



Report

Führung von Informatikprojekten durch Studentengruppen

Author(s):

Aebi, Daniel; Zehnder, Carl August

Publication Date:

1994

Permanent Link:

<https://doi.org/10.3929/ethz-a-001382244> →

Rights / License:

[In Copyright - Non-Commercial Use Permitted](#) →

This page was generated automatically upon download from the [ETH Zurich Research Collection](#). For more information please consult the [Terms of use](#).



Eidgenössische
Technische Hochschule
Zürich

Departement Informatik
Institut für
Informationssysteme

Daniel Aebi
Carl August Zehnder

**Führung von
Informatikprojekten durch
Studentengruppen**

November 1994

ETH Zürich
Departement Informatik
Institut für Informationssysteme
Prof. Dr. C.A. Zehnder
D. Aebi

Authors' addresses:

Institut für Informationssysteme, ETH Zentrum, CH-8092 Zurich, Switzerland
e-mail: aebi,zehnder@inf.ethz.ch

© 1994 Departement Informatik, ETH Zürich

Distribution: Departement Informatik, ETH Zentrum, CH-8092 Zurich, Switzerland

This report is also available in electronical form via anonymous ftp from
<ftp.inf.ethz.ch>

Führung von Informatikprojekten durch Studentengruppen

Zusammenfassung

Die Ausbildung in Informatik-Projektentwicklung bildet schon seit langem einen festen Bestandteil des Studienplanes der Abteilung für Informatik. Mit Beginn des Wintersemesters 1991/92 wurde dieses Ausbildungsangebot erweitert, indem zusätzlich die Projektarbeit in Gruppen gefördert und betreut wird. Der vorliegende Bericht befasst sich in vier Teilen mit verschiedenen Aspekten dieser Ausbildungsform: Im ersten Teil wird allgemein die Ausbildung im Bereich Informatik-Projektentwicklung an der Abteilung für Informatik beleuchtet. Der zweite Teil gibt eine detaillierte Uebersicht über die bisher gemachten praktischen Erfahrungen mit Gruppenarbeiten im Zeitraum 1991-1994. Der dritte Teil zeigt einige Grundregeln, welche Studierende für erfolgreiches Arbeiten in Gruppen kennen müssen. Der vierte Teil stellt ein Zeitplanungswerkzeug für Studenten vor.

Abstract

Training in computer project management has for a long time been part of the curriculum of the computer science department at ETH Zürich. Beginning in the winter semester 1991/92, this education offer was extended by promoting and supervising project work in teams. This report deals with various aspects of this education method: in the first part, training in the computer project management domain at the computer science department is shown in an overall view. The second part contains a detailed description of the practical experiences with teamwork during the years of 1991 to 1994. The third part illustrates some basic rules which students have to know in order to successfully work in teams. The fourth part introduces a time-planning tool for students.

Gliederung

Teil 1:	Projektführungsausbildung für Studierende <i>C. A. Zehnder</i>	4
Teil 2:	Angewandte Gruppenprojekte <i>D. Aebi</i>	7
Teil 3:	Zur Arbeit in Projektgruppen <i>C. A. Zehnder</i>	25
Teil 4:	Das Programm „Zeitplan“ <i>D. Aebi</i>	31

Teil 1

Projektführungsausbildung für Studierende

C. A. Zehnder

An der ETH Zürich werden seit Gründung der Abteilung für Informatik Informatik*ingenieurinnen und -ingenieure*¹ ausgebildet. Das weist bereits deutlich darauf hin, dass neben dem Fachwissen aus Informatik und einigen Naturwissenschaften (namentlich Mathematik, Physik, Elektrotechnik) auch methodische Aspekte im Curriculum ihren Platz haben müssen. In der Tradition der ETH Zürich wird aber das Methodische eng mit dem Fachlichen gekoppelt, wie dies beim Programmieren unter Verwendung moderner Programmierumgebungen besonders deutlich zum Ausdruck kommt. Auf diese Weise lässt sich das Hochschulstudium bis zum Diplomabschluss kurz halten (8 Studiensemester plus 1 Semester für die Diplomarbeit); erst die praktische Berufstätigkeit vermittelt den Absolventen eine breitere Erfahrung.

