Software Engineering I

6. Implementierung

Prof. Dr. Eckhard Kruse

DHBW Mannheim

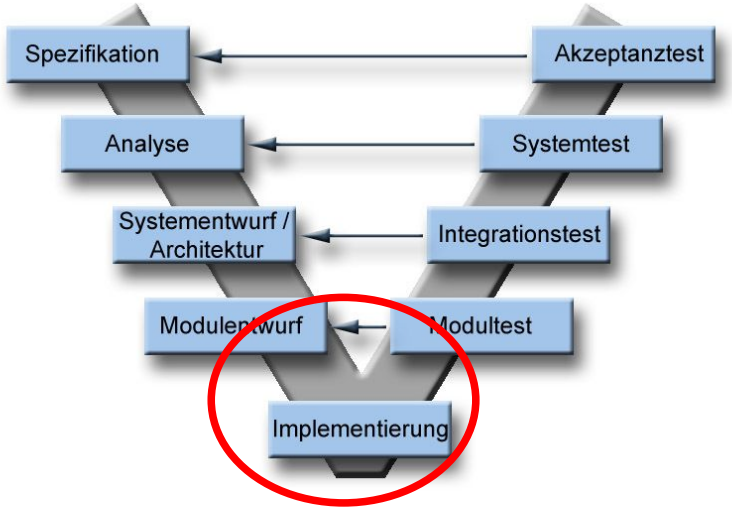
Implementierung

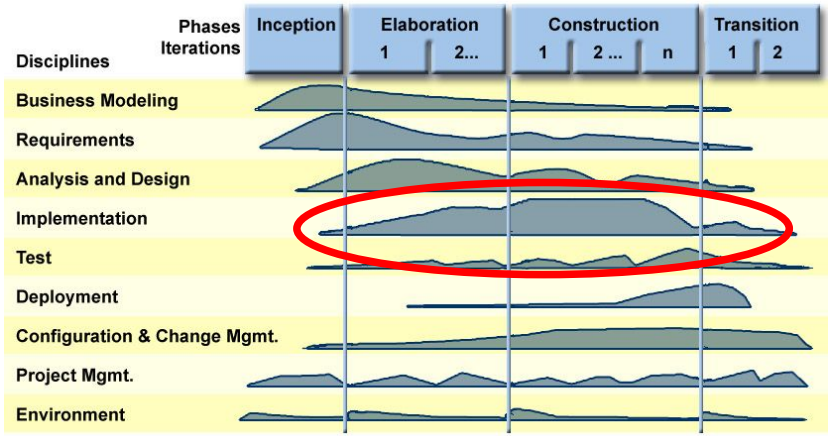
Definition

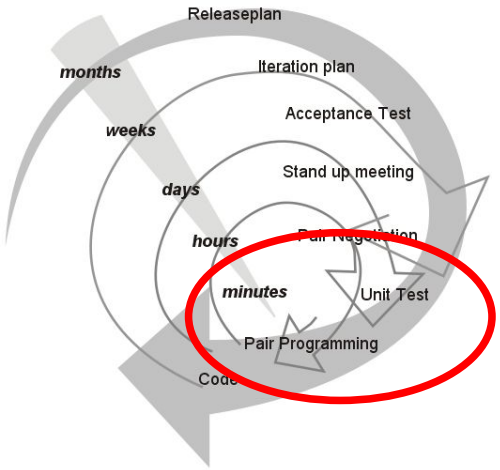
Die **Implementierung (implementation)** ist die Umsetzung des Entwurfs durch Programmierung (Codierung) der einzelnen Komponenten / Module.

- Zur Implementierung gehört auch die Durchführung von grundlegenden Tests auf Modulebene sowie die Dokumentation des Quellcodes.
- Während der Implementierung können noch Entwurfsentscheidungen innerhalb der Module erforderlich sein. Die Schnittstellen / Verträge zwischen den Komponenten sollten aber möglichst nicht mehr verändert werden.
- **Ergebnisse:** Dokumentierter Quellcode, kompilierte Komponenten/Systemteile, Testprotokolle.

Implementierung in den Vorgehensmodellen







Implementierung: Nur Programmieren?

