Softwareentwicklungsprozesse

by

Günter Kolousek

Tätigkeiten – SW-Entwicklungsproz.

- Anforderungsanalyse (engl. requirements engineering)
  - Erfassung und Beschreibung der Anforderungen
  - Anforderungsdokument oder Lastenheft (Auftraggeber)
  - Pflichtenheft (akkordiert zw. Auftraggeber und Auftragnehmer)
  - Welche Funktionaliät soll das System aufweisen?
- Analyse (engl. analysis)
  - Systemelemente und deren Beziehungen zur Umwelt
  - Ist-Zustand vs. Sollzustand
  - Wie ist die Domäne aufgebaut?
- Entwurf (engl. design)
  - Architekturdokument
  - Wie wird die Software gebaut?
- Implementierung (engl. implementation)
- Testen
- Deployment (dt. Einsatz)

Softwareentwicklungsprozesse

- Wasserfallmodell
- Rational Unified Process (RUP)
  - Open Unified Process (OpenUP)
    - angelehnt an RUP (von Eclipse)
  - Agile Unified Process (AUP)
- V-Modell XT
- XP (eXtreme Programming)
- Test Driven Delepment (TDD)
- Scrum
- Kanban

RUP - Prinzipien (best practices)

- Iterative Softwareentwicklung
- Projektbegleitendes Qualitätsmanagement
- Komponentenbasierte Architektur $\rightsquigarrow$ Testen
- Visuelle Modellierung $\rightsquigarrow$ UML
- Kontrolliertes Änderungsmanagement
- Anforderungsmanagment