Eine besonders ausgeprägte methodische Komponente bildet im Informatikstudium die Einführung in die selbständige Führung von Informatikprojekten. Mehrjährige Untersuchungen haben dabei gezeigt [Bürkler 81, Bürkler / Zehnder 81], dass dafür ein bestimmter Minimalaufwand für selbständige Projektarbeit nicht unterschritten werden darf, wenn die Studierenden wirklich verstehen sollen, um welche Probleme es bei der Projektleitung geht. Um diesen Minimalaufwand aber möglichst tief zu halten, sieht das Curriculum eine methodische Einführung „Informatik-Projektentwicklung“ („IPE“) von bloss zwei Semesterwochenstunden ($2 \times 16 = 32$ Lektionen) im 5. Semester vor, auf welche eine projektorientierte Semesterarbeit (ca. 150 Arbeitsstunden), ein Industriepraktikum (3 Monate) und bei Studienende die Diplomarbeit (ca. 800 Arbeitsstunden) folgen (Fig. 1). Bei diesen Gelegenheiten lernt jede Studentin und jeder Student, das theoretisch Gelernte in die Praxis umzusetzen.

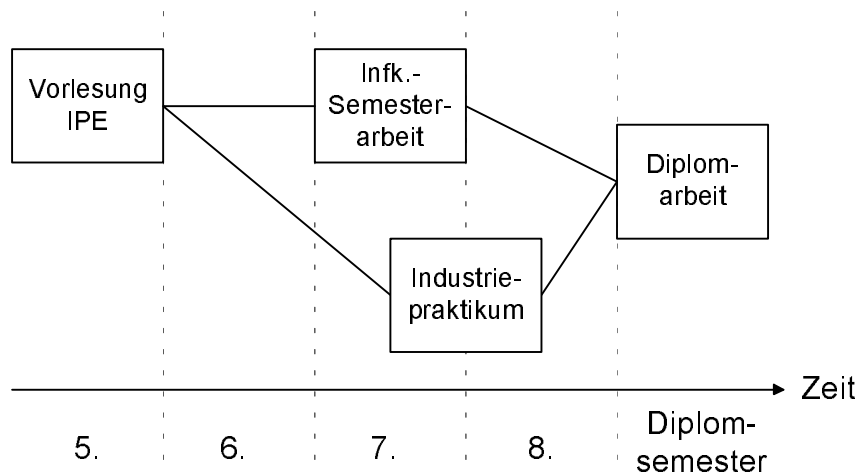


Fig. 1: Uebersicht über die wichtigsten Lehrveranstaltungen zur Ausbildung in Projektführungsfragen

¹ Im vorliegenden Bericht wird für Funktionsbezeichnungen in der Regel die kürzere (meist männliche) Form verwendet. Angehörige des jeweils nicht explizit genannten Geschlechts sind jedoch immer mitgemeint.

In der einführenden Lehrveranstaltung „Informatik-Projektentwicklung“ (Textbuch: [Zehnder 91]) erhalten die Studierenden neben vielen praxisorientierten Hinweisen vor allem eine methodische Einführung in die Projektarbeit anhand eines klassischen Phasenmodelles (Fig. 2). Dieses lässt sich aber auch flexibel abwandeln (z. B. für Prototypenbildung).

