

Agile Softwareentwicklung und Usability

Wie mit Best Practices eine Brücke geschlagen werden kann

UIG-Frühjahrstagung 2015
15. März 2015, Mannheim

Dominik Magin, Hartmut Schmitt

GEFÖRDERT VOM



Bundesministerium
für Bildung
und Forschung

Agile Entwicklungsvorgehen

- Eigenschaften/Aspekte
 - Kundenorientierung, -feedback
 - Flexibilität für Änderungsanforderungen
 - schnelles Ausliefern nutzbarer Software
 - kurze Entwicklungszyklen
 - selbstbestimmte Entwicklungsteams
- Ziele
 - Kundenzufriedenheit
 - hochqualitative Produkte
 - effiziente Entwicklung
 - Mitarbeiterzufriedenheit

Agile Entwicklungsvorgehen

- Vorgaben für Projektmanagement und Vorgehen (Sprints, Backlog ...)
- Vorgaben für Implementierungstätigkeiten (Pair Programming, Test-Driven Development ...)

aber

- Qualitätsbetrachtung mit Schwerpunkt auf Funktionalität
- kaum Unterstützung für die Erreichung (nichtfunktionaler) Produktqualität
 - Gebrauchstauglichkeit
 - Zufriedenheit
 - Performanz
 - ...

Erfolgreiche agile Projekte

- förderliche Projekteigenschaften
 - erfahrene Teams
 - kleine/wenige Teams
 - übersichtliche Systemgröße
 - überschaubarer Lebenszyklus
 - direkte Einbindung des Kunden
 - laufendes System wird kontinuierlich ausgeliefert, diskutiert und verbessert
- Erreichung von Produktqualität durch Mitarbeitererfahrung und Fähigkeiten

Einsatz oft in „suboptimalen“ Projekten

- frühe Qualitätsmängel
- Behebung durch „versteckte“ Nacharbeiten
- Reduktion der Entwicklungsgeschwindigkeit
- hohe Wartungskosten
- negative Auswirkungen auf Produktivität, Produktqualität und Kundenbeziehungen

