Internet-Learning
Systementwicklung

Marco Rademacher

2. Mai 2000

Zusammenfassung

In dieser Vorlesung sollen die Grundlagen hinsichtlich der Planung eines Softwareentwicklungsprozesses gelegt werden. Dazu werden verschiedene Prozessmodelle vorgestellt. Am Ende sollen die Teams selbst festlegen, wie sie ihren Softwareentwicklungsprozess gestalten wollen, und ihn mit der Semesterplanung verknüpfen. http://www.inf.fu-berlin.de/inst/ag-bg/lv/il/index.html

1 Prozessmodelle

Software ist heutzutage so komplex, dass sie in den seltensten Fällen einfach programmiert werden kann. Es sind vorüberlegungen anzustellen, damit die Software das leistet, was der Auftraggeber wünscht, und ihre Funktionfähigkeit sicherzustellen.

1.1 Quick & Dirty

Das populärste Entwicklungsmodell bei unprofessionellen Programmierern ist der Quick & Dirty (Q & D) Ansatz: Die Software wird »mal eben schnell« programmiert, wenn man sich über den Zweck und den Weg der Programmierug im Klaren ist. Dazu wird in zwei Phasen vorgegangen:

1. Schreibe ein Programm
2. Finde und behebe die Fehler in deinem Programm

Sah sich ein Programmierer schon längere Zeit mit einem Problem konfrontiert, dann hat er sich schon lange gedanklich damit befasst und es ist tatsächlich möglich, dass er nun mit dem Programmieren gute Software schreibt. Die Realität des Informatikers in seinem Berufsleben unterscheidet sich jedoch von dieser Situation, wenn er *für andere* Software schreiben soll. Hier kennt er weder das Problem genau, das die Software lösen soll, noch hat er in ausreichendem Maße Lösungsansätze herausgearbeitet.

Spätestens bei der Erweiterung des Programms hätte man sich vielleicht etwas mehr konzeptuelle Weitsicht gewünscht, wenn entscheidende Teile des Programms neu geschrieben werden müssen oder man lästige Altlasten mit sich herumtragen muss.[1]

[1] Es gab einmal ein kleines Betriebssystem, das auf diese Weise geschrieben wurde: QDOS. Irgendwann wurde es von einem genialen Geschäftsmann aufgekauft, lediglich etwas verändert und einem IBM-PC beigelegt, womit es dann seinen Siegeszug rund um die Welt antrat. Und seitdem ärgern sich die meisten Benutzer mit unnötigen Altlasten herum... Welches Betriebssystem ist hier wohl gemeint?

1.2 Das Wasserfallmodell

Das oben genannte Modell wird nun ausgebaut: vor der eigentlichen Programmierung findet eine Planungsphase statt, in der der Rahmen des Projekts geplant wird. Dann wird im Rahmen der Definitionsphase der Ist-Zustand analysert und die Ziele festgelegt. Während der Entwurfsphase werden die Ziele bis zu Konzepten heruntergebrochen und das neue System modellhaft entworfen, bis der Entwurf in den Code umgesetzt wird (programmieren). Schließlich wird das Programm getestet und in Betrieb genommen.

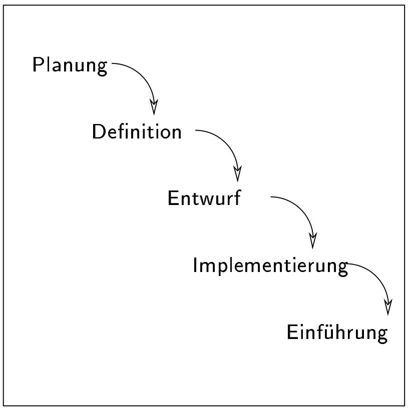

Abbildung 1: Das Wasserfallmodell

Diese Phasen werden der Reihe nach durchgeführt und es wird erst zur nächsten Phase gegangen, wenn eine Phase gänzlich abgeschlossen wurde. Stellt man einen Fehler fest, der in einer vorherigen Phase aufgetreten ist, so wiederholt man den Prozess von jener Phase an.

