Was ist "SOA"? Eigenschaften? Paradigmen?

Service-orientierte Architektur
- Geschäftsprozess-orientiert
- Anwendungen werden durch mehrere Services gelöst
- Services in einer Anwendung sollen auch aus anderen Anwendungen heraus aufgerufen werden können
    - Reuse
    - Modularisierung
- Zusammenstellung von Services in bestimmten Abläufen zur Umsetzung von Geschäftsprozessen (Orchestrierung)
- Services existieren auf mehreren Abstraktionsebenen (Business Service, Component Service, technisches Service, usw.)
- Beschreibt eine Architektur, keine Technologie (z.B. können Webservices verwendet werden zur Realisierung einer SOA – aber auch z.B. CORBA)

Paradigmen der Service-orientierten Architektur
- Loose Coupling
    - Niedrige Kopplung der Services ermöglicht hohen Reuse und Flexibilität
- Abstraktion
    - Reduktion von Neuentwicklungen – Spannungsfeld Abstraktion vs. Technische Anforderungen (z.B. KÖNNEN manche technische Services einfach nicht abstrakt gelöst werden)
- Standardisierung
    - Standardisierung erlaubt die Zusammenarbeit von Services
- Composable
    - Standardisierte Services können zu Komponenten zusammengefasst werden

Was versteht man unter der Spezialisierung von Frameworks? Wie geschieht die bei Whitebox Frameworks und Blackbox Frameworks?

Ein <u>Framework ist ein Applikationsskelett</u>, welches vom Entwickler angepasst (spezialisiert) werden kann, um Anforderungen optimal umzusetzen.

Whitebox Frameworks
- Innere Struktur muss sichtbar sein um es spezialisieren zu können
- Spezialisierung erfolgt (meist) durch Vererbung

Blackbox Frameworks
- Enthält stabile Hot Spots
- Spezialisierung kompositionsbasiert
    - Komponenten werden an das Framework übergeben

Erläutern sie das Vorgehen bei einem "SQL Injection" Angriff - Wie kann man sich davor schützen?

Eine Manipulation des SQL Statements durch einen Angreifer ist möglich.
- Einfaches Hochkomma: “te’st”. Fehlermeldung der Datenbank wird angezeigt. Schwachstelle erkennbar. Zusätzlich Informationen zur Datenbank und zum Aufbau des SQL Statements werden ausgegeben.
- Verwenden von Tautologien wie ’or’1’=’1 was zu
SELECT * FROM users WHERE user_name=’’ or ’1’=’1’ führt
- Auslesen von Daten aus weiteren Tabellen mittels UNION oder Sub
Queries
- “’UNION SELECT * FROM system”

Gegenmaßnahmen:
- Jede Eingabe muss validiert werden. Es ist nicht ausreichend nur Formularfelder zu validieren. Injection z.B. auch durch HTTP
Header-Felder
- Filterung spezieller Datenbankzeichen wie z.B. einfache Hochkomma
„ Blacklist Filtering“
- Besser: Prüfung der Eingaben auf gültige Zeichen „Whitelist
Filtering“
- Precompiled Statements verwenden. Das Statement wird in der Datenbank mit Platzhaltern kompiliert. Spätere Manipulation des
Statements durch Eingaben nicht mehr möglich.
- Minimierung des Schadens bei Angriffen: Einschränkung der Rechte in der Datenbank " Principle of ”Least Privilege“.

Erläutern sie die 5 Arten von kontrollflussorientierten Testverfahren (Co,..C4 Test). Warum ist eine Testabdeckung von 100% nicht immer möglich/sinnvoll?

Anweisungsüberdeckungstest (C0 -Test)
- Alle Knoten eines Kontrollflussgraphen werden von zumindest einem Testfall abgedeckt

Zweigüberdeckungstest (C1-Test)
- Zumindest jede Kante des Kontrollflussgraphen muss einmal durchlaufen werden

Einfacher Bedingungsüberdeckungstest (C2-Test)
- Die Auswertung jeder einzelnen atomaren Entscheidung muss einmal den Wert wahr und einmal den Wert falsch annehmen

Mehrfach-Bedingungsüberdeckungstest (C3-Test)
- Es wird versucht, zusätzlich zu den atomaren Teilentscheidungen und der Gesamtentscheidung auch alle zusammengesetzten Teilentscheidungen gegen wahr und falsch zu prüfen