Motivation für Forschungsprojekt

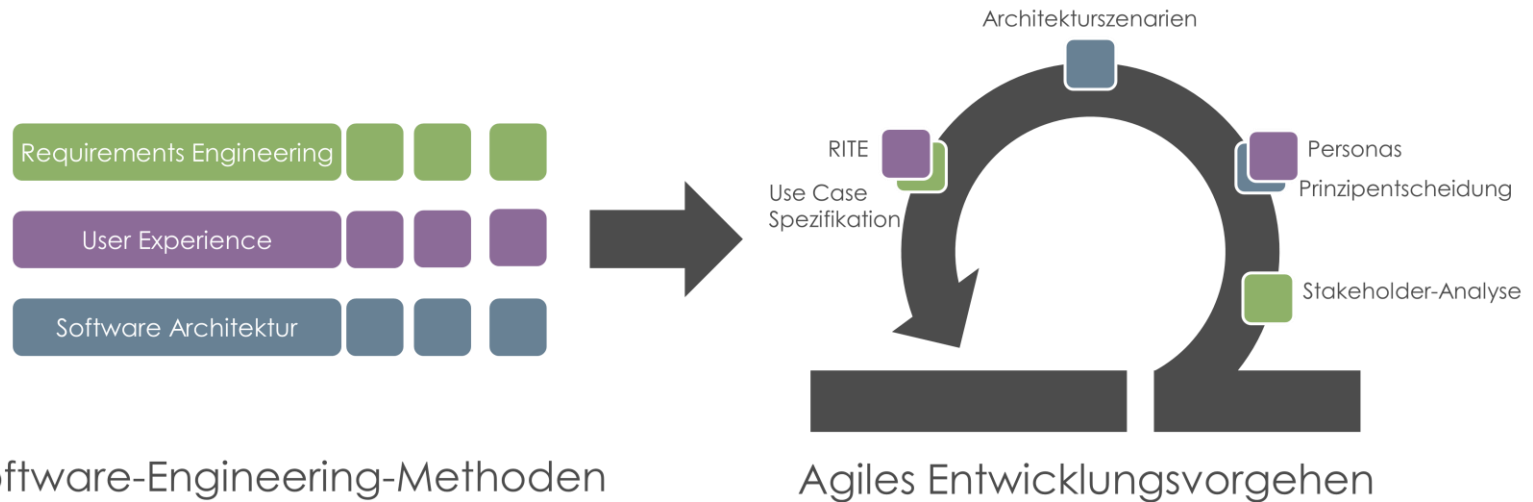


Vorhandenen Erfahrungsschatz nutzen

- Software-Engineering-Methoden
 - zur Erreichung von Produktqualität
 - aus verschiedenen Bereichen des Software Engineerings
 - entwickelt vor dem Hintergrund konventioneller Vorgehen (V-Modell, Spiral-Modell, Unified Process ...)
 - Anpassungsbedarf bei Integration in agile Entwicklungsvorgehen

Projektidee

„... etablierte Aktivitäten des Softwareengineerings in Best Practices zu zerlegen und Entwicklern für den selbstbestimmten Einsatz in agilen Projekten zur Verfügung zu stellen.“



Verbundvorhaben PQ4Agile

- Partner: HK Business Solutions, CAS Software, YellowMap, Fraunhofer IESE
- gefördert vom Bundesministerium für Bildung und Forschung
- Programm: KMU-innovativ, Softwaresysteme und Wissenstechnologien
- 24 Monate Laufzeit



Ziel

- Kompendium von SE-Best-Practices zur Erreichung von Produktqualität
 - effizient einsetzbar
 - kompatibel mit verschiedenen Vorgehensvarianten
 - Fokus: Entwickler, Generalist
 - selbstbestimmte Auswahl und Einsatz
 - speziell für KMU
 - konstruktive Aktivitäten