Dieses Modell bildet die Grundlage vieler anderer Vorgehensmodelle, da es logisch aufgebaut und durch seine Einfachheit klar und verständlich ist. Nachteil dieses Verfahrens ist, dass der Kunde lediglich an der Produktdefinition mitwirkt. Erst bei der Einführung erfährt er die Ergebnisse seines Auftrags und ist entweder zufrieden, oder eben nicht. Wenn nicht, kann es sein, dass der Entwicklung dermaßen daneben lag, dass gänzlich von vorne begonnen werden muss.

1.3 Das V-Modell

Das V-Modell erweitert das Wasserfallmodell dadurch, dass zahlreiche Verifikationen und Validierungen eingebaut werden: ausgehend von der fertigen Implementierung wird wieder zurückgangen bis zu einem Abnahmetest beim Kunden.

1.4 Das Prototypen-Modell

Neben dem »normalen« (Wasserfall-) Projektablauf ist für das Prototypen-Modell charakteristisch, dass dem Kunden häufig Anschauungsmaterial präsentiert wird, aufgrund dessen er sich für den Auftragnehmer entscheiden, die bisherige Projektarbeit absegnen oder seine Wünsche zur Software präzisieren kann.

Welche Art Prototyp erstellt wird, hängt vom Zweck bzw. der gerade bearbeiteten Projektphase zusammen:[2]

Demonstrationsprototyp: Dieser Prototyp dient zur Auftragsakquisation und soll dem Auftraggeber lediglich einen Eindruck vermitteln, was ein Produkt leisten könnte. Solche Prototypen dienen allein der Vorführung und werden nach der Demonstration weggeworfen.

Prototyp im engeren Sinne ist ein Prototyp, an dem spezielle Aspekte praktisch getestet werden sollen.

[2] Vgl. [1] S. 115

Labormuster sind Prototypen, die den Entwicklern ein Beispiel für eine bestimmte Funktionalität geben sollen. Im Rahmen der Vorlesung dieser Lehrveranstaltung werdet Ihr (hoffentlich) viele »Labormuster« erhalten, um ein Anschauungsmodell über bestimmte Funktionalitäten zu bekommen.

Pilotsystem: ist ein Prototyp, der nicht nur der experimentellen Beobachtung dient, sonder auch ein Teil bzw. der Kern des Endprodukts ist.

1.5 Das evolutionäre Modell

Das evolutionäre Modell sieht vor, dass es sich beim ersten Produkt noch nicht um das eigentliche Endprodukt handeln soll. Das Ziel ist, so früh wie möglich den Kern des Programms zu implementieren und dann fehlende Teile in erneuten Produktionszyklen zu entwickeln. Damit können die Erfahrungen und Wünsche des Kunden und der Benutzer in spätere Programmversionen einfließen.

Durch den kleineren Rahmen ist es sehr viel schneller möglich, dem Kunden etwas auszuliefern. Außerdem kann man durch Rückkopplung von den Anwendern erfahren, in welche Richtung das Produkt weiterentwickelt werden soll. Damit lässt sich vermeiden, dass der Auftragnehmer komplett am Kunden vorbeientwicklet und auf seinem Produkt sitzenbleibt. Auch im Bereich des »anonymen« Kunden wird häufig auf diese Weise entwickelt.

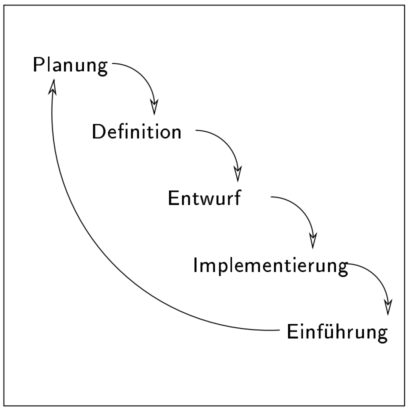

Abbildung 2: Das evolutionäre Modell

1.6 Das inkrementelle Modell

Während das evolutionäre Modell auch den Analysebereich auf das Gebiet eingeschränkt, das implementiert wird, analysiert das inkrementelle Modell den Problembereich vollständig, entwickelt dann aber das Produkt in Aufbaustufen.

So soll gewährleistet sein, dass man vorhandene Teile des Programms nicht etwa neu schreiben muss, wenn man bei einem erneuten Durchlauf der Definitionsphase andere Modelle wählt. Durch den ausführlicheren Anfang des Projekts kommt man aber auch nicht so schnell zu einem ersten Protoyp.

