

Prüfungsprotokoll (Gedächtnisprotokoll)

Vertiefungsprüfung (Software Technik), Prof. Rumpe, 08.02.2010

Themen

- Einführung in die Software-Technik (V3, Rumpe)
- Modellorientierte Software-Technik + zusätzlichen Inhalt beider Bücher (V3, Rumpe)
- Objektorientierte Softwarekonstruktion (V3, Lichter)
- Software-Qualitätssicherung (V3, Lichter)

Ablauf

Alle Fragen und Antworten sind sinngemäß wiedergegeben und auch nur in so weit wie ich mich erinnere. Er fragte womit wir anfangen sollten und ich antwortete er könnte mich gerne quer durch fragen, was er dann auch getan hat. Das war auch seine erste Diplomprüfung an der RWTH.

- Was fällt Ihnen denn zum Thema Muster ein?
Wir hatten in den Vorlesungen Analysemuster, Architekturmuster, einige Nicht-GoF Entwurfsmuster und einige GoF Entwurfsmuster und im zweiten Teil des Buches wurden noch einige Testmuster behandelt.
- Was halten Sie denn für die wichtigsten Muster?
Mein Lieblingsmuster ist der Interceptor, weitere wichtige Muster sind die Factory, der Iterator und je dann je nach Anwendung, für Frameworks z.B. Template oder Strategy.
- Beschreiben Sie mal die Factory.
Statische Objekterzeugung erklärt, Nachteile davon und Einsatzzwecke der Factory (Decorator, Proxy, Test Mocks).
- Sie hatten den Interceptor erwähnt, bitte.
Erklärt wozu er dient (dynamische Erweiterung von Frameworks) und das UML Klassendiagramm aufgezeichnet und erklärt. Außerdem habe ich noch die Anwendung im asp.NET Framework erläutert.
- Dass für jede Methode wäre ja ganz schön aufwendig, wann ist dann denn sinnvoll?
Da ich der Meinung bin, dass das fast immer sinnvoll ist, meinte ich für alle Kernmethoden und allgemein sollte ein Framework auch eine schmale externe Schnittstelle haben. War nicht das was er hören wollte.
- Gehen wir mal zu den Analysemustern, welche kennen Sie?
Abstrakte Oberklasse mit Spezialisierung Party, Koordinator, Rolle und Kompositum. (Das Werkzeug / Material Muster habe ich bewusst weggelassen, doch...)
- Erklären Sie mal das Rolle Muster.
Beispiel DB wird als Log System und für die Änderung von Konfigurationsdateien benutzt, als einmal nur Hinzufügen von Sätzen ohne Auslesen und einmal dann Lesen und Ändern. UML Diagramme für Schnittstellenvererbung und für Delegation erklärt.
- Was ist der Vorteil von Delegation?
Ich habe geantwortet, dass allgemein Delegation den Vorteil hat, dass man die Objekte dynamisch austauschen kann. Er wollte hören, dass man Rollen hinzufügen und entfernen kann, sagte aber das sei was Ähnliches. Obwohl ich dachte, dass dem auch nicht ganz so ist, hielt ich es für besser nichts zu sagen. Der Protokollführer kam eh schon nicht mit, weil ich so schnell redete.