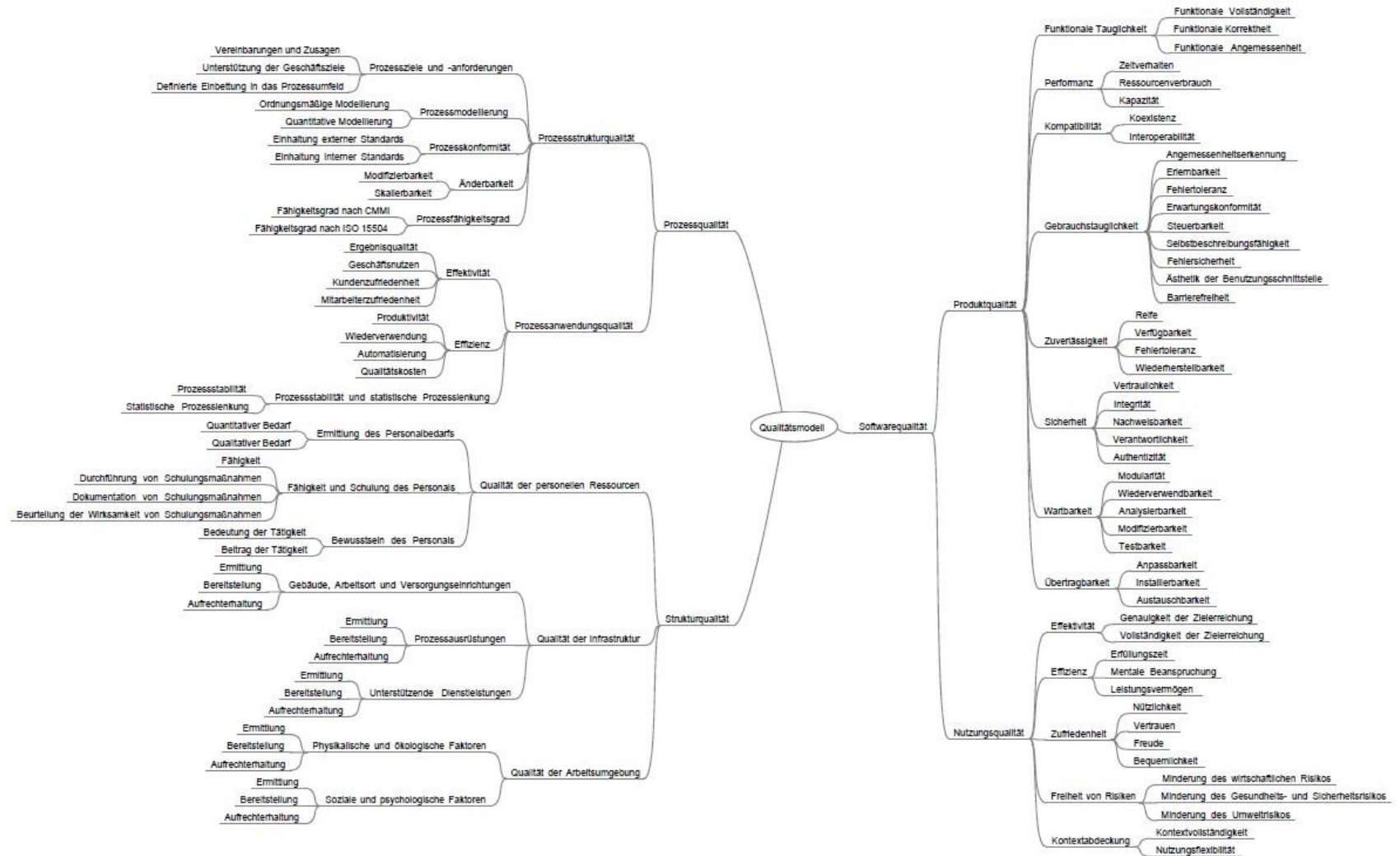
Software-Engineering-Best-Practices

- Auswahl
- Identifikation von Anpassungsbedarf
- Erarbeitung neuer Praktiken
- Werkzeugunterstützung
- Evaluierung
- Publikation (kostenlose Nutzung)