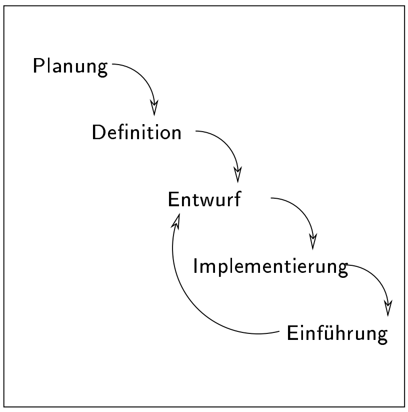

Abbildung 3: Das inkrementelle Modell

1.7 Das objektorientierte Modell

Die bisherigen Prozessmodelle verwendeten alle einen »top down«-Ansatz, d.h. sie entwarfen das Programm nach den Zielen, nicht nach den Möglichkeiten. Charakteristich für das objektorientierte Modell ist die Wiederverwendung bereits bestehender Programmteile (Klassen, Objekte). Daher wird hier eine »bottom up«-Komponente ins Modell aufgenommen. Das Design der Software richtet sich also auch nach bereits vorhandenen Programmteilen. Das objektorientierte Modell wird häufig in Verbindung mit evolutionären Ansätzen kombiniert.

1.8 Das nebenläufige Modell

Das nebenläufige Modell sieht vor, dass ein Programm stufenweise aufgebaut wird, die einzelnen Stufen aber parallel in Teams aufgebaut werden.

1.9 Das Spiralmodell

Das Spiralmodell gibt nur einen Rahmen vor, welche Fragen zu stellen sind, um ein geeignetes Prozessmodell zu entwerfen, zu hinterfragen und abzuändern. Da es somit einen hohen Managementaufwand bedeutet, eignet es sich nicht für kleine bis mittlere Anwendungen und Teams und soll hier nicht weiter ausgeführt werden.[3]

1.10 Abschließende Betrachtung

Welches ist nun das wahre Modell? Wie bei allen Modellen gibt es keine Wahrheit, sondern nur Zweckmäßigkeit. Welches Modell gewählt werden soll, hängt von hohem Masse von der Eigenart des Projekts, dem Auftraggeber und auch dem Entwickler selber (und seiner Flexibilität) ab. Es gibt einfache Modelle, die nicht viel mit der Realität zu tun haben, aber sehr verständlich sind, sodass man gut damit arbeiten kann. Und es gibt komplexe Modelle, die besser auf die Realität guter Softwareentwicklung passen, aber so schwer verständlich sind, dass man kaum mehr damit arbeiten kann. Hier ist also von jedem Entwicklungsteam und dem Kunden eine Entscheidung zu treffen, wie vorgegangen werden soll, und das Prozessmodell dem jeweiligen Projekt anzupassen.

[3] Mehr in [1].

2 Zusammenarbeit mit dem Kunden

»Auftraggeber kommen vom Mars, Auftragnehmer von der Venus«[4]

Während sich der Dienstleister gerne in sein Spezialgebiet vertieft (schnelle Programme, originelles Design, spektakuläre Ideen, usw.) hat der Kunde wahrscheinlich ein gänzlich anderes Interesse: die Software ist für ihn nicht einfach Spielerei, sondern kann über den Erhalt einer Firma entscheiden, indem mit der Software der Umsatz gesteigert werden soll. Auch in unserem Anwendungsgebiet, also im Schulbereich, hat der Kunde ein professionelles Interesse. Er will Software für einen bestimmten Einsatzzweck, nicht für Spielereien und Beschäftigung einiger Programmierer. Er erwartet, dass der Dienstleister sein Problem löst. Nicht mehr und nicht weniger.

2.1 Das Selbstverständnis der Dienstleistung

Für den Dienstleister heißt das, sich selbst möglichst zurückzuhalten. Gute, innovative Ideen sind natürlich mehr als erlaubt, insbesondere wenn der Kunde noch nicht genau weiß, was er will. Hier kann der Sachverstand der Informatiker auf der Seite der Software ein Anlass sein, seine Ideen und Kenntnisse einzubringen, um ein sinnvolles Produkt zu schaffen. Man sollte sich aber im Klaren darüber sein, dass niemand so leicht über seinen eigenen Schatten springen kann. Daher ist eine Qualitätskontrolle, also eine Person, ein Gremium oder ähnliches, das in Regelmäßigkeit über den Fortgang des Produkts entscheidet, eine sinnvolle Einrichtung, um nahe an den Zielen und dem Zeitrahmen des Kunden zu bleiben und sich nicht zu verzetteln.

