

Vorlesungsmodul Organisation von DV-Projekten - VorlMod OrgDvPrj -

Matthias Ansorg

11. Oktober 2004 bis 26. März 2005

Zusammenfassung

Studentische Mitschrift zur Vorlesung Theoretische Informatik bei Bernd Kloos (Wintersemester 2004/2005) im Studiengang Informatik an der Fachhochschule Gießen-Friedberg.

- **Bezugsquelle:** Die vorliegende studentische Mitschrift steht im Internet zum Download bereit. Quelle: Persönliche Homepage Matthias Ansorg :: InformatikDiplom <http://matthias.ansorgs.de/InformatikAusblgd/>.
- **Lizenz:** Diese studentische Mitschrift ist public domain, darf also ohne Einschränkungen oder Quellenangabe für jeden beliebigen Zweck benutzt werden, kommerziell und nicht-kommerziell; jedoch enthält sie keinerlei Garantien für Richtigkeit oder Eignung oder sonst irgendetwas, weder explizit noch implizit. Das Risiko der Nutzung dieser studentischen Mitschrift liegt allein beim Nutzer selbst. Einschränkend sind außerdem die Urheberrechte der angegebenen Quellen zu beachten.
- **Korrekturen und Feedback:** Fehler zur Verbesserung in zukünftigen Versionen, sonstige Verbesserungsvorschläge und Wünsche bitte dem Autor per e-mail mitteilen: Matthias Ansorg <<mailto:matthias@ansorgs.de>>.
- **Format:** Die vorliegende studentische Mitschrift wurde mit dem Programm LyX (graphisches Frontend zu L^AT_EX) unter Linux geschrieben und mit pdfL^AT_EX als pdf-Datei erstellt. Grafiken wurden mit dem Programm xfig unter Linux erstellt und als pdf-Dateien exportiert.
- **Dozent:** Bernd Kloos.
- **Verwendete Quellen:** <quelle> {<quelle>}.
- **Klausur:**

Inhaltsverzeichnis

1 Organisation	1
2 Was ist eigentlich Software?	2

1 Organisation

- Evtl. wird der Raum tw. in einen Computerraum gewechselt. Dort wird mit MS Project Projektplanung geübt (für weitere Aufgaben taugt dieses Programm nicht mehr) und mit einem weiteren Tool die Durchführung von Projekten.
- Die Folien vom Sommersemester 2003 befinden sich im Internet

- Voraussetzungen
 - Kenntnis über die Vorgehensweise bei der Softwareentwicklung
 - ggf. BPS
- Es wird eine Klausur am Semesterende geschrieben. Dauer: 90min. In der Klausur dürfen keine Unterlagen oder Hilfsmittel verwendet werden. Der Stoff steht größtenteils auf den 160 Folien.
- Von den 4 SWS gibt es 50-70% Vorlesung, 30-50% Übung, keine klare zeitliche Trennung
- Folien: <http://www.fh-giessen.de/fachbereich/mni/veranstaltungen.shtml> . Die Folien werden regelmäßig nach der Vorlesung aktualisiert. Zur Zeit liegen dort die Folien vom Sommersemester 2003 (Veranstaltung in Friedberg). Nur der Teil 2 ist bisher relevant.
- Man muss in der Veranstaltung nicht mitschreiben, es steht alles auf den Folien. Zeichnen in den Folien:
 - ! aktualisierte Seiten
 - × Entwurf
 - nicht überarbeitete Folie
- Zum Bestehen der Klausur reich:
 - Definitionen aus den Folien. Man muss die Folien nicht auswendig lernen, z.B. keine DIN-Normen usw..
 - Zuhören in der Vorlesung
 - gesunder Menschenverstand
- Die angegebene Literatur:
 - TomDeMarco: »Der Termin« (Roman). Eine gute Illustration der Praxis, sehr lehrreich.
- Es gibt keine Anwesenheitsliste, aber im dritten Vorlesungsblock eine Teilnehmerliste, um die Übungsgruppen bilden zu können.
- Es wird eine Projektsimulation im Praktikum geben.
- Anscheinend gibt es aber keine formalen Klausurvoraussetzungen.

2 Was ist eigentlich Software?

Ist Software ein Investitionsgut? Eine gekaufte Software zählt zum Anlagevermögen des Betriebes: das in der Bilanz, wofür die Firma ihr Geld ausgegeben hat, wogegen Umlaufvermögen das ist, womit das Unternehmen sein Geld verdient. Wenn eine Firma dagegen eine eigene EDV-Abteilung hat, konnte die Software nicht bilanziert werden und wurde dann auch so eingeschätzt: ihr wurde kein wirklicher Wert zugemessen. Das ist kein guter Umgang mit Software. Wo doch Software Wissen verkörpert: das Wissen eines Betriebes, das Wissen über seine Prozesse.