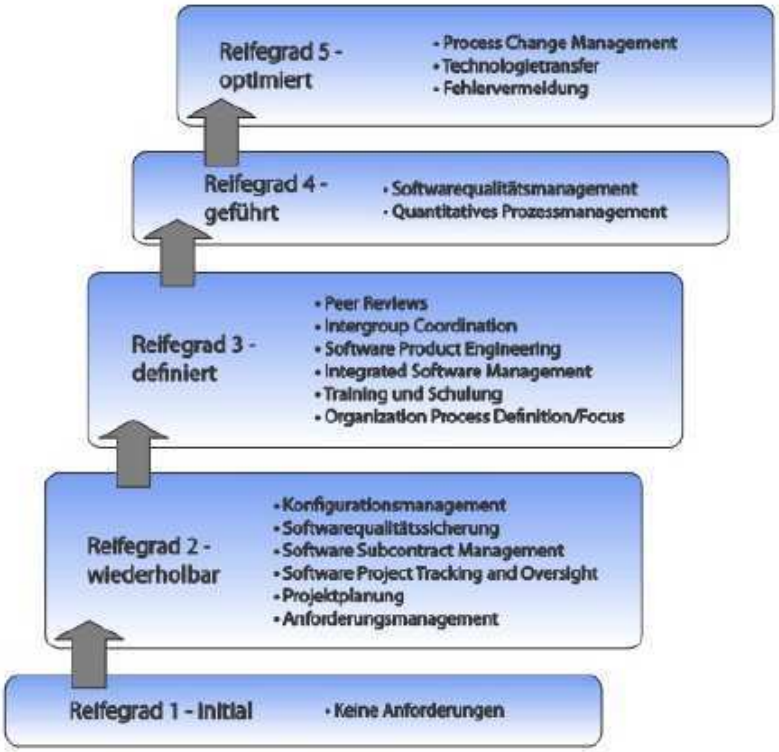
Pfad-Überdeckungstests (C4-Test)
- Es müssen alle unterschiedlichen Pfade des Testobjektes in zumindest einem Testfall durchlaufen werden

Testfallselektion notwendig, da eine vollständige taxative Aufzählung aller möglichen Systemzustände aufgrund der Komplexität von Systemen unmöglich ist.

Wozu dient CMM? 5 Reifegrade.

• Ziel, die Vergleichbarkeit und Qualität von Softwareprozessen im Rahmen der Vergabe an externe Lieferanten zu verbessern
• 2003 ein Nachfolger, das CMMI (Capability Maturity Model Integrated), entwickelt
• CMMI Liefert eine Analyse des **Ist- Zustandes** eines Unternehmens und ist kein international anerkanntes Zertifikat wie ISO 9000-3
• Ein CMM-Assessment beurteilt die Fähigkeit einer Organisation, Software unter bestimmten Rahmenbedingungen erstellen zu können

Reifegradstufen:



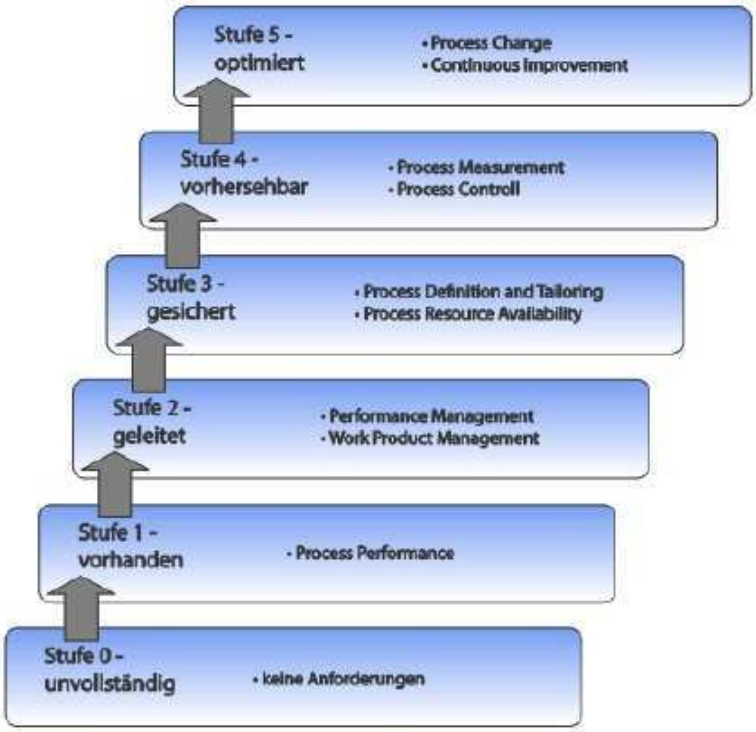
Erklären Sie die wesentlichen Unterschiede zwisch en einem Framework und einer Bibliothek-

- Eine Bibliothek bietet wiederverwendbare Funktionalität an, während ein Framework wiederverwendbares Verhalten bereitstellt
- Methoden von Bibliotheken werden durch den eigenen Code aufgerufen, während bei der Verwendung von Frameworks der eigene Code durch das Framework aufgerufen wird
- Frameworks bestehen aus abstrakten Klassen und bieten Funktionalität in (teilweise) implementierten Methoden an. Diese Methoden können Funktionalität aus Bibliotheken beziehen.

Wozu dient SPICE? 6 Reifegrade.