2.2 Auftraggeber und Anwender

Der Kunde besteht meist aus einem Entscheidungsträger der Führungsebene einer Firma. Ab einem bestimmten Punkt ist er bestrebt, die eigentliche Arbeit zu deligieren. Sodann wird mit dem Mittelbau kommuniziert, der sich oft dadurch auszeichnet, mehr Sachkenntnis zu haben, aber über weniger Entscheidungskompetenz zu verfügen. Ferner sind Mitarbeiter aus dem Mittelbau scheinbar noch nicht am Ende ihrer Karriere: bei ihrer persönlichen »Politik« spielen deshalb auch persönliche Karrierepläne eine Rolle. Ob sich der Dienstleister darauf einlässt, diese Eitelkeiten zu befriedigen oder nicht, hängt sicherlich von jedem einzelnen ab. Entscheident ist aber, das persönliche Wunschdenken der Einzelnen zu erkennen und von den Bedürfnissen des Auftraggebers und seinen Zielen zu trennen.

Oft wird bei der Systementwicklung aber die entscheidende Ebene unterschlagen, nämlich die Anwender. Der normale Angestellte arbeitet nicht so sehr für das Firmenwohl, wie für das eigene Auskommen. Ihn interessiert die Umsatzsteigerung nicht so sehr wie der Erhalt seines Arbeitsplatzes. Solange der Computereinsatz nicht nur mit Steigerung der Produktivität, sondern auch mit Entlassung einhergeht, muss sich der Systemanalytiker nicht wundern, wenn er bei den Anwendern auf Widerstände stößt. Solche Widerstände können einfach nur ärgerlich oder zeitraubend sein – sie können aber auch das Ergebnis der Analyse dahingehend verfälschen, dass die Entwicklung der Software auf gänzlich falschen Annahmen beruht. Insofern muss das Bestreben des Informatikers dahin gehen, Vertrauen zu schaffen, um Wahrheit zu erfahren.

[4] Aus: [2]

2.3 Wahrheit, Vertrauen, Transparenz

Um das richtige Softwareprodukt zu entwerfen müssen die Voraussetzungen richtig sein. Gemeinhin ist die Analyse bestehender Systeme eine Grundvoraussetzung vor dem Design eines neuen Systems. Hier ist der Systemanalytiker also auf wahre Angaben der Anwender angewiesen. Um nicht als Jobkiller oder einfach unliebsamer Besucher zu erscheinen, ist es wichtig, die eigenen Absichten so zu formulieren, dass die Anwender Vertrauen entwickeln können. Auch der Kunde (die Chefetage) ist daran interessiert, auf eine ordnungsgemäße Arbeit der Informatiker zu vertrauen. Dieses Vertrauen kann nur mit Kommunikation und ständiger Präsenz aufgebaut und gehalten werden. Was sich hier anbietet, ist (mal wieder) das Internet.

Auf einer Website kann der Dienstleister seinem Kunden darstellen, wie er das Projekt geplant hat und wie es tatsächlich bisher ablief. Mit solcher Dokumentation hilft er nicht nur seinem Kunden, den Blutdruck zu senken, er kann sich auch damit selbst entlasten, da der Kunde nicht so oft »sicherheitshalber« nachfragen muss. Ausserdem zwingt solche – natürlich ehrliche – Ablaufdokumentation den Informatiker, das Projekt gut zu planen und erfährt schneller, wenn etwas nicht so läuft, wie er es plante, und kann schon früh geeignte Maßnahmen ergreifen.

In großen Projekten oder großen Firmen wird solche Qualitätskontrolle auch oft ausgelagert, um hier verschiedene Aufgaben auch verschiedenen Rollen zuzuordnen, die sich dann voreinander rechtfertigen müssen.

