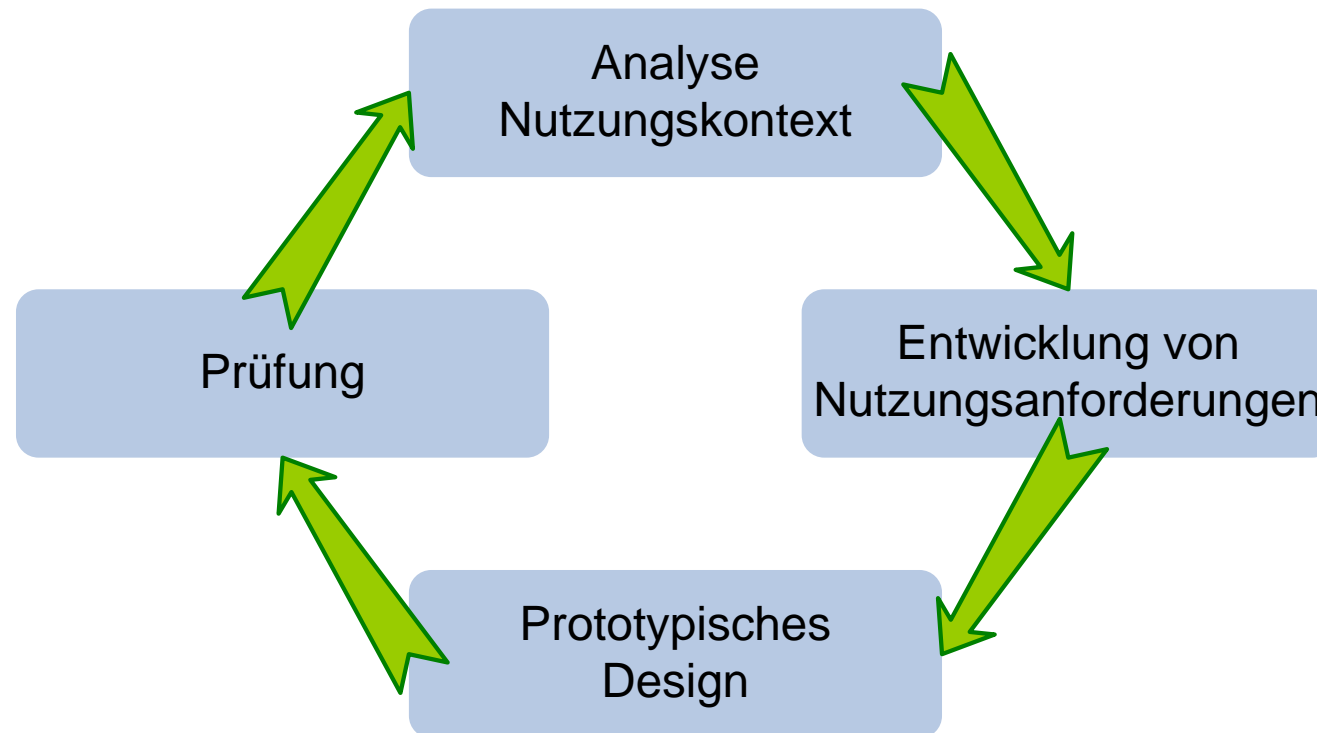
10.12.2007

F.Riemenschneider
+49 177 291 68 32
falko.riemenschneider@itemis.de

Inhalt

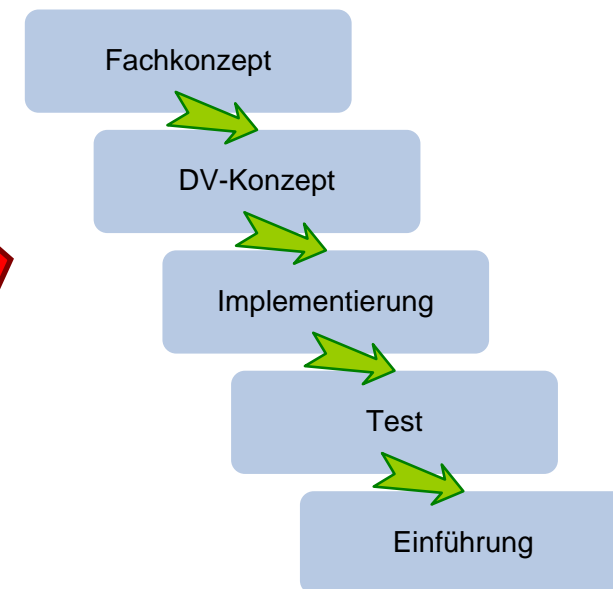
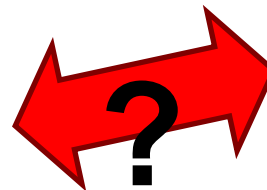
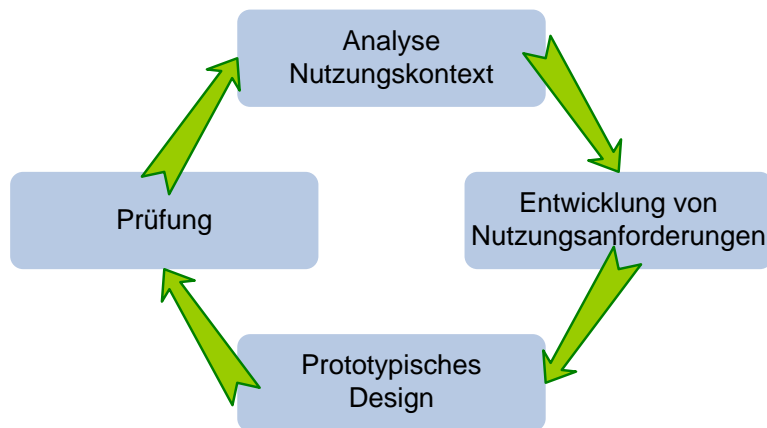
- Problemstellung
- Beispiel für ungünstigen Project Lifecycle