Programmierer:

- Verständnis der Systemarchitektur + Entwurfsrichtlinien
- Feinentwurf: Wie werden die einzelnen Module realisiert
- Dokumentation
- Programmierung: Codieren auf Modulebene
- Einhalten der Programmierrichtlinien
- Kommunikation, Abstimmung mit dem Team
- Code Reviews
- Debugging
- Modultests (Unit tests): Schreiben von Testcode, Durchführen der Tests, ggf. Dokumentation in Testprotokollen
- Konfigurationsmanagement und Build-Prozess: Einpflegen des eigenen Codes, Befolgen der Prozesse
- Bug Tracker und Change Requests: Einpflegen von Fehlern, Auswerten/Korrigieren von Fehlern, die den eigenen Aufgabenbereich betreffen
- Refactoring

Architekt:

- Beratung, Steuerung
- Unterstützung Feinentwurf
- Änderungsbedarf?

UI Designer:

- GUI Entwurf
- Usability
- Grafiken, HTML, CSS ...

Konfig.manager:

- Build (daily? weekly?)
- Überwachung CMS
- Prozesse, Fristen

Change control board

- Bug tracker, change request

Qualitätsmanager:

- Abnahme der Tests
- Codierrichtlinien
- Prozesse

Was ist guter Code?

Wodurch zeichnet sich qualitativ hochwertiger Code aus?
Worauf sollten Programmierer achten?

Was ist guter Code?

- Zielgerichtet und effizient
  - dient der Umsetzung der Spezifikation (funktional und nicht-funktional)
  - folgt dem Entwurf
  - ist dabei so einfach wie möglich
  - und korrekt
- Verständlich
  - verwendet Kommentare + ggf. externe Dokumentation
  - sinnvolle Variablen- und Methodennamen
  - hält Stilkonventionen ein
  - verwendet ggf. Standardvorgehen (Patterns)
- Sicher und zuverlässig
  - Ausnahmezustände (Exceptions) werden getestet/abgefangen
  - Code ist gut (automatisiert) testbar (z.B. Testinterfaces)

Coding Conventions

Definition

Coding Conventions schreiben einen einheitlichen Stil bei der Programmierung vor, um so die Verständlichkeit des Codes zu verbessern und die Softwarequalität (vor allem Wartbarkeit, Korrektheit, Zuverlässigkeit) zu steigern.

Allgemeine Vorschriften

- Formatierung (Einrücken, Klammerung usw.)
- Namenskonventionen (Variablen, Funktionen)
- Dateisystemstruktur und Dateinamen
- Überprüfen und Behandeln von Ausnahmen (Exceptions)
- Sicherheit (z.B. nur sichere Funktionen verwenden → s. Buffer overflow)
- Lokalisierbarkeit (Native language support)

Anwendungsspezifische Vorschriften:

- Zu verwendende Schnittstellen, verbotene Schnittstellen
- Empfohlene Entwurfsmuster (Patterns)
- ...

Coding Conventions

Übung

6.1 Coding Conventions

a) Recherchieren Sie nach Coding Conventions and zugehörigen Tools.

b) Wie beurteilen Sie die Einführung von Coding Conventions, wenn diese nicht durch Analysewerkzeuge automatisch überprüft werden?

Konfigurationsmanagement

Definition

Unter **(Software-)Konfigurationsmanagement (software configuration management)** versteht man die Verwaltung der in der Softwareentwicklung entstehenden Produkte wie Source Code, Konfigurationsdateien, Dokumente usw. unter Berücksichtigung des definierten Entwicklungsprozesses and verwendeter Werkzeuge.

- Versionierung, Labeling (und ggf. Konfliktbehandlung) aller Artefakte
- Definition und Verfolgung von Prozessen
- Zugriffskontrolle, verschiedene Zugriffsrechte
- Protokollierung aller Vorgänge
- Integration mit anderen Werkzeugen, z.B. mit der Entwicklungsumgebung und der System-Build-Umgebung.

Code Review

Definition

Bei einem **Code Review** wird ein Teil des Quellcodes während oder nach der Entwicklung von/mit einem Gutachter (typischerweise ein anderer Entwickler → Peer Review) manuell durchgegangen, um Fehler zu finden bzw. die Qualität des Codes sicherzustellen.