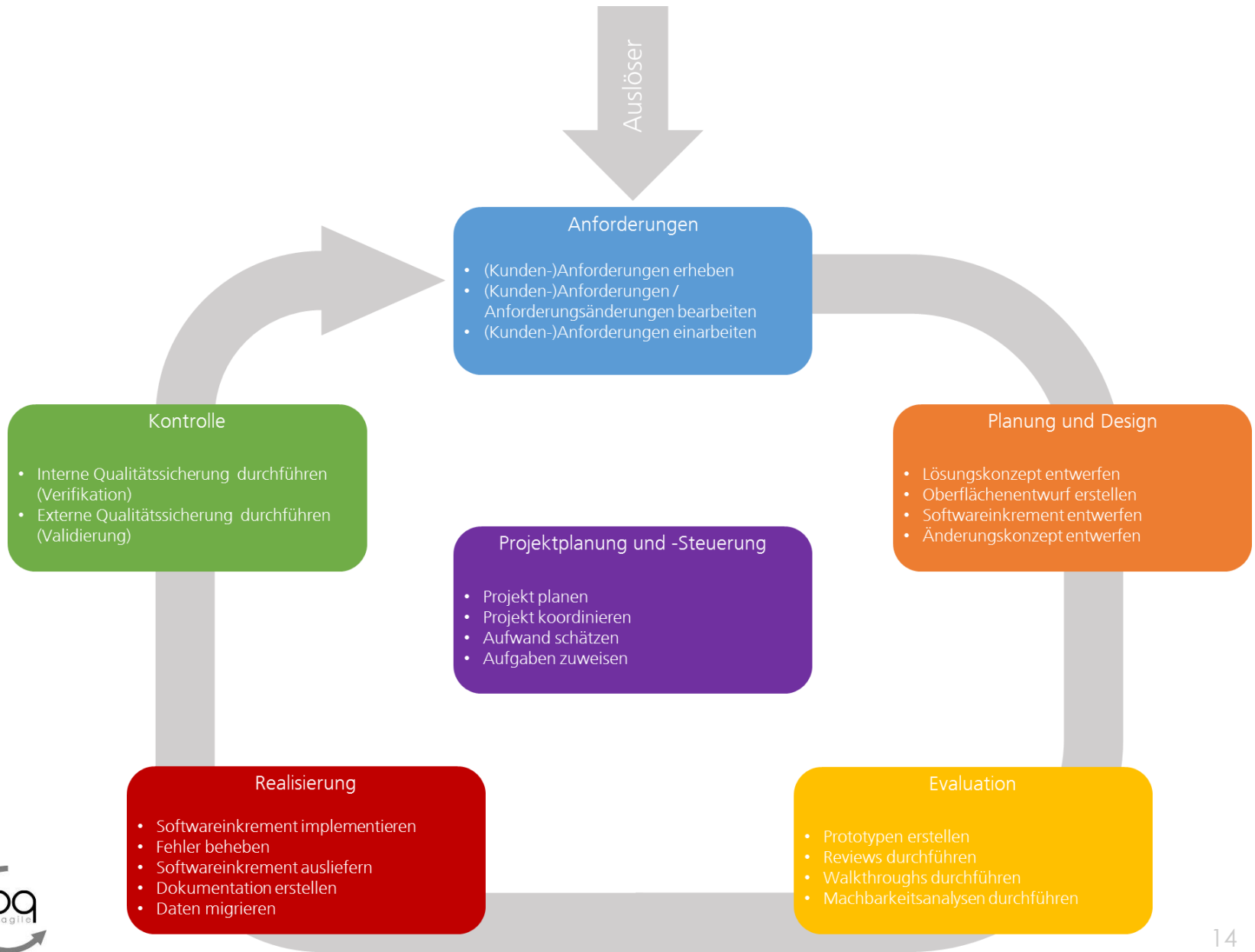
Methodische Vorarbeiten



Qualitätsmodell



Agiler Referenzprozess



Entwicklung des Best-Practice-Templates

- Sammlung von Beschreibungsaspekten
 - Literatur
 - Erfahrung
 - vergleichbare Projekte
- Diskussion, Abstimmung, Test
- Dokumentation der Patterns

User Experience

Titel

Bereich
Bereich des Referenzprozesses

Aktivität
Aktivität des Referenzprozesses

Ziele

- Ziel 1
- Ziel 2
- ...

Aufwand ▲

Schwierigkeit ▶

Agilitätsfaktor ▼

Motivation/Problemstellung
Motivation für den Einsatz der Best Practice bzw. Problemstellung, bei der sich deren Einsatz empfiehlt

Kurzbeschreibung
Kurzbeschreibung der Best Practice




Abbildung 1: Beispiel einer Abbildung

Input

- (optional) benötigter Input für die Ausführung der Best Practice, z.B. aus vorausgehenden Aktivitäten, inkl. Format

Output

- Ergebnisse, die durch den Einsatz der Best Practice erreicht werden sollen


Rahmenbedingungen

Ausführender
Rolle, die die Best Practice ausführt

Werkzeuge, Hilfsmittel
(optional) zur Ausführung der Best Practice benötigte Werkzeuge und Hilfsmittel

Vorkenntnisse/Erfahrungen

Ort/Umgebung

 www.pq4agle.de

1

Best-Practice-Template

- Titel
- Bereich und Aktivität des Referenzprozesses
- Ziele
 - Ziel 1
 - Ziel 2
 - ...
- „Ampeln“ für Aufwand, Schwierigkeit, Agilitätsfaktor
- Motivation/Problemstellung
- Kurzbeschreibung
- Input/Output
- Rahmenbedingungen
 - Ausführer, Tools, Vorkenntnisse, Ort ...

User Experience

Titel

Bereich
Bereich des Referenzprozesses

Aktivität
Aktivität des Referenzprozesses

Ziele

- Ziel 1
- Ziel 2
- ...

Aufwand ▲

Schwierigkeit ▢

Agilitätsfaktor ▼

Motivation/Problemstellung
Motivation für den Einsatz der Best Practice bzw. Problemstellung, bei der sich deren Einsatz empfiehlt

Kurzbeschreibung
Kurzbeschreibung der Best Practice




Abbildung 1: Beispiel einer Abbildung

Input

- (optional) benötigter Input für die Ausführung der Best Practice, z.B. aus vorausgehenden Aktivitäten, inkl. Format

Output

- Ergebnisse, die durch den Einsatz der Best Practice erreicht werden sollen


Rahmenbedingungen

Ausführender
Rolle, die die Best Practice ausführt

Werkzeuge, Hilfsmittel
(optional) zur Ausführung der Best Practice benötigte Werkzeuge und Hilfsmittel

Vorkenntnisse/Erfahrungen

Ort/Umgebung

 www.pq4agle.de

1

Best-Practice-Template

- Vorgehensweise
 - Vorbereitung, Durchführung
 - Nachbereitung
- Gütekriterien/Empfehlungen
- Risiken
- Einordnung in den agilen Referenzprozess
 - Vorgänger, Nachfolger, Alternativen
- Einordnung in das Qualitätsmodell

User Experience

Titel

Bereich: Bereich des Referenzprozesses Aktivität: Aktivität des Referenzprozesses

Ziele **Aufwand** ▲

- Ziel 1
- Ziel 2
- ...