• SPICE = Software Process Improvement for Capability Determination
• Grundsätzlich sind SPICE und CMMI sehr ähnlich aufgebaut.
Zielsetzungen und Forderungen dieser beiden Modelle überdecken
sich zu ca. 80 – 90%.
• SPICE besteht, genauso wie CMMI, aus einem zweidimensionalen Referenzmodell.
• Zu diesem Referenzmodell gehören eine Prozessdimension und eine Fähigkeitsdimension zur Reifegradbestimmung der einzelnen Prozesse

Reifegradstufen:



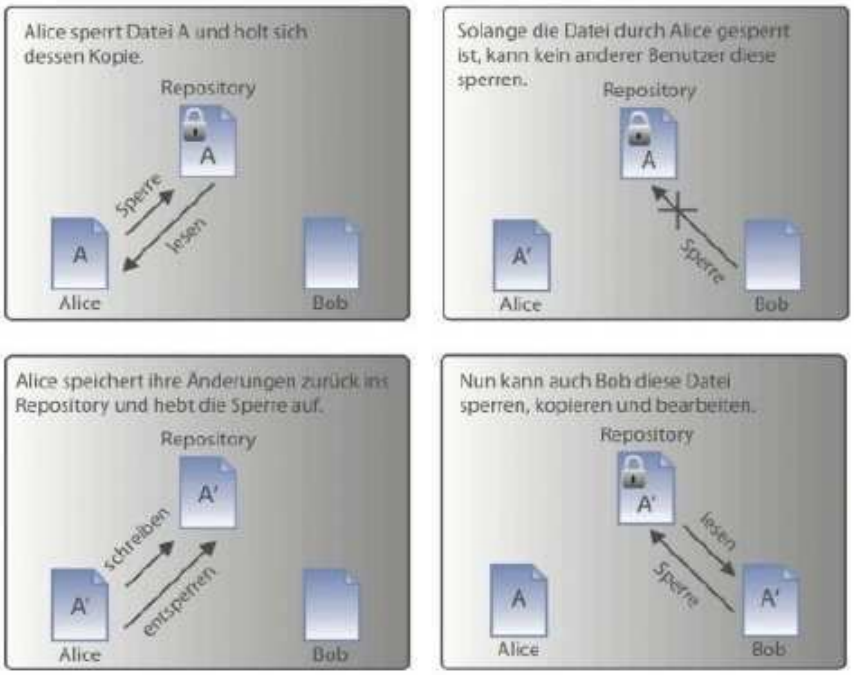
Was versteht man unter "Stakeholder"? Nennen sie fünf typische Stakeholder in einem Softwareprojekt.

Stakeholder = „Anteilhalter“ der ein Interesse im Projekt vertritt und damit eine Anforderungsquelle darstellt

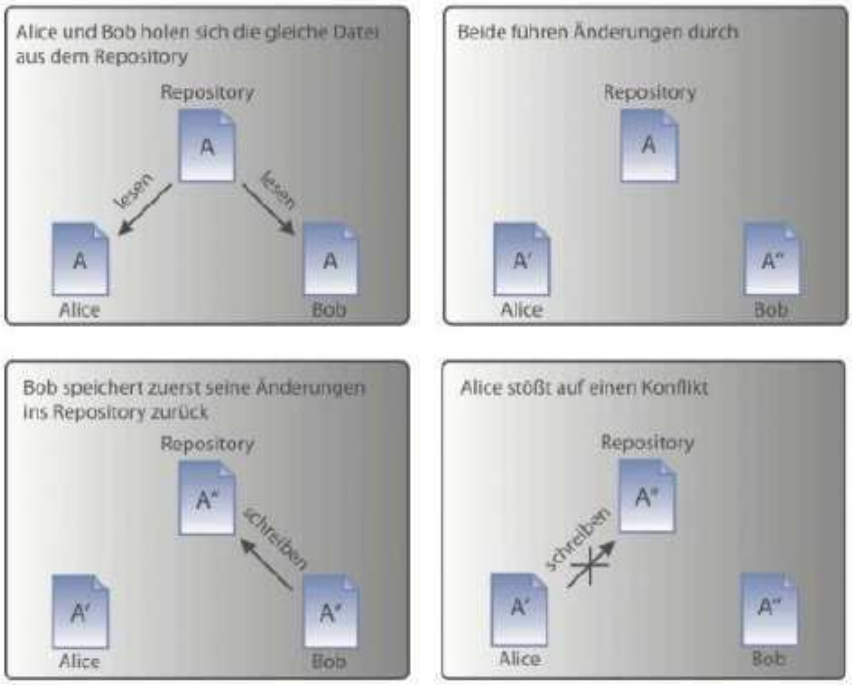
Typische Stakeholder:
Kunde/Auftraggeber/Sponsor
Geschäftsführung/Abteilungsleiter (Auftraggeber- und Auftragnehmerseite)
Promotor
Legacy Owner (Besitzer des Altsystems oder der Altdaten)
Projektleiter
Projektsteuergruppe