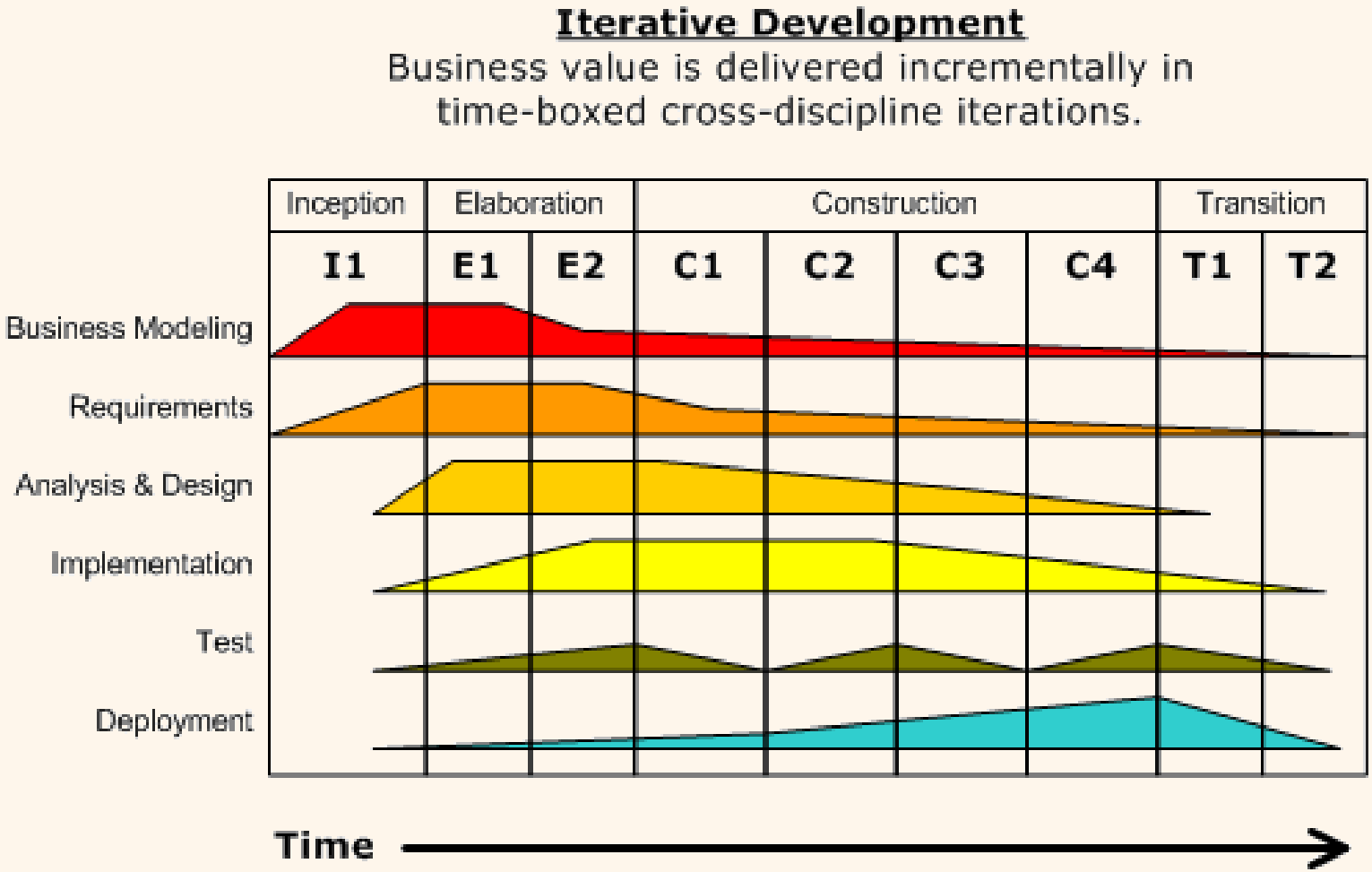
RUP - Phasen

1. Inception (dt. Anfang, Beginn, Gründung)
   Konzeption, Ausarbeitung einer Vision, eines klaren Zieles
2. Elaboration (dt. Ausarbeitung)
   Erstellung einer Architektur, Ausarbeitung der Use Cases
3. Construction
   Entwicklung und Testen
4. Transition
   Übergabe und Auslieferung

RUP - Tätigkeiten

- Business Modelling (dt. Geschäftsprozessmodellierung)
  Dokumentation und Optimierung der Geschäftsprozesse
- Requirements Engineering (dt. Anforderungsanalyse)
- Analysis & Design (dt. Analyse und Entwurf)
- Implementation (dt. Umsetzung)
- Test
- Deployment

RUP - Übersicht



Quelle: de.wikipedia.org

Produktvision

- Zielgruppen
- Bedürfnisse und Probleme der Zielgruppe
- Skizze des Produktes, das die Bedürfnisse und Probleme löst.
  - Kann 3-5 Key-Features enthalten
- wird im Zuge von Scrum verwendet

Systemidee

- Eckdaten der Systemidee:
  - ca. halbe Seite
  - sollte auf einen ”Produktkarton” aufdruckbar sein und dementsprechend präsentiert werden können
- Inhalt der Systemidee:
  - Name des Produktes
  - beschreibt was mit dem System erreicht werden soll
  - enthält eine Auflistung der 5-15 wichtigsten Eigenschaften und Leistungsmerkmale
  - enthält Randbedingungen (SW & HW Anforderungen, Finanzen, Termine, organisatorische und personelle Voraussetzungen,…)

Anforderung

- Anforderung beschreibt ein oder mehrere
  - Eigenschaften oder Verhaltensweisen,
  - die stets erfüllt sein müssen
- Art, d.h. Unterteilung in:
  - funktional
  - nichtfunktional

Nichtfunktionale Anforderungen

- Benutzbarkeit
- Performance
- Zuverlässigkeit
- Wartbarkeit
- Administrierbarkeit
- Rahmenbedingungen

Aufbau von Anforderungen

- Name
- Art
- Beschreibung
- Stabilität: absolut stabil, stabil, instabil, flüchtig
- Verbindlichkeit: Pflicht, Wunsch, Absicht, Vorschlag
- Priorität: hoch, mittel, niedrig
- Detailbeschreibung: Motivation, Ursache, Hintergrund, Ansprechpartner, Unterlagen, Beispiele, Randbedingungen,…
- Verweise, Änderunghistorie, Bemerkungen