Schwierigkeit ▶

Agilitätsfaktor ▼

Motivation/Problemstellung
Motivation für den Einsatz der Best Practice bzw. Problemstellung, bei der sich deren Einsatz empfiehlt

Kurzbeschreibung
Kurzbeschreibung der Best Practice





Abbildung 1: Beispiel einer Abbildung

Input **Output**

- (optional) benötigter Input für die Ausführung der Best Practice, z.B. aus vorausgehenden Aktivitäten, inkl. Format
- Ergebnisse, die durch den Einsatz der Best Practice erreicht werden sollen

Rahmenbedingungen

Ausführender Rolle, die die Best Practice ausführt	Werkzeuge, Hilfsmittel (optional) zur Ausführung der Best Practice benötigte Werkzeuge und Hilfsmittel
Vorkenntnisse/Erfahrungen	Ort/Umgebung

 www.pq4 agile.de 1

Entwicklung von Best Practices

- Sammlung (verbreiteter) Methoden, Praktiken und Aktivitäten
 - im Bereich UUX 130 Aktivitäten
- Auswahl erfolgversprechender Kandidaten
 - im Bereich UUX 23 Best-Practice-Kandidaten
- Identifikation von Anpassungsbedarfen
 - Anpassung auf Eigenschaften agiler Entwicklungsvorgehen
- Entwicklung neuer Best Practices (nach Bedarf)
- Dokumentation der Best Practices

Best Practices UUX

- Informationsarchitektur erstellen
- Kontextuelles Benutzerinterview erstellen
- Usability-Review durchführen

(in Kürze)

- Benutzer charakterisieren
- Lösungsideen konzipieren
- Usability-Patterns verwenden
- Funktionalen Prototyp erstellen
- Prototyp im Nutzungskontext evaluieren
- ...

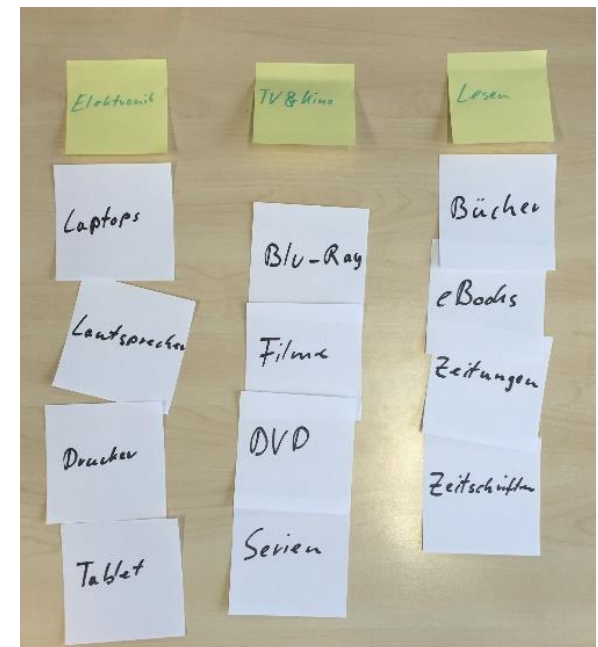
Best Practices Usability / User Experience

Beispiel „Informationsarchitektur erstellen“

- Ziele
 - Ermöglichung einer leichten und intuitiven Bedienung
 - Effiziente Interaktion mit der Software-Anwendung
 - Verbessern der pragmatischen Qualität des Systems
- Motivation/Problemstellung
 - Benutzer verbringt lange Zeit mit der Navigation und Suche gewünschter Inhalte
 - Inhalte werden häufig nicht optimal repräsentiert, sodass eine effiziente Nutzung der Software nicht möglich ist

Beispiel „Informationsarchitektur erstellen“

- Kurzbeschreibung
 - Start: schlechte Navigation und Suche durch die Software sowie schlecht repräsentierte Inhalte
 - Sammlung der zur Verfügung stehenden Inhalte
 - Aufteilung, Strukturierung und Benennung dieser Inhalte
 - Organisation der Inhalte → an welcher Stelle werden welche Informationen angezeigt