- Besserer Project Lifecycle

Der Usability Engineering Prozess



Geht das irgendwie zusammen?

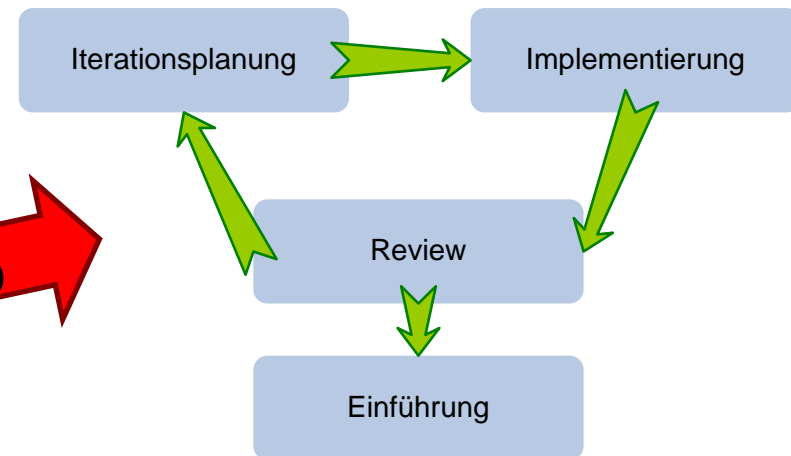
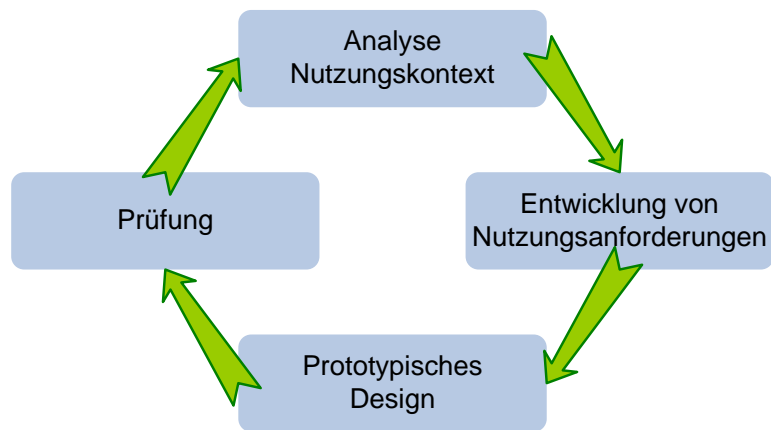
Usability Engineering Prozess



Softwareentwicklung nach "Wasserfall"-Modell

Oder eher das?