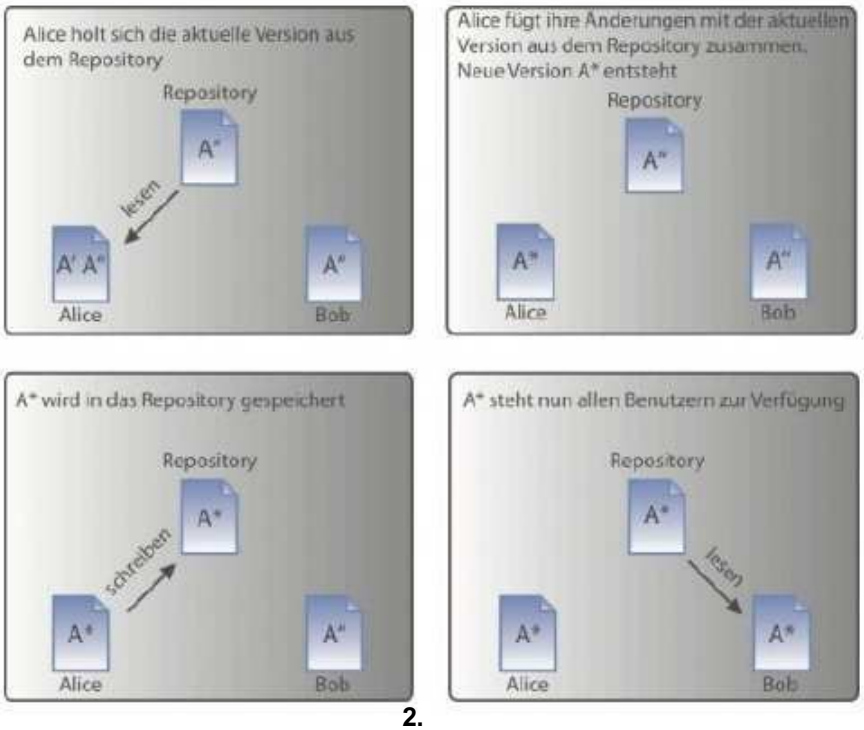
Erklären sie den Unterschied zw. Optimistic Revision Control und Pessimistic Revision Control. Erklären sie diese Strategien anhand Skizzen.

Pessimistic Revision Control (Lock-Modify Unlock Strategie):



Optimistic Revision Control (Copy-Modify-Merge Strategie):





2.

Was versteht man unter einer "Baseline" in der Analyse? Was passiert üblicherweise bei Änderungen (Changes) nachdem eine Baseline gesetzt wurde?

Es existiert niemals ein echtes Einfrieren (Freeze) der Anfoderungen Trotzdem sind stabile Zustände in der Analyse unabdingbar .

- Baseline

• „A specification or system that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development and can be changed only through formal change control procedures.“ – IEEE-Standard 610.12 (1990)

Nach dem setzen einer Baseline wird <u>überlicherweise bei Änderungen folgendes durchgeführt</u>

• Change Impact Analysis
• Re-Costing

Versioncontroler / Versionmanagement
Unterschied zw. Zentrale, Dezentrale, Lokale Versionmanagement

„Versionsmanagementsysteme sind Computersysteme, die Änderungen an Dateien und Verzeichnissen über die Zeit hinweg archivieren und in einem oder mehreren Entwicklungszweigen für die gemeinsame Bearbeitung durch mehrere Personen bereitstellen“

Wichtige Funktionen
Verzweigung / Branching:

- Unter Branching versteht man das Anlegen einer Kopie von Objekten im Versionsmanagementsystem

Markieren / Tagging:

- Ein Tag kennzeichnet einen konkreten Entwicklungsstand
  Beispiel: eine zur Auslieferung bestimmte Version

Protokollierung / Historisierung:

- Protokolliert welche Änderungen von welchem Benutzer zu welcher Zeit vorgenommen wurden
- Ermöglicht es jeden beliebigen Versionsstand aus der Vergangenheit wiederherzustellen

Unterschiede
Zentrales: Zentraler Server - Beliebig viele Clients (Beispiele: CVS – Concurrent Versions System, SVN - Subversion, Perforce)

Dezentrales: Kein zentrales Repository. Jeder Entwickler besitzt sein eigenes Repository – Änderungen werden zwischen den Repositories ausgetauscht. (Beispiele: Git, Bazaar)

Lokales: Älteste Form von Versionsverwaltung. Beschränkt sich auf die Versionierung einzelner Dateien. Nicht geeignet für Team-Projekte. (Beispiele: RCS, SCCS)

Softwaremetriken

Softwaremetriken ermöglichen, die Qualität von Software zu interpretieren

- Bei Prozessmetriken handelt es sich um quantitative Daten des Softwareentwicklungsprozesses
- Produktmetriken messen die Software oder Teile davon, ohne dabei zu berücksichtigen, wie das Produkt entstanden ist oder warum es sich gerade in einem bestimmten Zustand befindet

Mit dem Ansatz der objektorientierten Programmierung wurden neue Metriken notwendig, welche auch die Beziehungen von Objekten untereinander und deren Struktur berücksichtigten

Objektorientierte Metriken:

- CBO (coupling between objects)
- DIT (depth of inheritance tree)
- NOC (number of children)
- RFC (response for a class)
- WMC (weighted methods per class)
- LCOM (lack of cohesion in methods)