Anwendungsfall

- Stellt strukturierte Beschreibung der Interaktion mit dem System dar
- siehe UML Anwendungsfalldiagramm!
- Arten von Anwendungsfällen (engl. use case)
  - Geschäftsanwendungsfälle
    - geschäftlicher Ablauf ohne systemtechnische Umsetzung
    - $\rightarrow$ Geschäftsprozessmodellierung
  - Systemanwendungsfall
    - beschreibt Interaktion mit System (HW, SW)
- Text!

Aufbau eines Anwendungsfalles

- Name
- Kurzbeschreibung
- Akteure
- Auslöser
- Ergebnisse
- Hauptablauf
- Eingehende Daten, Ausgehende Daten
- Ausnahmen, Fehlersituationen
- Vorbedingungen, Nachbedingungen
- Offene Punkte, Änderungshistorie, Bemerkungen
- Anhänge wie Diagramme, ext. Dokumente
- meist auch: Aktivitätsdiagramme

Beispiel eines Anwendungsfalles

Name Termin erfassen

Kurzbeschreibung Der Benutzer erstellt einen Termin mit einem Datum, einer Uhrzeit, einer Liste der Teilnehmer und einem Text

Akteure Benutzer

Hauptablauf

1. Datum und Zeit werden ausgewählt und müssen in der Zukunft liegen
2. Gewählter Termin wird mit den Teilnehmern auf Kollisionen überprüft
3. Der Kalender wird aktualisiert
4. Die Teilnehmer werden per E-Mail bzw. Fax verständigt

Nachbedingungen keine Überschneidungen der Termine für die angegebenen Teilnehmer

User-Stories

- Aufbau: `Als <Rolle> möchte ich <Ziel/Wunsch>, um <Nutzen>`
  - kürzere Version: `Als <Rolle> möchte ich <Ziel/Wunsch>`
- Beispiel:

  ”Als Benutzer möchte ich einen Termin erfassen, um zwei Tage im Voraus erinnert zu werden.”

Lizenzen

- Closed-Source
- Open-Source
  - copyleft
    - Veränderung oder Integration in eigenen Quellcode $\rightarrow$ selbe Lizenz
    - bedeutet keine Abwesenheit von Copyright, im Gegenteil Copyright ist notwendig (wer hat…)!
  - permissive
    - Software kann (auch abgeändert) unter jeder beliebigen Lizenz verwendet werden
    - Beispiele: BSD, Apache, MIT
- Public Domain
  - Frei von Urheberrechten (Copyright)
    - in Kontinentaleuropa nicht möglich?!

Lizenzen – Copyleft

- Veränderung oder Integration in eigenen Quellcode $\rightarrow$ selbe Lizenz
- starkes copyleft, wie z.B. GPL: Verkauf oder Einbindung in andere (eigene) SW, dann Quellcode muss zur Verfügung gestellt werden
  - extrem starkes copyleft, wie z.B. AGPL (Affero GPL): Verwendung der Software über Netzwerk, dann Quellcode muss weitergegeben werden!
- schwaches copyleft, wie z.B. LGPL (Lesser GPL): Quelloffene SW kann in proprietärer SW genutzt werden, solange Benutzer diese selbständig verwenden kann $\rightarrow$ dynamisches (oder statisches) Linken. Die Lizenz des proprietären Teiles muss nicht unter die Open-Sourze-Lizenz gestellt werden.
  - Installationsanleitungen, damit Benutzer SW mit eigener Version linken kann.

Lizenzen – Versionsproblematik

- Software unterschiedlicher Lizenzen oder Lizenzversionen sind meist schwer miteinander zu kombinieren
  - Da oft verlangt wird, dass die gesamte Software unter eine Lizenz gestellt wird (auch bzgl. der Versionen, wie z.B. GPLv2 vs. GPLv3)