Usability Engineering Prozess



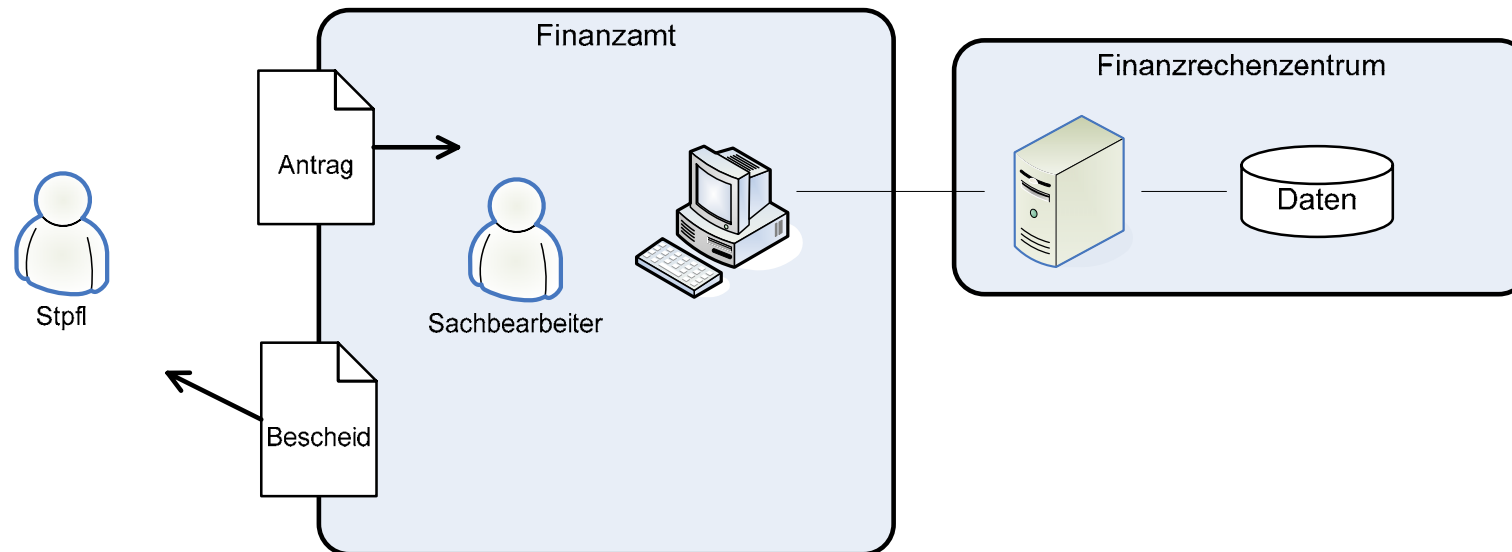
Agile Softwareentwicklung

Software Project Lifecycle

- Die richtige Frage lautet nicht:
"Gestalte ich das Projekt nach Wasserfall oder agil?"

- Die richtige Frage ist vielmehr:
"Wie sieht ein individueller Software Project Lifecycle aus, der zu meinen Rahmenbedingungen passt?"