- Vorteile: Bessere Code-Qualität, gegenseitiges Lernen
- Nachteile/Probleme: Aufwand, erfordert Offenheit der Entwickler
- Typischerweise nur für zentrale, wichtige Systemkomponenten
- Begriffe: Code Review, Peer Review, Code Walkthrough
- Kriterien: Nachvollziehbarer Entwurf, Einhalten von Coding Standards, Kommentierung, Verständlichkeit des Codes usw.
- ggf. gezielte (dokumentierte) Maßnahme im Qualitätsmanagement, z.B. als Ergänzung zum Testen

Lieferant: Code Review

Software-Engineering-Projekt

P.15 Lieferant: Code Review

Führen Sie einen Code Review durch, indem Sie ausgewählte Teile Ihres Codes präsentieren und erläutern.

a) Welche Code-Teile bieten sich für einen Code Review an (= versprechen Nutzen bzgl. Qualitätsverbesserung)?

b) Bereiten Sie sich auf den Review vor. Was ist die Aufgabe des Codes? In welchem Zusammenhang ist seine Funktion zu verstehen? Worin bestehen die Schwierigkeiten und die gewählten Lösungsansätze?

c) Führen Sie den Review durch.

d) Arbeiten Sie Änderungen/Verbessungsvorschläge in den Code ein.

System Build

Definition

Mit dem **(System) Build (Prozess)** bezeichnet man das Kompilieren und Linken aller am System beteiligten Softwarekomponenten (sowohl aus Eigenentwicklung als auch 3rd party Bibliotheken) zu einem ausführbaren System.

- Routineaufgabe (z.B. täglich oder wöchentlich), um möglich frühzeitig und regelmäßig ein testbares Gesamtsystem zur Verfügung zu haben.
- Sollte daher weitestgehend automatisiert ablaufen.
- Für große Systeme potenziell schwieriger, aufwändiger Prozess (z.B. → speziell verantwortlich Mitarbeiter als 'Build Manager')
- Hunderte bis tausende Dateien betroffen.
- Sorgfalt! Jeder einzelne Mitarbeiter kann durch seine Module den Build-Prozess scheitern lassen.
- hilfreich: lokale (teilweise) Buildumgebung für die beteiligten SW-Entwickler.

Don‘t break the build!

How to recognize a good programmer?

http://www.inter-sections.net/2007/11/13/how-to-recognise-a-good-programmer/
How do you recognise good programmers if you are a business guy?
(Assumption: The CV does not help much)

Positive Indicators:

- Passionate about technology
- Programs as a hobby
- Will talk your ear off on a technical subject if encouraged
- Significant (and often numerous) personal side-projects over the years
- Learns new technologies on his/her own
- Opinionated about which technologies are better for various usages
- Very uncomfortable about the idea of working with a technology he doesn't believe to be "right"
- Clearly smart, can have great conversations on a variety of topics
- Started programming long before university/work
- Has some hidden "icebergs", large personal projects under the CV radar
- Knowledge of a large variety of unrelated technologies (may not be on CV)

How to recognize a good programmer?

http://www.inter-sections.net/2007/11/13/how-to-recognise-a-good-programmer/
How do you recognise good programmers if you're a business guy?
(Assumption: The CV does not help much)

Negative Indicators:

- Programming is a day job
- Don't really want to "talk shop", even when encouraged to
- Learns new technologies in company-sponsored courses
- Happy to work with whatever technology you've picked, "all technologies are good"
- Doesn't seem too smart
- Started programming at university
- All programming experience is on the CV
- Focused mainly on one or two technology stacks (e.g. everything to do with developing a java application), with no experience outside of it

Lieferant: Alpha-Version

Software-Engineering-Projekt

P.16 Lieferant: Alpha-Version

Eine erste lauffähige (aber noch nicht funktional vollständige) Version Ihrer Software ist fertig. Präsentieren Sie diese Alpha-Version dem Kunden.

a) Ist der Kunde zufrieden? Ist das Projekt auf dem richtigen Weg?

b) Aktualisieren Sie ggf. Ihre Projektplanung und Ihr Risikomanagement.