Spiralmodel nach Böhm

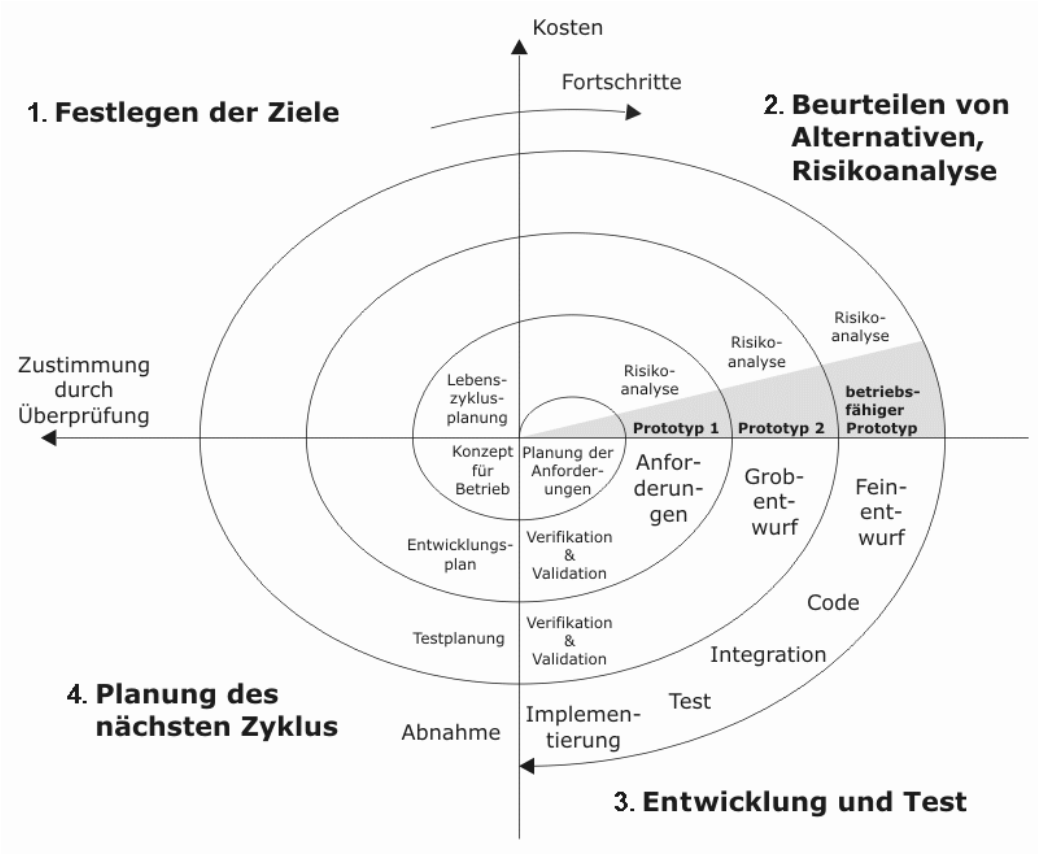
Spiralmodel nach Böhm berücksichtigt die möglichen Projektrisiken der sequentiellen Entwicklung und ist daher iterativ aufgebaut.

Der gesamte Prozess ist in vier Phasen gegliedert.
Alle Phasen werden im Rahmen einer evolutionären Softwareentwicklung mehrmals durchlaufen

Phasen:

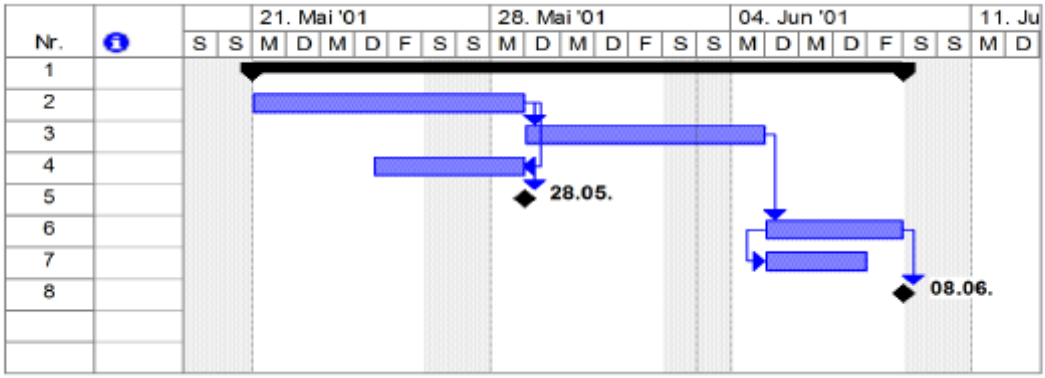
1. Definition von Zielen
2. Risikoanalyse
3. Durchführen der Arbeitsschritte
4. Planen der nächsten Phase

Die Anzahl der notwendigen Durchläufe ergibt sich erst im Projektverlauf und ist von den der auftretenden Risiken abhängig.