Beispiel Projekt für Finanzverwaltung

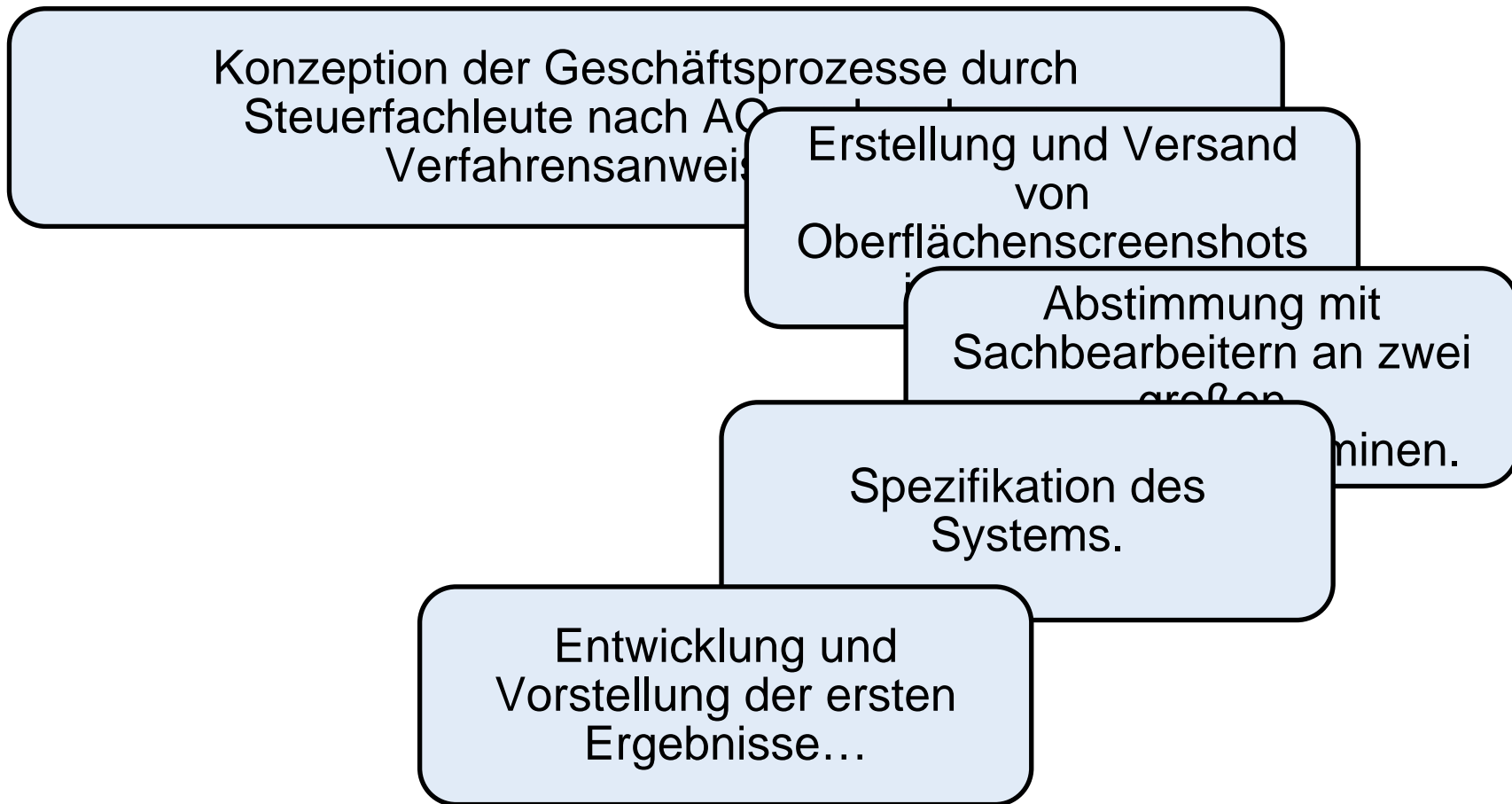


- Antrag auf Stundung oder Erlass erfassen
- Entscheidung bearbeiten
- Bescheid drucken

Beispiel Projekt für Finanzverwaltung

- Projektrahmen
 - Bestehende technische + fachliche(!) Frameworks
 - Umfang scheinbar festgelegt (aber nicht definiert)
 - Fixer Termin, fixes Budget
- Vorgehen nach V-Modell
(leicht angepasst, aber im Grunde ein Wasserfall)

Zeitlicher Ablauf



Schmerzliche Erkenntnisse

- Fachwissen war im Projekt nur scheinbar vorhanden.
- Bearbeiter/Auftraggeber waren bzgl. Fachlichkeit selbst unsicher.
- Lösungen in Frameworks z.T. nicht dem Problem angemessen.
- Oberflächen im Gebrauch ineffizient bis ineffektiv.