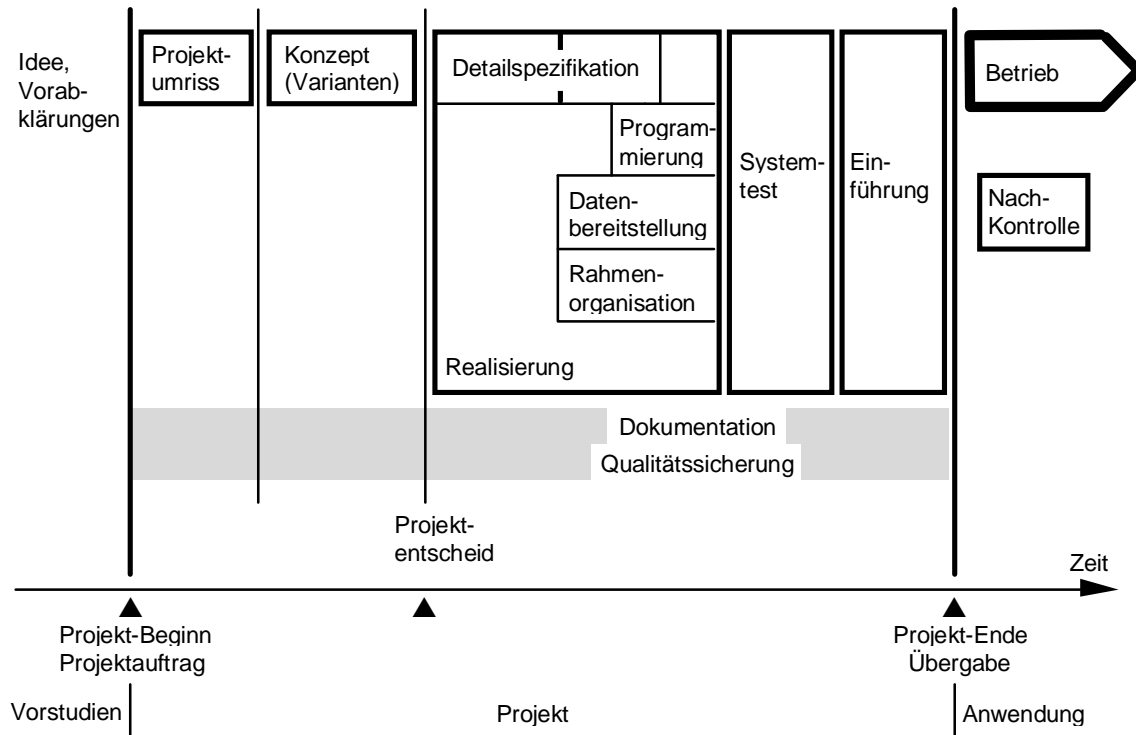


Fig. 2: Phasenmodell [Zehnder 91, Fig. 1.4]

Dieses Phasenmodell lässt sich schon im Studium mehrfach nutzbringend anwenden, nämlich für die vorgeschriebene Semester- und Diplomarbeit. Die Studierenden müssen dabei haushälterisch mit ihrer eigenen Arbeitskapazität umgehen, sowohl bei der bereits durch den Lehrplan als „projektorientiert“ gekennzeichneten Semesterarbeit zu einem Informatikthema im Umfang von ca. 150 Arbeitsstunden als auch bei der termingebundenen abschliessenden Diplomarbeit von exakt vier Monaten. Die Studierenden erhalten zur Selbstüberwachung ihres Arbeitsaufwandes auch technische Hilfe: Das im 4. Teil des vorliegenden „gelben Berichtes“ vorgestellte Programm „Zeitplan“ steht ihnen dazu gratis zur Verfügung.

Das methodische Ausbildungskonzept, wie es in Fig. 1 dargestellt ist, weist aber - wie meist in europäischen Hochschulen - einen grundsätzlichen Mangel auf: Es fördert einmal mehr das Individuum und nicht die Tätigkeit im Team. Seit 1991 versuchen wir daher, auch Gruppenprojekte systematisch zu unterstützen. Wie im 3. Teil erwähnt wird, konnten wir uns dabei auch auf Erfahrungen von Frederick P. Brooks an der University of North Carolina in Chapel Hill abstützen, der seit bald 20 Jahren auch Gruppenprojekte von Studierenden betreut (und dessen berühmtes Buch „The Mythical Man-Month“ [Brooks 75] manche hilfreichen Hinweise zur Projektführung ganz allgemein enthält). Ueber die ersten Zürcher Erfahrungen mit systematischer Betreuung von Gruppenprojekten berichtet der 2. Teil.

Damit sind wir bei der Zielsetzung des vorliegenden „gelben Berichtes“. Mit diesem wollen wir über die seit Jahren etablierte Ausbildung in Informatik-Projektentwicklung hinaus auch die Arbeit in Projektgruppen fördern. Die verschiedenen Texte sollen

- den Studierenden bei der Organisation ihres Gruppenprojektes helfen, und
- Kolleginnen und Kollegen Hinweise vermitteln, wie Gruppenprojekte effizient unterstützt und von vermeidbaren Schwierigkeiten entlastet werden können.

Dass die Förderung der Gruppenarbeit für künftige Informatiker wichtig ist, dürfte wohl unbestritten sein.

Literatur

[Brooks 75]

F. P. Brooks. *The Mythical Man-Month: Essays on Software Engineering*. Addison Wesley, 1975.

[Bürkler 81]

H. P. Bürkler. *Ausbildung in Projektentwicklung im Informatik-Studium an einer Hochschule*. Dissertation Nr. 6928 der ETH Zürich, 1981.