GANTT-Diagram (Balkendiagram)

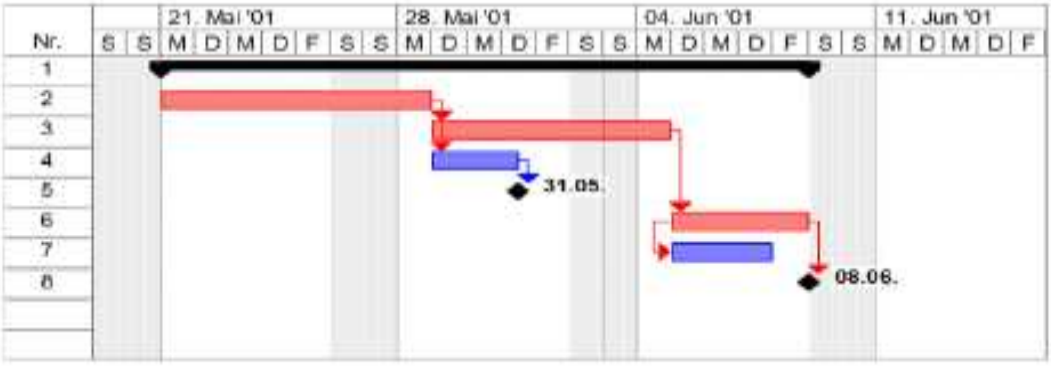
GANTT ist der älteste und am weitesten verbreitete grafische Methode
Aufgaben werden zur Terminplanung dargestellt (Aufgaben oder Personenbezogen)



Pufferzeit eines Vorgangs

- Gesamtpuffer: Zeitspanne, um den sich ein Vorgang verzögern darf, ohne dass das Projektende verzögert wird
- Freier Puffer: Zeitspanne, um die sich ein Vorgang verzögern darf, ohne dass andere Vorgänge verzögert werden

Kritischer Vorgang: Vorgang ohne Pufferzeit
Kritischer Pfad: enthält nur kritische Vorgänge



Was sind Integrationstests? Erklären sie unterschiedliche Integrationsformen!

Review-Prozess. Warum wichtig? Erklären sie den Ablauf, und die Rollen.

Was sind Buildmanagementsysteme? Erklärung und 3 Beispiele.

Ein Buildmanagementsystem ist ein Werkzeug um immer wiederkehrende Aufgaben zum Bauen einer Applikation automatisiert ablaufen zu lassen.

Stellt Abhängigkeiten zwischen Komponenten sicher.

Gängige Buildmanagementsysteme:

- Make
- Ant, nAnt
- Maven
- Buildr

Was ist der Unterschied zwischen Internationalisierung und Lokalisierung?

Internationalisierung (i18n): „sämtliche Maßnahmen, eine Software so zu gestalten, dass diese möglichst einfach an andere Sprachen und Kulturen angepasst werden kann“.

Lokalisierbarkeit (i10n): Wie gut lässt sich ein Produkt übersetzen, ohne in den Source Code eingreifen zu müssen?

Internationalisierung – Tipps:

- Kein „hard-coden“ von Nachrichten und Beschriftungen
- Schriftgrößen nicht einschränken
- Nachrichten nicht „stückeln“
- Identische Nachrichten in unterschiedlichem Kontext mehrfach speichern
- Gleich bei Implementierungsbeginn vorsehen – nachträgliche Umsetzung sehr teuer
- Extrahieren in Ressource Files (Alle Teile, mit denen der Benutzer agiert; Kommentierung; Sinnvolle Gruppierung)

Lokalisierung (i10n):

- Der „eigentliche“ Übersetzungsprozess
- Anpassung von lokalen Datenformatierungen (Zahlen, Datum, Währung,...)
- Hinzufügen von spezifischen Code
- Für jede Sprache/Kultur separat notwendig
- Vollständige Lokalisierung sehr aufwändig (Leserichtung, ...)

Äquivalenzklassenanalyse & Grenzwertanalyse erklären und Unterschiede

Äquivalenzklassenanalyse

- Einteilung möglicher Ein- und Ausgabewerte in Klassen gleichem Systemverhalten
- Es werden Werte von Klassen gewählt, bei denen angenommen wird, dass Fehler auftreten, die auch bei allen anderen möglichen Werten der Klasse auftreten können

Grenzwertanalyse

- Spezialfall der Äquivalenzklassenanalyse
- Tatsache, dass Fehler besonders oft an den Grenzen der Äquivalenzklassen auftreten

(näheres: Folie - „SEPM VO5 Integration Testen“)