Konsequenzen

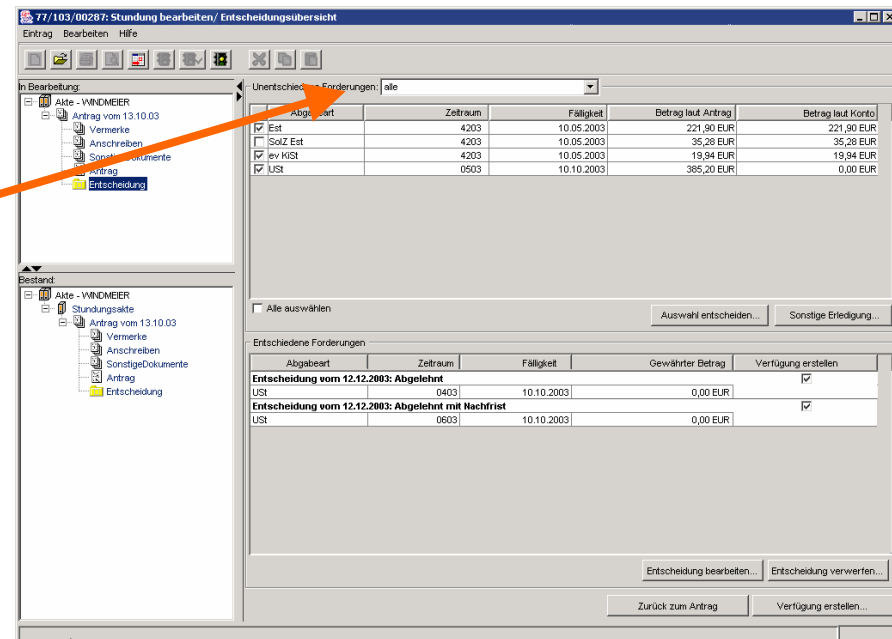
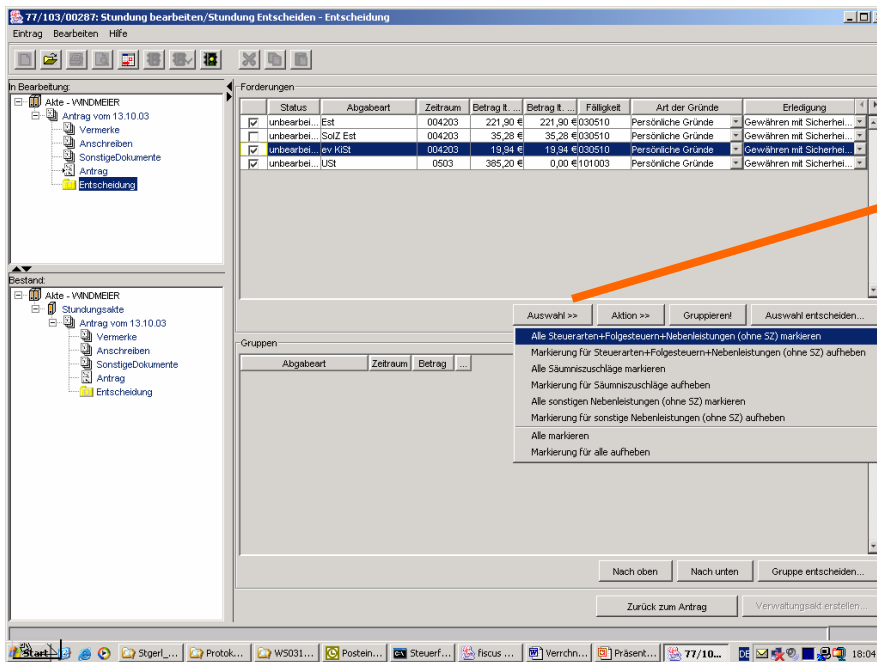
- Review der Oberfläche ergibt umfangreichen Änderungsbedarf.
 - Neukonzeption
 - Neue Realisierung der Oberflächen
- Budget gesprengt.
- Beim Versuch, den Termin zu halten, wurden viele Defekte eingebaut.
- Notwendige Stabilisierung führt zu deutlicher Terminverschiebung.

- Die Neukonzeption hat sich trotzdem gelohnt. Heute ist die Software im Land Niedersachsen im Einsatz.

