

Diplomprüfung Vertiefung Softwarekonstruktion

Gebiete: SWT, OOSK, SQS, PM, PLE

Prüfer: Prof. Lichter

Datum: 30.06.2005

Dauer: 47min

Note: 1,3

SWT

Was ist denn Softwaretechnik, oder Software Engineering?

Wurde auf der Nato Summer School 1968 als Provokation geprägt, dass Softwareentwickler ingenieurmäßig vorgehen sollen.

Wieso denn als Provokation? Was war der Hintergrund?

Softwarekrise. Viele Projekte abgebrochen, weil kein richtiges Verständnis von einem Entwicklungsprozess bzw. Prozessmodellen.

Was gibt es denn für Prozessmodelle?

Wasserfall als Standardmodell, und verbesserte wie RUP, Spiralmodell, V-Modell. Und XP als agiles Prozessmodell.

Warum heißt das Wasserfallmodell denn Wasserfallmodell?

Weil man da die einzelnen Stufen runter fließt und wenn überhaupt erst sehr spätes Feedback bekommt.

Nennen Sie die Stufen mal.

Anforderungsanalyse, bla bla.

Sie nannten einige „weiter entwickelt“ als das Wasserfallmodell. Was ist an dem Modell schlecht? Oder wie versucht man, das Modell zu verbessern?

Zum einen gibt es Ansätze mit Prototyping, zum anderen mit mehreren Iterationen, indem man den Wasserfall mehrmals durchmacht, und versucht, das Feedback möglichst direkt in die einzelnen Stufen zu integrieren.

Dann zeigen sie mal RUP.

RUP besteht aus 4 Phasen und 9 Workflows. Die Phasen sind...blabla und mal auf. Ich musste die Aufwandsverteilungen nicht einmalen, aber die Iterationen.

OK. Was ist denn Requirements Engineering?

Der Prozess, die Wünsche aus dem Kopf des Kunden zu modellieren und zwar möglichst so, dass man daraus eine Architektur ableiten kann. Man eruiert, analysiert, strukturiert und modelliert dann die Wünsche und muss mehrmals mit dem Kunden kommunizieren, um auch sicher zu gehen, dass beide dasselbe meinen. Dazu versucht man die Sprache recht „formal“ und eindeutig zu formulieren, wozu es einige Regeln gibt.

Welche Notationen hat man denn im RE?

Vor allem Use Cases / Diagramme.

Und was noch?

Structured-Analysis, ERD.

Wann würde man das denn benutzen?

Bei Datenmodellierung, bzw. Datenbanken.

Was modelliert man denn mit Use Cases?

Die funktionalen Anforderungen.

Und was sind nichtfunktionale Anforderungen?

Zum Beispiel Zeitverhalten und Design.

Das sind die Restriktionen.

OOSK

Da haben wir ja die Architektur... Was sind denn Schichten?

Ööööh (das stand doch gar nicht wirklich in den Folien?) Hm ja da sind Klassen, Bibliotheken und Frameworks drin.

Ja aber was ist das?

-ratlos-

Wie stehen die denn miteinander in Beziehung?

Die können auf einander zugreifen.

Welche genau?

Nur die benachbarten.

Na die untere kann auf die nächst höhere zugreifen.

Und was ist wenn man über mehrere Schichten will?

Dann muss man das durchreichen.

Ja. (Puh war das ätzend)

Was sind denn Entwurfsprinzipien?

Meinen Sie sowas wie Open-Closed-Principle?

Ja.

Dann Open-Closed, Maximale Kohäsion-Minimale Kopplung, Datenkapselung, Information Hiding.

Was ist denn Open-closed?

Das ist offen gegenüber Erweiterungen und geschlossen gegenüber Änderungen.

Und wie realisiert man das?

Ööööh (Shit was fragt er alles für'n Mist). Das hab ich auf den Folien gar nicht erklärt gefunden.

Überlegen Sie mal. Wann ist eine Klasse offen für Erweiterung aber abgeschlossen über die Schnittstelle.

Öhm bei Vererbung, also Subclassing.