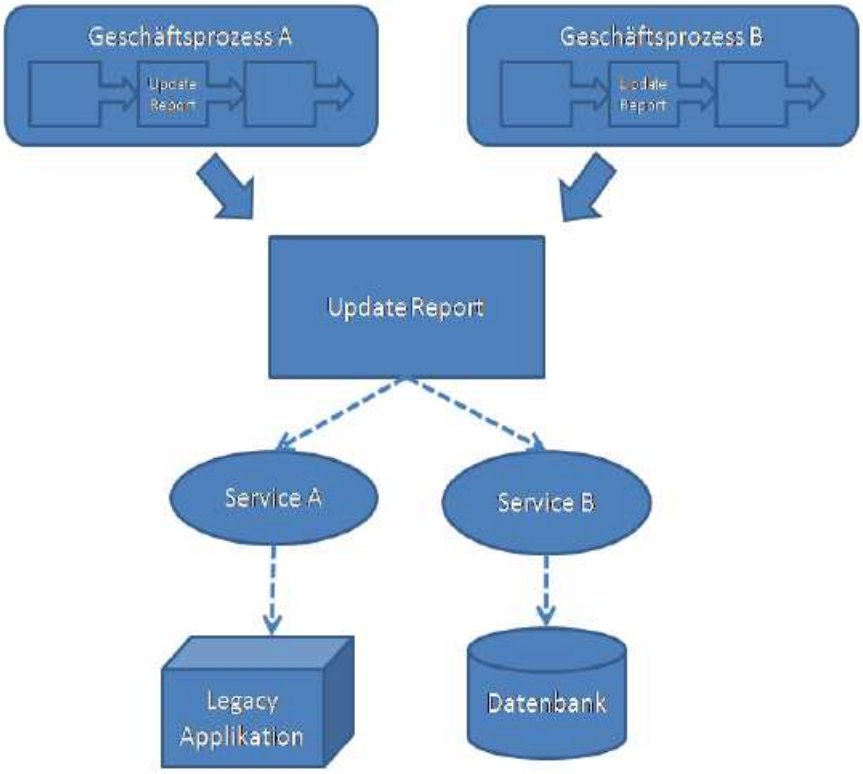
Was ist BPM (Business Process Modelling)? Zusammenhang SOA und BPM?

Im Service-orientierten Entwurf wird ein Schritt in einem Geschäftsprozess extrahiert und gekapselt → dabei handelt es sich um ein sog. Business Service

BPM und SOA sind eng miteinander verknüpft – Anwendungen werden maßgeblich von der Geschäftsprozess-Modellierung beeinflusst.

Zusammenhang von BPM und einer SOA:

- Geschäftsprozess A und B greifen auf das gleiche Business Service „Update Report“ zu
- „Update Report“ wird in Form von zwei technischen Services gelöst



~~Blackbox und Whitebox Frameworks. Unterschied? Wie werden diese spezialisiert?~~ Erklären von "Method Hooks" und "Hot Spots"

Method Hooks

- Abstrakte Methoden, die vom Entwickler überschrieben werden um das gewünschte Verhalten zu erstellen

Hot Spots

- „Punkte“, an denen ein Framework spezialisiert wird

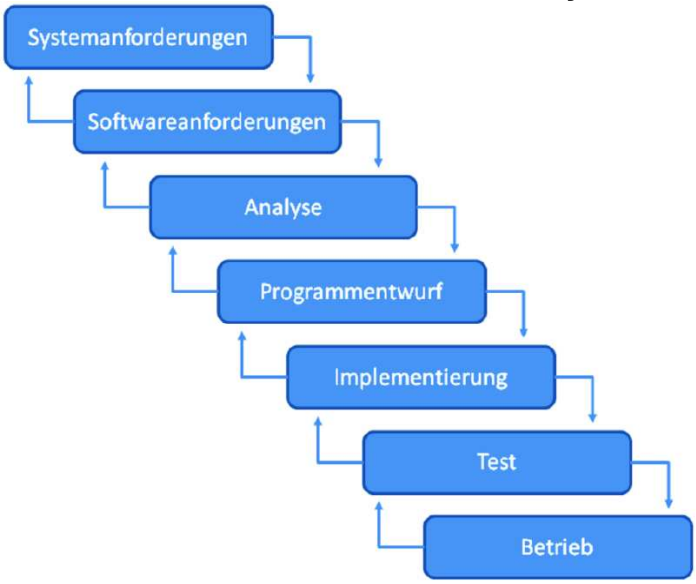
Was sind Vorgehensmodelle und wozu sind sie gut?

Ziel ist es, die Softwareentwicklung auf eine etablierte, ingenieurmäßige Basis zu stellen, um Verbesserungen in der Produktivität, Qualität und Vorhersagbarkeit zu erreichen.
Es existiert kein optimaler Entwicklungsprozess, eine Art Universalprozess, für jedes Softwareprojekt. Je nach Projektart ist ein geeignetes Vorgehen auszuwählen

