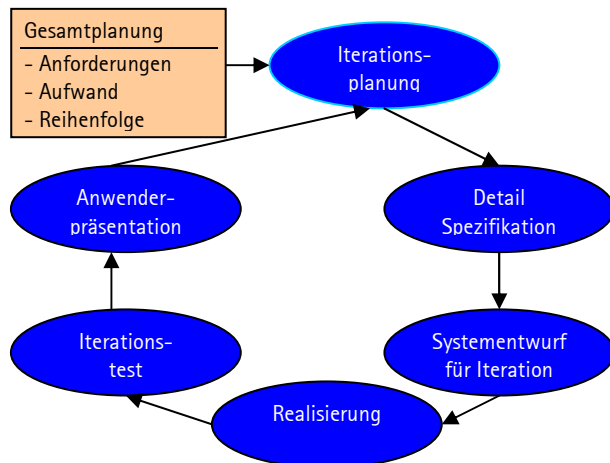


Agile Projekte brauchen flexibles Requirements-Engineering Use Cases erfüllen diese Aufgabe hervorragend

Agile Projekte zeichnen sich dadurch aus, dass die Realisierung der Gesamtfunktionalität in Iterationen erfolgt. In festen Zeitintervallen wird Software erstellt, die einen sinnvollen Funktionsumfang für die Anwender enthält. Das Ergebnis wird vorgestellt und das Entwicklungsteam bekommt eine entsprechende Rückkopplung. Bei diesem Vorgehen muss vor dem Start



der Realisierung kein Pflichtenheft vorliegen, in dem jede Anforderung bis ins Detail spezifiziert ist. Die funktionalen Anforderungen müssen aber beschrieben sein, damit die einzelnen Iterationen geplant werden können. Für jede Anforderung müssen die zur Planung notwendigen Informationen dokumentiert sein. Für die Realisierung benötigte Detailinformationen werden erst später ergänzt. Die Methode, mit der Anforderungen beschrieben werden, muss diese Flexibilität bieten. Use Cases eignen sich für das agile Vorgehen hervorragend. Jeder Use Case wird durch Attribute beschrieben. Informationen können zu dem Zeitpunkt ergänzt werden, an dem sie benötigt werden.

Planung in Agilen Projekten

Für die Planung stellen Use Cases folgende Attribute zu jedem Anwendungsfall zur Verfügung:

- Primärer Akteur: Hauptanwender
- Ziel und Inhalt: Kurze Inhaltsangabe
- Ebene: Beschreibungsebene
- Akteure und Interessen: Wer verfolgt welche Ziele?
- Vorbedingung: Bedingungen vor Start des Use Cases
- Minimal Garantie: Minimalergebnis
- Auslöser: auslösendes Ereignis
- Hauptobjekt: zentrales Objekt
- Business Priorität: Priorität der Anwender
- Komplexität: Entwicklungskomplexität

Das **Gesamtmodell** aller Use Cases beschreibt den Umfang des Projektes. Mit dem Ebenenkonzept werden die Anforderungen strukturiert.

Die Attribute *Ziel und Inhalt*, *Akteure und Interessen* sowie *Komplexität* liefern den Input der **Aufwandsschätzung**.

In welcher **Reihenfolge** werden die Anforderungen Iterationen zugeordnet? Use Cases mit hoher *Business Priorität* werden frühzeitig realisiert (Easy Wins). Aber auch Anforderungen mit hoher *Komplexität* müssen rechtzeitig umgesetzt werden, um Risiken zu minimieren. Die *Vorbedingungen* für die Realisierung eines Use Case müssen allerdings immer erfüllt sein. Es kann sinnvoll sein, Use Cases die dasselbe *Hauptobjekt* bearbeiten, in einer Iteration zu realisieren.

Die *Minimal Garantie* liefert wichtige Hinweise, wie die **Funktionalität eines Use Cases auf Iterationen** verteilt werden kann.

Detaillierte Spezifikation der Funktionalität einer Iteration

Die Planung liefert die Use Cases, die in der nächsten Iteration realisiert werden sollen. Bisher haben die Use Cases allerdings noch keine Detailinformationen. Es müssen noch weitere Informationen vor der Realisierung ergänzt werden:

Ablaufbeschreibung:

Schritte die im Haupt-Erfolgsszenario – dem Normalfall – durchlaufen werden

Variationen:

Die Varianten des Normalfalls werden mit ihren einzelnen Schritte beschrieben.

Fehlerbeschreibungen:

Im Fehlerfall wird jeder Schritt beschrieben, der durchlaufen werden soll.

Die detaillierte Beschreibung, wie eine Anforderung umgesetzt werden soll, erfolgt also erst direkt vor der Realisierung der Anforderung in der nächsten Iteration.

Zusätzliche Informationen für die Realisierung

Use Cases definieren die funktionalen Anforderungen an das System. Damit liefern sie zentrale Informationen für die Realisierung, aber nicht alle. So muss das Klassenmodell wie die Use Cases vor jeder Iteration weiter detailliert werden. Schnittstellen zu anderen Systemen müssen berücksichtigt werden. Nicht funktionale Anforderungen z.B. an das GUI-Design oder die Performance müssen vor der ersten Realisierung beschrieben sein. Das iterative Vorgehen liefert auch hier eine schnelle Rückkopplung mit den Anwendern.

Vergleich zu klassischen Pflichtenheften

Bei dem leichtgewichtigen Vorgehen steht in jeder Projektphase nur die Dokumentation zur Verfügung, die auch benötigt wird. Das Konzept der Use Cases ist so flexibel, dass notwendige Informationen ergänzt werden können und die Gesamtdokumentation

erhalten bleibt. Für die Planung wird beschrieben, **WAS** im Projekt umgesetzt werden soll. Erst vor der konkreten Umsetzung wird definiert, **WIE** realisiert werden soll. Dadurch entstehen zusätzlich Vorteile, da folgende Aspekte zeitnah berücksichtigt werden können:

- aktuelle Rahmenbedingungen - Gesetze / Verordnungen
- technische Weiterentwicklungen
- bisherige Erfahrungen aus dem Projekt
- zusätzliche Wünsche der Anwender

Requirements-Engineering bei GEDOPLAN

GEDOPLAN hat Use Cases in verschiedenen Projekten zur Beschreibung der Anforderungen eingesetzt. Wir setzen die full dressed Form – tabellarische Form – der Use Cases ein. Jeder Anwendungsfall ist in derselben, leicht verständlichen Struktur dargestellt. Dadurch wird die Kommunikation mit dem Anwender vereinfacht. Die Dokumentation erstellen wir in Word. Wir nutzen die Use Cases für Aufwandsschätzungen und als Vorgabe für die Realisierung. Die Tests müssen die in den Use Cases definierten Ergebnisse liefern. Aus der Use Case Beschreibung generieren wir die Anwenderdokumentation.



Über den Autor

Reinhard Brüggemeyer ist Berater und Projektleiter im Enterprise Java Team bei GEDOPLAN.

Kontakt:

GEDOPLAN GmbH
Stieghorster Str. 60
33605 Bielefeld

Fon: 0521/2088910

Fax: 0521/2088945

Email: info@gedoplan.de