Ja genau.

Ah so einfach ist das. :)

Und da gabs noch die Trennung von Interaktion und Funktion.

Was kennen sie da für Beispiele?

MVC. Bla bla und aufgemalt.

Warum macht man das?

Damit man die View und das Model getrennt voneinander weiter entwickeln kann, die quasi entkoppelt hat.

Die Entwurfsmuster wollen ja so was wie größtmögliche Flexibilität erreichen. An welchem Beispiel würden Sie das erklären wollen?

Hmm da würd ich mal den Observer nehmen.

Wo sehen Sie da die Flexibilität?

Naja ich kann die Beobachter an- und abmelden, und den Beobachteten getrennt davon bearbeiten.

Nagut. Kommen wir zu Rahmenwerken. Was ist denn das?

Das sind Makroarchitekturen. Im Gegensatz zu Klassenbibliotheken nicht nur eine Sammlung von Klassen, sondern die Klassen in einem Design eingefügt. Man muss das Rahmenwerk dann mit anwendungsspezifischem Code vervollständigen.

Welche Arten gibt es da?

Blackbox und Whitebox.

Was ist denn an den Blackbox-Rahmenwerken Blackbox?

Man kennt die innere Struktur nicht und muss sie durch Parametrisierung spezialisieren, am besten Werkzeugunterstützt.

Genau, und die Whitebox-Frameworks werden dann irgendwie anders genutzt.

SQS

Was ist denn Qualität?

Die Gesamtheit aller Merkmale...

Und das mal in andern Worten?

Qualität muss vorher definiert werden. Damit man sich was unter den Qualitäten vorstellen kann, hat Barry Boehm 1976 einen Qualitätenbaum entwickelt bla bla. Das sind auch ungefähr die, die man in der DIN ISO 9126 findet.

Welche Verfahren kennt man denn in der Qualitätssicherung?

Analytische und konstruktive. Konstruktive sind fehlervermeidend und werden beim Entwicklungsprozess eingesetzt. Bei den analytischen Maßnahmen testet man.

Sagen wir besser: Prüfen. Testen ist eine Form der dynamischen Prüfung. Was gibt es da noch?

Reviews.

Und noch?

Für mich gibt's nur die beiden.

Es gibt noch rechnergestützte.

Nagut.

Bei Blackbox, woher kriegt man die Testfälle?

Die werden aus der Spezifikation abgeleitet.

Dann sagen sie mal welche Blackbox-Testverfahren es gibt.

Ein/Ausgabeüberdeckung, Funktionsüberdeckung, Error Guessing, UWG.

Und bei Whitebox?

C0, C1, C2, Termüberdeckung, bla bla.

Und wie sucht man Testfälle aus?

Testfälle müssen repräsentativ, ökonomisch, fehlersensitiv sein, und ähm...überleg

Nach ein paar Sekunden: OK geschenkt! Aber wie suche ich die genau aus?

Mit Äquivalenzklassenbildung beispielsweise, oder Grenzwertanalyse. (Schieß aber nicht das zu sein was er hören wollte.)

Wann hör ich denn auf mit Testen?

Wenn der Test keine Fehler mehr findet, oder jeder weitere Test ökonomisch kein Sinn mehr macht. Also wenn ich pro Testfall über einer ökonomischen Grenze liege.

Aber ein Testfall kostet doch nichts mehr, wenn er einmal hergestellt ist.

Achso, ja ich meinte jeden neu gefundenen Fehler, nicht Testfall.

Gut. Warum ist eine Metrik ein Modell?

Na das ist sehr abstrakt, weil man nur auf eine Zahl abbildet und nicht auf etwas, was man sich sonst als Modell vorstellt. Dabei wird ein Objekt zuerst in ein empirisches Relationssystem abgebildet, und von da aus in ein numerisches. Die Abbildung bildet sowohl die Entitäten ab als auch die Relationen.

Wenn Ihnen jemand eine Metrik gibt, was brauchen sie dann um sie zu benutzen?

Hm ja die Abbildung halt. Und den Wertbereich, also die Skala.