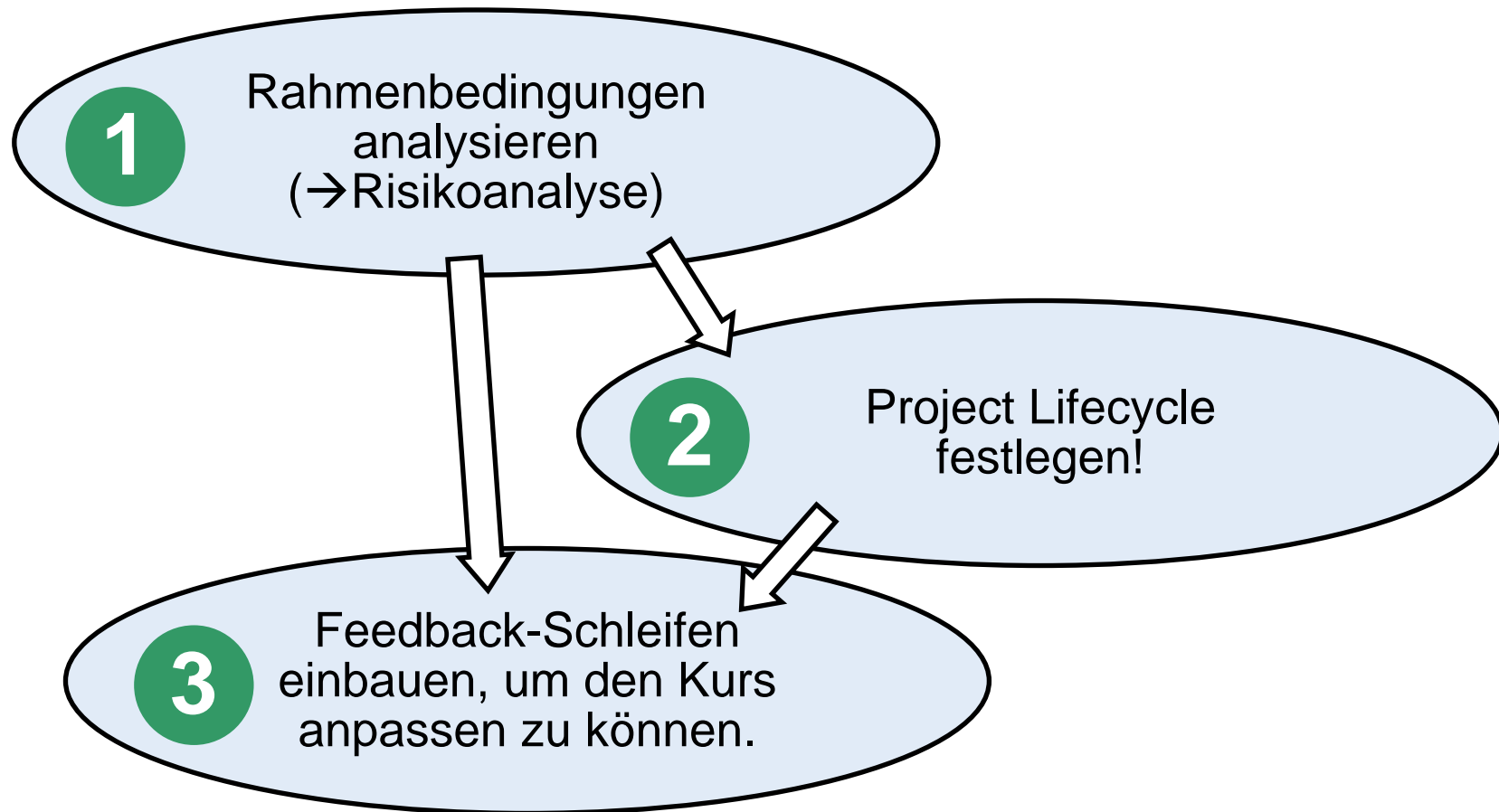
Beispiel für Effizienzgewinn

vorher

nachher



Lehren



Fragen zu Rahmenbedingungen

Frage	Antwort
Wie kritisch ist die Gebrauchstauglichkeit für den Projekterfolg?	
Besitzt das Projekt echtes, aktuelles Fachwissen von Benutzern?	
Was ist tatsächlich fix? Umfang, Termin, Kosten?	
Gibt es mögliche Lieferstufen?	
Wieviel Agilität lässt das Umfeld zu (Projektmitarbeiter, Stakeholder, Betrieb, QM, Architektur, ...)?	
Wieviel Übung besitzt das Projektteam in diesem Umfeld?	

Sinnvolle Feedback-Schleifen

- Benutzungstests!
- Pflege der Beziehung zu Stakeholdern (insbesondere Benutzergruppen).
- Change-Prozess etablieren.
- Ist-Kosten für Herstellung sammeln.
- Qualität (#Bugs) früh messen.

Rahmenbedingungen Beispiel Projekt

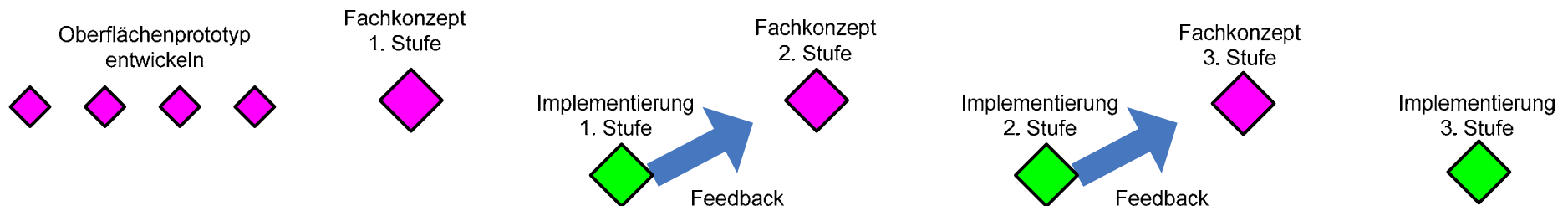
Frage	Antwort
Wie kritisch ist die Gebrauchstauglichkeit für den Projekterfolg?	Sehr kritisch.
Besitzt das Projekt echtes, aktuelles Fachwissen von Benutzern?	Nein, nicht ausreichend.
Was ist tatsächlich fix? Umfang, Termin, Kosten?	Umfang angeblich, Termin politisch gewünscht, Budget fix.
Gibt es mögliche Lieferstufen?	Nicht erwünscht.
Wieviel Agilität lässt das Umfeld zu (Projektmitarbeiter, Stakeholder, Betrieb, QM, Architektur, ...)?	Kaum Spielraum.
Wieviel Übung besitzt das Projektteam in diesem Umfeld?	Keine.

Passende Heuristiken (1/2)

- Zyklische Entwicklung eines Oberflächenprototypen zusammen mit Benutzern.
- Auf Build-to-Budget vorbereiten:
 - Umfang ist Manövriermasse (→Req.+Change Management)
 - Sinnvolle Lieferstufen stichwortartig formulieren.
 - Konservativ die Featuremengen festlegen.