- In der Vorlesung von Herrn Lichter hatten wir das Werkzeug Material Muster, was ist das?
Ich habe zuerst erklärt was das Muster bedeutet mit dem Beispiel Papier, Bleistift und Spitzer. Dann habe ich erklärt wie es technisch mit Aspekt-Klassen realisiert wird und welche Vorteile das bezüglich der Kopplung hat.
- Wo kann man das gut nutzen?
In der Vorlesung war wohl ein Beispiel dazu, ich habe mich nicht mehr erinnert und sagte ich wüsste nicht dass wir das hatten. Habe auf visuelle Editoren getippt, war nicht das was er hören wollte. Er hat dann angenommen, dass nicht mehr in der Vorlesung enthalten war. Ein Blick auf meine Folien gerade sagt mir aber, dass das so nicht ganz stimmte.
- Kompletter Cut. Er fragte dann nach Prozessen bzw. nachdem ich die Frage genauer haben wollte nach Vorgehensmodellen.
Wasserfall als Grundmodell, das erstmals Phasen einteilt, Weiterentwicklung Richtung Test zu V-Modell, dann iterative und inkrementelle Entwicklung als Richtlinien für Vorgehen, dann evolutionäre Entwicklung, RUP, XP und modellorientierte Softwareentwicklung.
- Nehmen wir mal den RUP, was ist das?
Fokus auf UC, Prototyping und Meilensteine. Unterschied zu Wasserfallmodell, dass die klassischen Phasen Workflows sind. Dann habe ich die einzelnen Phasen alle detailliert erklärt mit Inhalt, Anteil der Workflows und vor allem die Resultate der jeweiligen Phase.
- Wie ist das mit Qualitätsmanagement?
Das hatte ich vergessen und halt dazu dann noch erklärt, dass QM, Konfigurations sowie Projekt- und Prozessmanagement in allen Phasen vorhanden sind.
- Thema QM: was gibt es denn für QM Maßnahmen?
Analytisch, Fehler finden – nachgefragt nach Beispielen – Test und Review, konstruktiv, Fehler vermeiden, Beispiele Code Guidelines oder z.B. Codegenerierung (habe gehofft ihn drauf lenken zu können, hat aber nicht geklappt) und organisatorische Maßnahmen, planen und so.
- Dann kam eine Frage mit der ich nichts anfangen konnte: was gibt es denn für analytische Maßnahmen außer Test und Reviews? Modellchecking fand er zu Highlevel, Walkthrough wollte er auch nicht hören und dann meinte er Metrik.
Dann habe ich erklärt was eine Metrik ist (Modellbildung und so) und habe dann erklärt wieso ich finde, dass eine Metrik sich nicht zur Fehlerfindung eignen und auf den GQM Ansatz verwiesen.
- Welche Metrik könnte man denn nehmen um Fehler zu finden?
Ich habe dann Komplexitätsmaß genannt, was aber nicht echt Aussagekräftig ist und objektorientierte Maße für Vererbungstiefe und interne Kopplung.
- Wie misst man die Kopplung denn?
Ich habe erklärt was die helfen (was er gar nicht wissen wollte wie er mir danach sagte) und dass man sie halt durch relative Anzahl messen kann.
- Was ist denn GQM?
Erklärt wieso meist unnützes Zeug gemessen wird und die einzelnen Schritte von GQM erklärt.
- Was könnte man denn versuchen um wirklich Fehler zu finden?
Prüfen auf Testbarkeit.