Naja aber können Sie dann damit was anfangen?

Achso man braucht noch eine Interpretation dazu.

Ja genau.

PM

Was ist ein Projekt?

Eine einmalige Tätigkeit, in vorgegebener Zeit mit den vorgegebenen Mitteln ein vorgegebenes Ziel zu erreichen.

Und was sind Aufgaben im PM?

Planen, Teamführung, blabla.

Was plant man genau?

Planungsgrößen sind Kosten, Ressourcen und Leistungen.

Dann noch CM- wofür braucht man das?

Verwaltung von Dokumenten. Man will ja auch Nachverfolgen können, welche Versionen zusammenpassen, welche Revisionen oder Alternativen zu den Dokumenten es gibt.

Nur Dokumente?

Hmmm ja Code vielleicht noch. Aber Code ist doch auch Dokument.

Die ganze Softwareentwicklung erzeugt doch nur Dokumente.

Ja richtig.

(War also ne Veräppelfrage!)

Und das eigentliche haben sie jetzt noch nicht genannt.

Ja die Konfigurationen. Und wenn man das an den Kunden ausliefert, nennt man es Release.

Wir hatten Risikoanalyse.

Da muss man die Risiken zuerst identifizieren, indem man erfahrene Entwickler befragt, oder Fragebögen und Checklisten verwendet. Danach analysieren, indem man den Summen im Schadensfall und Eintrittswahrscheinlichkeiten Werte von 1 bis 3 zuordnet und als deren Produkt den Risikofaktor bestimmt. Für hohe Risiken muss man präventive Maßnahmen definieren, für mittlere Risiken Notfallmaßnahmen und weniger wichtige Risiken erst mal gar nichts. Das ganze ist ein ständiger Prozess und muss immer wiederholt werden.

PLE

Was ist der große Unterschied zur normalen Softwareentwicklung?

Dass man 2 Prozessmodelle braucht, eins für Domain Engineering und eins für das spezifische Produkt. Diese sind unabhängig voneinander, aber durch das Management gekoppelt.

Wie sieht denn so eine Domain Analysis aus?

Man muss ein Scoping durchführen, wobei Gemeinsamkeiten und Variabilitäten festgelegt werden. Mit FODA kann dies dann modelliert werden. Die Variabilitäten können in UML nicht ausgedrückt werden, und dieses Problem zieht sich durch alle weiteren Schritte hindurch.

Ok noch eine letzte Frage.

Die weiß ich nicht mehr.

Ich hatte nicht den Eindruck, dass Prof. Lichter irgendwelche Standard-Fragen rausgekramt hatte. Ich war an einigen Stellen unangenehm überrascht, dass er nach Sachen fragte, die ich auf den Folien irgendwo mal in der Ecke entdeckt hatte, wie Open-Closed-Principle und die Architekturlayer. Oder hatte ich zu alte Folien? Das sollte einen in der Prüfung aber nicht unbedingt verunsichern. Einige Sachen lernt man halt auch erst in der Prüfung :)

Zu den Vorlesungsfolien muss ich sagen, dass ich selten besseres Material gesehen habe. Die Folien sind quasi selbsterklärend, gut gegliedert und ansprechend aufgemacht. Auch der Inhalt der Vorlesungen kommt in meinem Fall dem nahe, was ich von den Vorlesungen erwarte. Im Gegensatz dazu ist das Skript von Prof. Nagl zu „Einführung in die Softwaretechnik“ ein Gräuelp. Ich habe selten so ein zerhacktes unformatiertes unverständliches unansprechendes out-of-date Skriptgekleisterte gesehen. Eigentlich schon unverschämt. Lest einfach ein bisschen in dem Buch von Sommerville zu SWT. Ich kann also nur jeden ermutigen, der nur teilweise weiß, welcher Stoff ihn erwartet. Einzig bei PLE (Folien 2003) muss man sich durch Kapitel „RE“, „architecture“ und „implementation“ etwas durchbeißen, was aber im 2. Anlauf durchaus zu schaffen ist.