Aufgaben

Die folgenden Aufgaben dienen zur Vorbereitung des ersten Treffens mit dem Kunden und den Anwendern. Die erste Aufgabe wird gleich im Plenum vorgestellt. Die anderen Aufgaben sollten dann bis Montag (2000-05-08) in der Gruppe bearbeitet werden. Nachzügler bearbeiten bitte auch die Aufgabe auf der Webseite http://www.inf.fu-berlin.de/inst/ag-bg/lv/il/il-ss00-aufgabe1.html

1. Überlegt Euch in, welches Prozessmodell Ihr für Eure Softwareentwicklung zugrunde legt, und wie es angepasst wird, damit es in den Zeitrahmen dieses Sommersemesters passt. Wann sind welche Phasen abgeschlossen? Mit welchen Ergebnissen (Meilensteine) werden sie abgeschlossen? Berücksichtigt eventuelle Feiertage, eigene Freizeit und »Stresszeiten« und die Ferien an der FU und der Schule.

2. Wann wären Euch nach Eurer Projektplanung weitere Treffen mit dem Kunden bzw. mit den Anwendern am liebsten? Was sollte bei solchen Treffen thematisiert werden?

3. Am nächsten Dienstag haben wir eine Exkursion zur Hildegard-Wegscheider-Oberschule in Wilmersdorf. Dort werden wir also einen ersten Kundenkontakt vornehmen. Mahlt Euch die Situation aus und überlegt, wie Ihr gegenüber dem Kunden und den Anwendern auftreten wollt (aktiv/reaktiv; Anrede Sie/Du; was erzählt Ihr bei Eurer Vorstellung; usw).

4. Das erste Treffen soll noch keine Verträge abschließen, sondern soll lediglich das Gebiet einer Zusammenarbeit ausloten und kann Euch auch schon dazu dienen, wichtige Fragen im Rahmen einer Ist-Analyse zu klären. Überlegt Euch, zu welchen Stichpunkten Ihr Fragen stellen wollt oder was Ihr selbst erforschen wollt.

5. Die Schüler (Anwender) werden eine Aufgabe gestellt bekommen, bei der sie mit dem Internet umgehen müssen. Ziel soll es sein, Euch einen Blick über die Schulter zu ermöglichen, um zu erfahren

   - wie geübt die Anwender mit dem Computer/Internet sind,
   - worauf es ihnen bei Internet-Diensten ankommt und
   - wie sie bestehende Internet-Dienste beurteilen.

   Einige Websites sollen zur Analyse der Anwender vorgegeben werden. Macht Euch mit den bisher vorgesehenen vertraut (www.cycosmos.de und www.das-berlin.de) oder bringt eigene Vorschläge ein!

6. Welche Hypothesen habt Ihr bzgl. der Nutzungsgewohnheiten der Schüler am Rechner/Internet?

7. Erstellt ein kurzes, übersichtliches Paper, das alle Angaben enthält, die Ihr am nächsten Dienstag zur Exkursion benötigt (Angaben zum Kunden wie Namen und Adresse, offene Fragen, eigene Vorstellungen/Ideen zum Projektfortgang).

8. Wie wollt Ihr mit dem Auftraggeber in Kontakt bleiben? Wollt Ihr Ihn über den Fortgang Eurer Arbeit weiterhin unterrichten? Auf welche Weise soll dies geschehen?

Literatur

[1] Helmut Balzert
Lehrbuch der Software-Technik
Spektrum Akademischer Verlag, Heidelberg, 1998

[2] David Siegel
Das Geheimnis erfolgreicher Websites – Business, Budget, Manpower, Lizenzen, Design
Markt und Technik Verlag, München, 1998

[3] Don Tapscott
Net Kids – Die digitale Generation erobert Wirtschaft und Gesellschaft
Gabler, Wiesbaden, 1998

In den USA ist die Netznutzung weiter fortgeschritten, als bei uns. Aus einer Analyse der Nutzungsgewohnheiten von Kindern und Jugendlichen wird gefolgert, welche Einstellungen und Erwartungen eine durch das Internet geprägte Generation hat und wie sie dadurch die Gesellschaft verändern wird.

Ein durchaus visionäres, aber leicht zu lesendes Buch, das etwas Licht auf die Anwenderseite wirft. Der Originaltitel lautet »Growing up Digital«. Außerdem gibt es eine Website zum Buch mit einigen Buchausschnitten und Hintergrundinfos:
http://www.growingupdigital.com