Die Zeit der großen, revolutionären Softwareentwicklungen ist einigermaßen vorbei: Abrechnungssysteme für Krankenkassen und Flugbuchungssysteme für Lufthansa muss nicht mehr geschrieben werden. Man traut sich nicht, die Software neu zu entwickeln: es kostete viel Erfahrung, all die Fehler

auszubauen. Deshalb sollte Software heute nicht anders behandelt werden als teure Maschinen: im Unternehmen selbst entwickelte Software kann deshalb heute auch auf der eigenen Bilanz erscheinen.

Dass Software wertvoll ist, ist der Grund warum sie auch ingenieurmäßig entwickelt werden muss. Das ist der Grund für diese Veranstaltung.

Schwierigkeiten bei der Softwareentwicklung:

- risikobehaftet
 - Es gibt manchmal technologische Grenzen, die Risiken darstellen: man weiß nicht, was man zu tun hat.
 - Krankheit von Entwicklern.
- nicht deterministisch. Weil Software innovativ ist: ich weiß, was ich will, aber nicht wie ich dorthin komme.
- interdisziplinär. Die Programmierer haben kein Fachwissen von der Anwendungswelt, müssen das auch nicht haben.
 - Software wurde zuerst im Militär, dann im betriebswirtschaftlichen Bereich entwickelt.
 - Es werden Personen benötigt, die zwischen Informatikern und Leuten aus der Anwendungswelt vermitteln können. Ein Grund, warum es heute z.B. Studiengänge wie Wirtschaftsinformatik gibt.
- hoher Abstraktions- und Komplexitätsgrad.
 - Abstraktion: Software ist immer nur ein Abbild der realen Welt.
- niedriger Normierungsgrad (Neuartigkeit und Einmaligkeit)
 - Es muss nur entwickelt werden, was es noch nicht gibt. Es wird stets etwas Neues entwickelt.
 - Vor der Entwicklung muss *durch Rückfragen* klar sein, was die Anwender tatsächlich brauchen. Der Kunde muss das Verständnis des Softwareentwicklers *bestätigen!*

Aufgrund dieser Schwierigkeiten wurde gefordert: Software muss innerhalb einer geeigneten Organisationsform geplant, gesteuert und kontrolliert werden. Verrichtungsorientierte Organisation von Unternehmen ist ungeeignet für Softwareprojekte: stattdessen werden Projektgruppen aus verschiedensten Verrichtungsbereichen kombiniert. Professionelle Softwareentwicklung bedeutet: nicht nur das Produkt als Ziel erreichen, sondern auch aufpassen, wieviel das kostet.

Repository: Datenbank mit den Konfigurationsparametern für ein System. Customizing: Einstellungen der Software machen, mit Auswirkung auf das Repository.

Unterschied zwischen Dienstleistungsvertrag und Werkvertrag:

Dienstleistungsvertrag Bezahlt nach Zeit; auch wenn das Sachziel nicht erreicht wird, muss der Kunde zahlen. Aber: heute gibt es auch Beraterhaftung.

Werkvertrag Bezahlt wird nur für das verlangte Werkstück.

Nach einem neulichen Urteil kann es sein, dass eine Softwarefirma für die Leistung, einen Schaden zu beheben kein Geld mehr bekommt. »Wenn sie einen Vertrag mit jemandem machen wollen, sehen sie zu, dass der Vertrag auf eine Seite passt.« Denn: Verträge sind nur nötig, wenn man sich »nicht mehr leiden kann«, und weil es schon fast beliebig viele Gesetze gibt, muss man nur festlegen, wer wann zahlen muss.

Warum gehen Projekte schief:

- fasche Planung
- keine klaren Anforderungen, niemand wusste was das Ziel sein soll
- Kompetenz
- Kompetenz des Projektleiters

»Projektleitung funktioniert, solange sich alle gern haben.« Der Projektleiter muss auf drei Dinge achten: Ergebnis, Termine, Kosten. Wichtige Regel: genau sobald man merkt, dass etwas nicht mehr funktionieren wird, mit dem Auftraggeber reden statt auf 120% Last zu arbeiten. Das ist ein Bestandteil des Projektmanagements. Das Team auszunutzen ist kein gutes Projektmanagement, sondern sein Team zu beschützen. Auch vor Aufgaben zu falschen Zeiten: nicht jeder Fehler muss sofort behoben werden.

Um Probleme nicht erst lösen zu müssen, wenn sie im Einsatz auftreten, hat die Automobilindustrie APQP (advanced product quality planning). Eine Software zu entwickeln, die gegen geltendes Recht verstößt, bedeutet, sich strafbar zu machen.

DIN-Normen sind keine Arbeitsanweisungen. Die Serie DIN 69 900 beschäftigt sich mit Projektmanagement.

Tagesgeschäft und Projektarbeit müssen in einer Firma (personell) getrennt werden. Projekte werden in Arbeitsaufträge unterstrukturiert. Die wichtigste Frage, die ein Projektleiter stets beantworten können muss: wo stehen wir mit dem Projekt? Eine falsche Entscheidung hat gegenüber keiner Entscheidung den Vorteil, zu erkennen welche Lösung falsch ist.

Literatur

[1] <titel>