Wasserfallmodell nach Royce 1970

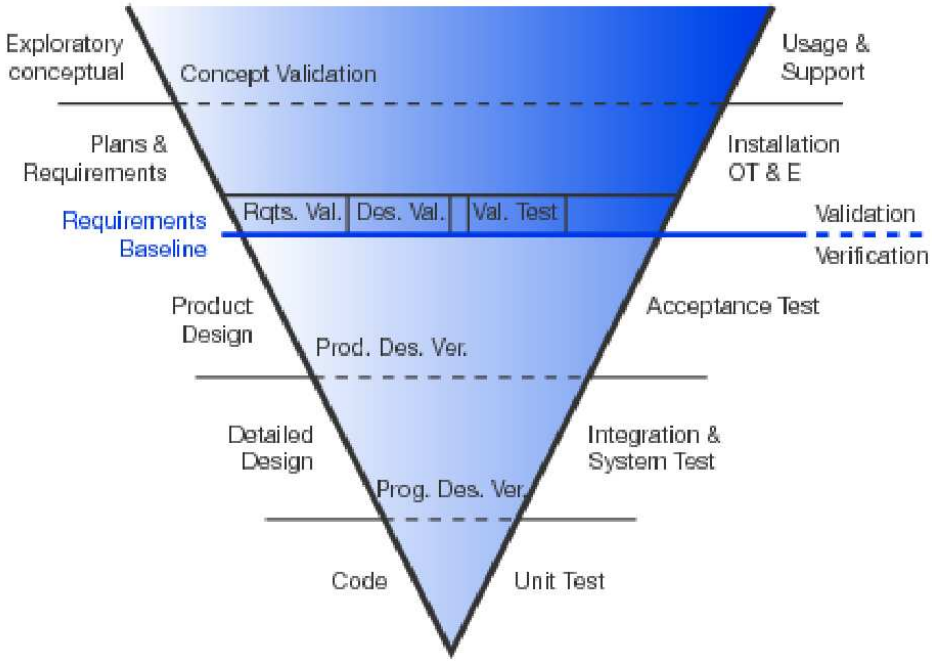
- Das am weitesten verbreitete Vorgehensmodell, von dem es viele unterschiedliche Varianten und Ausprägungen gibt.
- Modell in seiner Grundform vollständig sequentiell.
- Lässt in einer Erweiterungsstufe einen Rückwärtsschritt von einer Phase auf den direkten Vorgänger zu.
- Ein Arbeitsschritt kann erst dann abgeschlossen werden, wenn alle vorgesehenen Produkte fertig gestellt wurden.
- Berücksichtigt wesentliche Dokumente und Review-Schritte im Modell.
- Wird in der modernen Softwareentwicklung als inhärentes Risiko betrachtet und hat zur Entwicklung der iterativen Modelle geführt.

Fundamentales Wasserfallmodell nach Royce



V-Modell nach Böhm

Basis für viele aktuelle Entwicklungsmodelle, die insbesondere im öffentlichen Sektor zum Einsatz kommen (V-Model 97, V-Model XT, BVM). Das V-Modell ist auf folgenden Eigenschaften aufgebaut: Korrelierende Phasen; Validierung und Verifikation

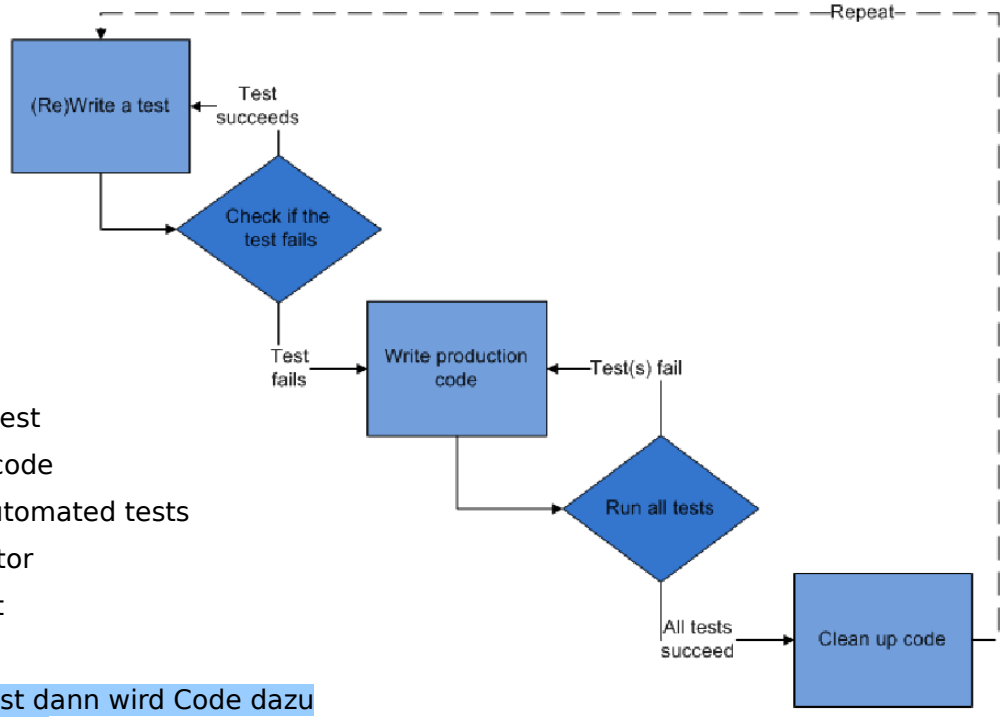


Erklären Sie das Test Driven Development anhand einer Skizze. Welchen Bezug zu automatisierten Tests zu TDD gibt es?

Definition

„TDD is a style of development where

- You maintain an exhaustive suite of programmer tests,
- No code goes into production unless it has associated tests,
- You write the test first,
- The tests determine what code you need to write“. (Astels 2003)



1. Write test
2. Write code
3. Run automated tests
4. Refractor
5. Repeat

Zuerst Test dann wird Code dazu geschrieben.

Fundamentale Prozessmodelle

Grundlegenden Idee des Phasenmodells
Phasen dieses Modells sind in unterschiedlicher Form in jedem Prozessmodell enthalten