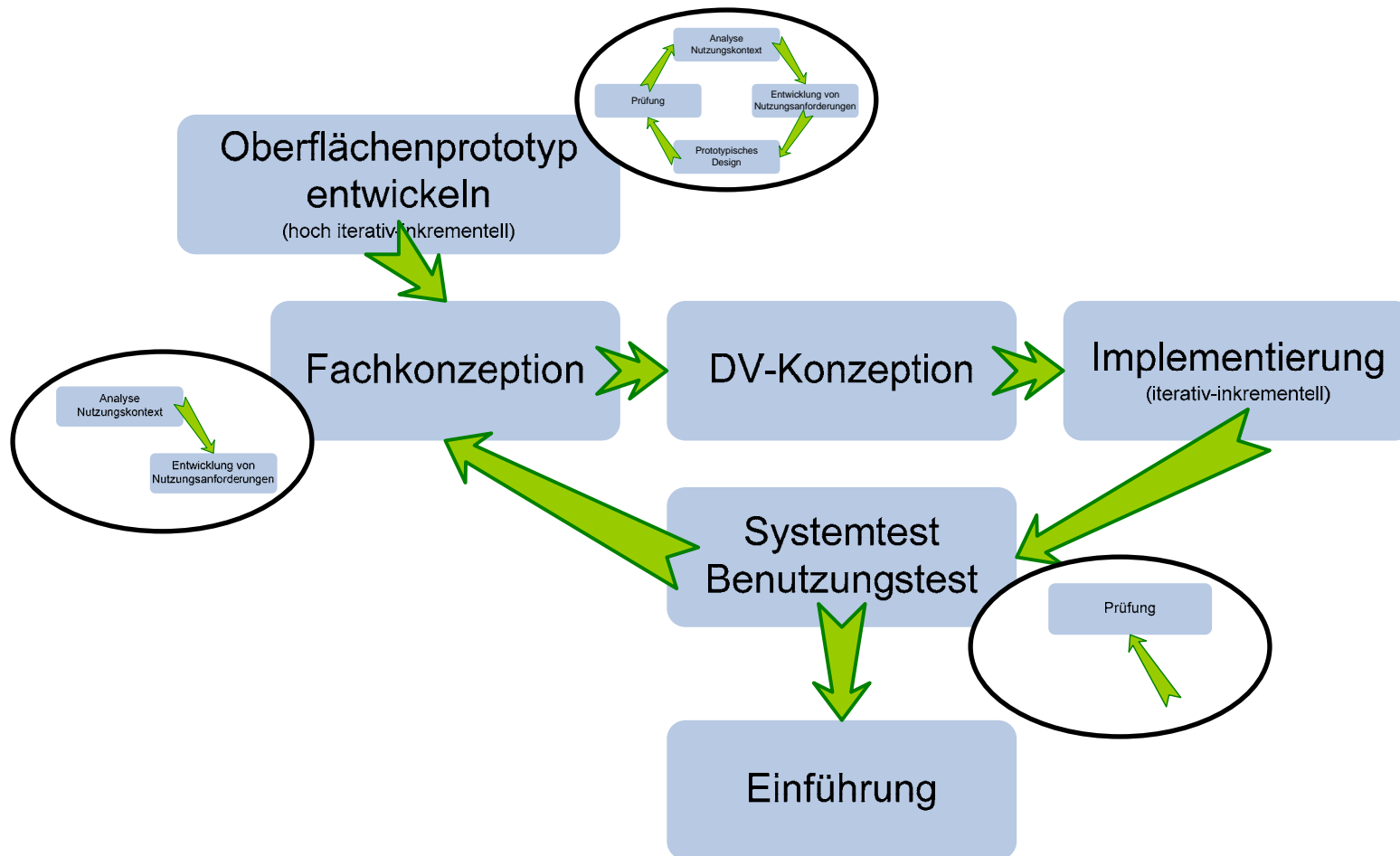
Passende Heuristiken (2/2)

- Größere Implementierungszyklen (ca. 3-4 Monate)
 - Grob planen. Mit Budget-Tracking steuern.
 - Stabilisierung gegen Ende mittels Systemtest.
 - Benutzungstests.