Beispiel „Informationsarchitektur erstellen“

- Unterschiedliche Ansatzpunkte
 - Top-Down IA
 - Beschäftigt sich mit dem Weg durch die Software vom Einstiegspunkt aus betrachtet
 - Bottom-Up IA
 - Beschäftigt sich mit der Navigation durch die Software von einer Detailseite aus betrachtet
- Card Sorting zur Erstellung der Navigationswege, Benennung, Kategorisierung und Strukturierung der Inhalte – Repräsentation in Baumform
- Struktur des Inhalts wird in Form von Wireframes festgelegt

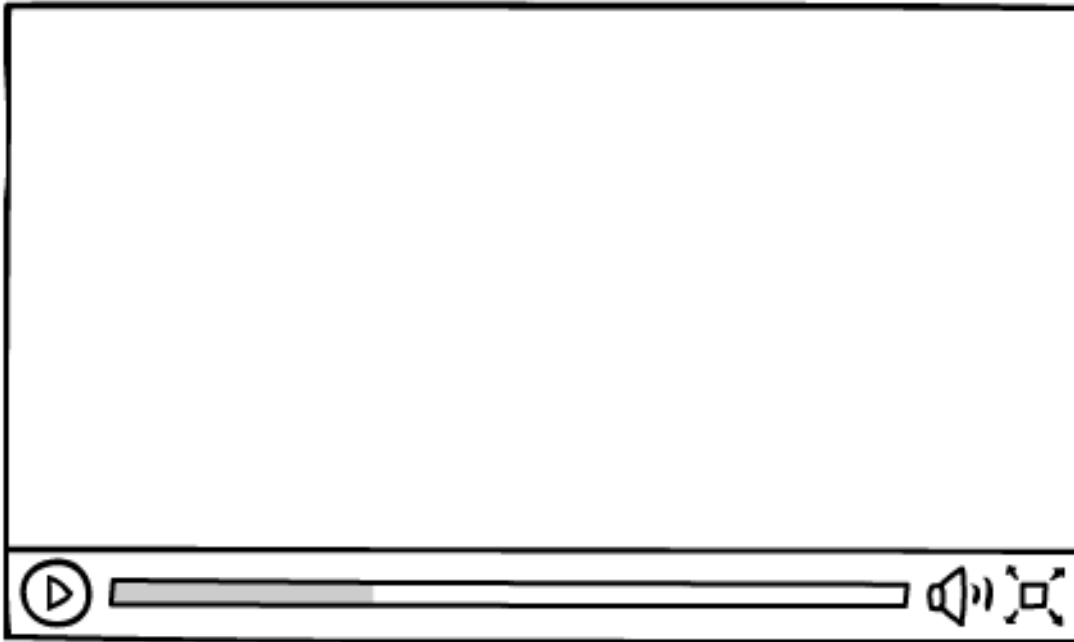


Suche

[Peter Mustermann](#) [Logout](#)

[Home](#) | [Videos](#) | [Blog](#) | [Blog](#)

[Home](#) > [Videos](#) > Xyz



John Smith
[Folgen](#)

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat,

Beispiel „Informationsarchitektur erstellen“

- Gütekriterien/Empfehlungen
 - Früher Kontakt zu Endanwendern bereits während Anforderungsermittlung aufbauen
 - Bei Card-Sorting benannte Begriffe sollten Endanwender aus ihrer täglichen Arbeit kennen
 - Erstellen der IA 1-sprint ahead
- Risiken
 - Keine repräsentative Endanwendergruppe
 - Zu wenig Wissen über Inhalte

Beispiel „Template-basierte UI Konzeption“

- Ziele
 - Entwicklung einer konsistenten Benutzeroberfläche
 - Schnelle und konsistente Umsetzung von Änderungswünschen bezüglich des UI-Designs
 - Erreichung einer Verfolgbarkeit von Änderungen im UI-Design
 - Schaffung einer Grundlage für eine effektive und effiziente Implementierung von UI-Designs
 - Verbesserung der Kommunikation zwischen UI-Designer und Programmierer
- Motivation/Problemstellung
 - Inkonsistenz zwischen Design und Code und zwischen Screens durch häufige Änderungen
 - Fehlende Nachverfolgbarkeit von Implementierung und bereits konzipierten Prototypen