- Wie ginge das?
Eigene Fehler einbauen und schauen wie viele gefunden wurden, dann noch Hinweis zu Testendekriterien.
- Das wollte er gar nicht wissen, er korrigierte seine Frage was im Test denn falsch sein könnte. Probleme und dann auch Lösungen (Testmuster aus dem zweiten Buch) erklärt: Statische Methoden und Attribute durch Singleton / Factory Kombi, Seiteneffekte in Konstruktoren durch init(), Nebenläufigkeit aufgrund des Indeterminismus der Reihenfolge mit ActiveObject Muster und das Problem von Zeit durch Nutzung von eigenen Zeitobjekten, die man durch Mocks austauscht. Ich brauchte auch nichts im Detail erklären, der Protokollführer kam wieder mal nicht mit gerade glaube ich und das reichte ihm dann auch dem bezüglich.
- Was ist denn CMMI?
Zertifizierungsmaß für Prozessqualität. Dann habe ich angefangen die einzelnen Stufen zu erklären. Ab Stufe 4 hat er nicht mehr zugehört. *hehe*
- Was ist denn die verbreitete Stufe?
Ich habe 2.5 bis 3 geantwortet und dann erklärt wieso die meisten Firmen gar nicht mehr wollen und für welche Bereiche Level 4 und 5 Zertifizierungen benötigt werden.
- Was sind denn so typische Level 3 Bereiche?
Hab gesagt weiß ich nicht, würde aber mal vermuten so Wirtschaftsbereiche und Geschäftslogik-Anwendungen. War glaube ich auch nicht was er hören wollte, nahm er aber so hin.
- Was wären typische Level 3 Maßnahmen, die auf Level noch nicht vorhanden sind?
Leider wusste ich nur die paar Level 5 Maßnahmen, die ich von der Wiki mir gemerkt habe. Vom Standpunkt „variierende Qualität“ habe ich dann mal Reviews geantwortet.
- Was prüft man denn mit Reviews?
Sicherheitskritische Dokumente, als erstes Beispiel kam mir dann in den Kopf „wenn man einen neuen Mitarbeiter hat“, dann haben sie gelacht. Na ja, dann habe ich noch einige Bereiche aufgezählt, wo man auch andere Dokumente akribisch testen muss wie Luftverkehr.
- Wo würden Sie Verifikation da einordnen?
Level 4. Mehr konnte ich nicht sagen, dann unterbrach er mich und erzählte wann es in Level 3 sinnvoll sein könnte, aber das in der Regel Level 4 der Normalfall sein sollte.
- Dann fragte er mich nach irgendeinem Gesetz, weiß nicht wie es hieß. Ich habe gesagt habe ich noch nie gehört und gefragt ob er mir einen Tipp geben könnte. Er hat mir was erzählt und ich habe nur wiederholt: habe ich noch nie gehört.
- Dann fragte er was man über mehrere Projekte denn so als Firma gewinnen könnte.
Ich habe den Bausteingewinn als Prozessqualitätsmerkmal. Einmal Know-How Gewinn der Mitarbeiter und dann wiederverwendbare Komponenten: Klassenbibliotheken, im nächsten Schritt Frameworks und dann Produktlinien.
- Was sind denn Produktlinien?
Ich habe erklärt was eine Produktlinie ist (gemeinsame Plattform, individuelle Produkte, gemeinsame Weiterentwicklung), welche Vorteile die haben (Usability, allgemeine Vorteile von Wiederverwendung) und dass es aus Domain- und Applikations Engineering besteht. Dann habe ich noch erklärt was man im Domain-Engineering in den einzelnen Phasen (Analyse, Entwurf, Implementierung und Test) jeweils macht, was die Ergebnisse davon sind und das reichte ihm dann endgültig und er hat mich rausgeschickt. Dauerte aber nicht sehr lange bis ich wieder rein durfte.

Dauer der Prüfung insgesamt: 30 Minuten

Auf meine Frage, ob ich zu schnell und zu viel geredet hätte, antworteten sowohl Prof. Rumpe als auch der Protokollführer gleichzeitig mit ja.

Note: 1.0

Ich war gut vorbereitet und konnte das in der Prüfung auch gut zeigen. Atmosphäre war super – wobei das Gefühl hat man bei Monologen meistens und man wird auch nicht auf irgendwas festgenagelt, was man nicht weiß. Auch halbkorrekte Antworten werden akzeptiert.

Fazit: Prof. Rumpe ist als Prüfer absolut empfehlenswert!

Lernmaterial

- Vorlesungsfolien „Einführung in die Softwaretechnik“ (Stand WS 2009 / 2010)
- Vorlesungsfolien Objektorientierte Softwarekonstruktion (Stand WS 2007 / 2008)
- Vorlesungsfolien Software-Qualitätssicherung (Stand SS 2009)
- Buch: Modellierung mit UML (Rumpe)
- Buch: Agile Modellierung mit UML (Rumpe)
- Buch: Entwurfsmuster in C#
- Buch: Ergänzungen bei Unklarheiten mit dem englischen „Design Pattern“ von Gamma
- Buch: UML Glasklar 2
- Eigene Mitschriften aus den Vorlesungen Einführung in die Softwaretechnik, OOSK und Modellorientierte SW-Entwicklung

Abschließender Hinweis: ich empfehle die Vorlesungen von Prof. Rumpe über alle Maße. Zu JEDER hingehen. Man kriegt dort einfach ein gutes Gefühl was wirklich wichtig ist, außerdem kann er Stoff lebhaft vortragen.