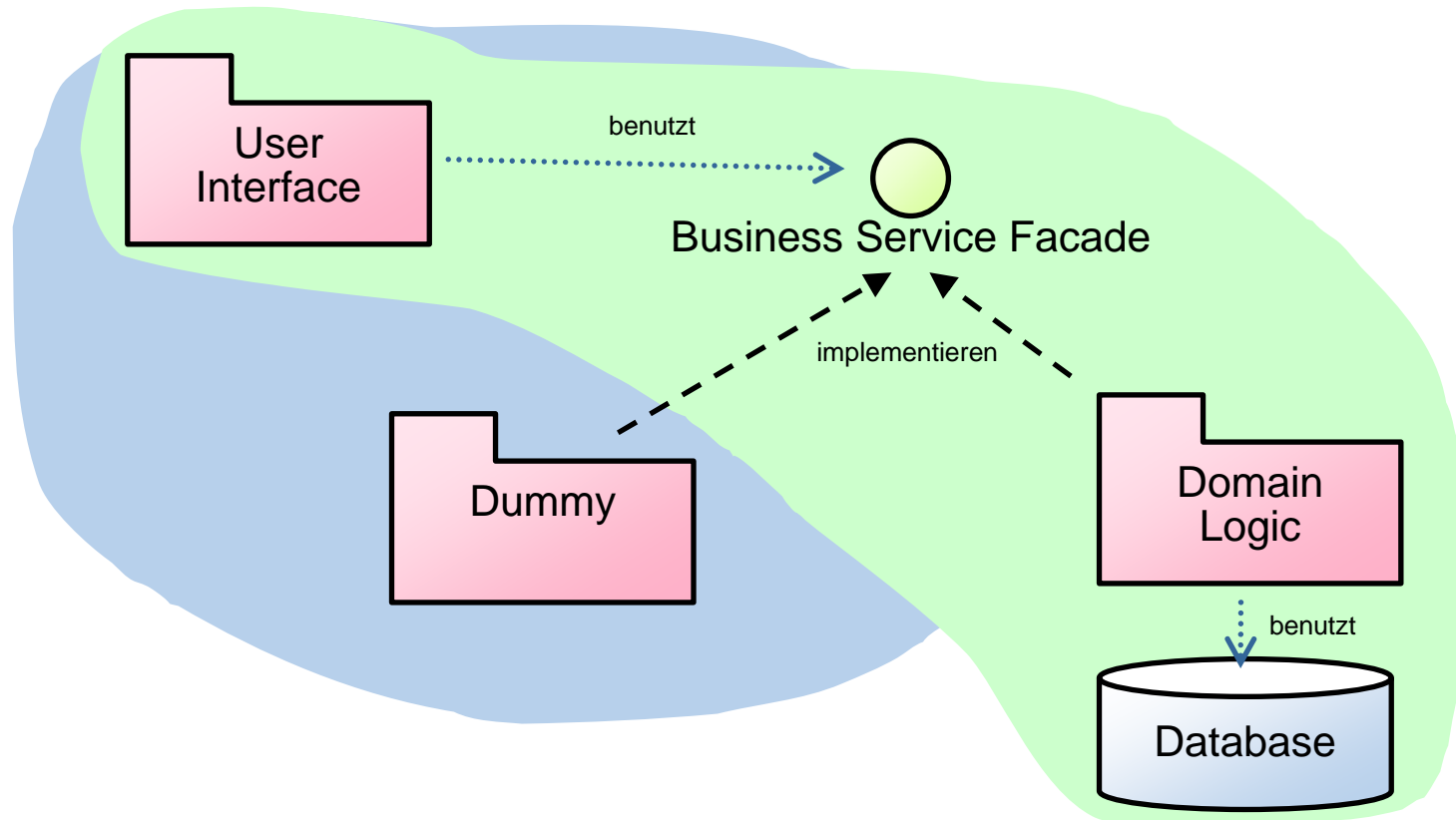
- Interne Meilensteine alle vier Wochen setzen. In diesen Abständen fein planen sowie Reviews von Qualität und Schätzgüte durchführen.

Angemessener Project Lifecycle



Software Architektur

- Entwurfsprinzip: Design-for-Change
 - "Was sich stark ändern kann, sollte gekapselt werden."



Zusammenfassung

- Der Software Project Lifecycle muss an individuellen Rahmenbedingungen ausgerichtet werden.
- Wenn Usability ein kritischer Erfolgsfaktor ist, muss der Software Prozess beweglicher werden, um Prototypentwicklung und Berücksichtigung der Ergebnisse von Benutzungstests zu erlauben.
- Bewegliche Prozesse benötigen eine Software Architektur, die Bewegungsfreiheit bietet. Das erfordert sehr hochwertiges Design!

Vielen Dank für die Aufmerksamkeit!

Fragen?



Kontakt

Falko Riemenschneider
+49 177 291 68 32
falko.riemenschneider@itemis.de

itemis AG – Niederlassung Bonn
Bundeskanzlerplatz 2-10
53113 Bonn