Beispiel „Template-basierte UI Konzeption“

- Kurzbeschreibung
 - Erstellung und Verwendung von Gestaltungsvorlagen
 - Vorlagen definieren Grundgerüst einer Anwendung durch instanziiierbare UI-Elemente (z.B. Schaltflächen, Tabellen, Header, Navigationsmenüs)
 - Festlegen einer hierarchischen Struktur dieser Elemente
 - Spezifizieren sämtlicher Attribute der UI-Elemente
 - Traceability zwischen Templates, Prototypen und Implementierung

Beispiel „Template-basierte UI Konzeption“

- Vorgehensweise
 - Festlegen, welche UI-Elemente im Verlauf des Projekts verwendet werden sollen
 - Alternativ: erstellen einzelner Screens der Software und Identifikation der UI-Elemente
 - Unterscheidung zwischen Basiselementen (Text, Schaltfläche, Eingabefeld) und komplexen, zusammengesetzten UI-Elementen (Header, Navigationsmenü, Suchfeld ...)
 - Spezifikation der Attribute dieser Elemente (Form, Farbe, Größe ...)
 - Festlegen der Hierarchie der Elemente (wie beeinflussen sich diese gegenseitig, welche Attribute werden vererbt ...)

Beispiel „Template-basierte UI Konzeption“

- Vergabe von eindeutigen IDs zur Nachverfolgbarkeit
- Erstellung der Prototypen durch Instanziierung zuvor festgelegter Elemente
- Verweise auf verwendete IDs (Template – Prototyp – Code)
- Änderungen an Template, Prototyp oder Implementierung können so entsprechend nachverfolgt werden

Logo

35x35

Label

35x35

Label

35x35

Label

35x35

Label

35x35

Label

35x35

Label

35x35

Label

35x35

Label

35x35

Label

35x35

Label

(Optional)
137x58
Padding-Left/Right 10px

Label

Label

Label

Label

Label

Label

Label

Label

Top Information / Headlines / Search / Progress

Variable Height
Width = 1230 px
Margin-Bottom = 10px

Content Area

Variable Height
Width = 1230 px
Margin-Bottom = 10px

Button Area
Height = 35px
Width = 1230px

Beispiel „Template-basierte UI Konzeption“

- Gütekriterien/Empfehlungen
 - Vorwiegend für größere Entwicklungsprojekte geeignet (höhere Anzahl an Prototypen, verwendeter UI-Elemente und implementierten Codes)
 - Erstellung der Hierarchie sehr zeitaufwändig und daher eher für große Projekte geeignet
 - UI-Element-Hierarchie toolgestützt erstellen und verwalten
- Risiken
 - Bei kleinen Projekten keine Amortisation des Aufwands für Erstellung der Templates


Und jetzt: Ausprobieren! Feedback geben!

Best Practices | PQ4Agile - x

www.pq4agile.de/best-practices/

Apps ACM Digital Library IEEE IEEE Xplore - Home Google Scholar Fraunhofer eLib SIGMA und SIGMA S... Save to Mendeley Google Keep Sp


START PQ4AGILE PROJEKT BETEILIGTE AKTUELLES TERMINE **BEST PRACTICES** SERVICE KONTAKT

 Produktqualität für Agile Softwareentwicklung

Best Practices

[Deutsch](#)
[English](#)

Ziel des Projekts PQ4Agile ist es, agilen Entwicklern mit erprobten Best Practices eine möglichst effiziente und systematische Unterstützung bei der Erreichung nichtfunktionaler Qualitätsmerkmale zu bieten. Die Sammlung umfasst Best Practices aus den Bereichen Requirements Engineering, Softwarearchitektur und Usability/User Experience sowie unterstützende Praktiken. Alle beschriebenen Methoden und Aktivitäten können von den Mitgliedern eines agilen Teams flexibel und nach Bedarf eingesetzt und leicht in agile Entwicklungsprozesse integriert werden.





Vielen Dank für die Aufmerksamkeit

Dominik Magin, Fraunhofer IESE (dominik.magin@iese.fraunhofer.de)

Hartmut Schmitt, HK Business Solutions (schmitt@hk-bs.de)



www.pq4agile.de