[Bürkler / Zehnder 81]

H. P. Bürkler, C. A. Zehnder. *The Training of University Students in Computer Project Management*. IFIP Conference on Computers in Education, Lausanne. In: Lewis & Tagg (ed.): *Computers in Education*. North Holland Publishing Co., Amsterdam 1981, p. 731-737.

[Zehnder 91]

C. A. Zehnder. *Informatik-Projektentwicklung*. Verlag der Fachvereine, Zürich und B.G. Teubner, Stuttgart, 2. Auflage, 1991.

Teil 2

Angewandte Gruppenprojekte

Ein Erfahrungsbericht aus der Ausbildung
in Informatik-Projektentwicklung

D. Aebi

Zusammenfassung

Die Abteilung für Informatik der ETH Zürich bildet Informatikingenieure in einem vierjährigen Ausbildungsgang aus. Unter der Bezeichnung „Angewandte Gruppenprojekte“ wurde im WS 91/92, im SS 93 sowie im SS 94 für Studenten und Studentinnen im 5.-7. Semester eine Veranstaltung angeboten mit dem Ziel, im Bereich Informatik-Projektentwicklung die Teamarbeit besonders zu fördern. Sechs Gruppen zu je vier Studierenden nahmen die Gelegenheit wahr, als Projektteams weitgehend selbständig ein Informatik-Projekt durchzuführen. Der folgende Bericht schildert die Erfahrungen und Probleme mit dieser Veranstaltung und gibt Anregungen für künftige Ausbildungsangebote dieser Art.

1. Einleitung

Die Lehrveranstaltung „Angewandte Gruppenprojekte“ (AGP) wurde an der Abteilung für Informatik der ETH Zürich erstmals im Wintersemester 91/92 im Sinne eines Versuches angeboten. Sie hatte zum Ziel, einzelnen Gruppen von Studenten¹ die Möglichkeit anzubieten, unter Anleitung ein kleineres Informatikprojekt weitgehend selbständig als Teamarbeit durchzuführen. Eine genügende Mitarbeit in einer solchen Gruppe sollte im Rahmen des Studienplanes als sogenannte grosse Informatiksemesterarbeit (150 Stunden) akzeptiert werden [ETH89, ETH91]. Aufgrund der positiven Resonanz der Teilnehmer wurde entschieden, die Veranstaltung im Sommersemester 93 bzw. 94 wiederum anzubieten.

Im Rahmen der AGP sollten die Studierenden eine ganze Reihe von Fähigkeiten und Techniken kennenlernen und anwenden, die in anderen Veranstaltungen nicht oder nur schlecht vermittelt werden können. Dazu sind zu zählen:

- Schulen der Zusammenarbeit in einer (strukturierten) Gruppe. Entwickeln von *Teamgeist*. Erarbeiten einer *gemeinsamen* Leistung.
- Möglichkeit schaffen, ein grösseres *zusammenhängendes* Problem zu bearbeiten.
- Umsetzen und Anwenden des in einer speziellen Vorlesung „Informatik-Projektentwicklung“ vermittelten Stoffes.
- Schulen der kommunikativen Fähigkeiten, Gesprächsführung.
- Aufbauen einer Zusammenarbeit mit fremden Personen („Auftraggebern“ bzw. „Anwendern“).
- Wecken des Bewusstseins für längerfristige Konsequenzen von Entscheiden.
- Arbeiten unter Zeitdruck über längere Zeit.

Alle Arbeiten wurden konsequent als Gruppenleistungen beurteilt. Im Vordergrund standen Projektführungs- und Organisationsaspekte. Die technischen Ergebnisse der einzelnen Projekte wurden als durchaus wichtig angesehen, ebenso bedeutsam war aber das von den einzelnen Gruppen gewählte *Vorgehen*.

Insgesamt haben bisher sechs Gruppen zu je vier Studenten, zwei im WS 91/92, drei im SS 93 und eine im SS 94, an dieser Veranstaltung teilgenommen. Die Betreuung wurde von Prof. C. A. Zehnder und einem Assistenten, dem Autor dieses Berichtes, wahrgenommen, im SS 93 und im SS 94 wurden für die anwendungsbezogene Betreuung zusätzliche Personen als „Auftraggeber“ beigezogen.

Der Bericht ist wie folgt gegliedert: Im Kapitel 2 wird die Methodik, nach der die Studenten arbeiten mussten, sowie die Organisation der Betreuung erläutert. Kapitel 3 beschreibt die Bildung der Gruppen und die Wahl der Aufgabenstellungen sowie die Unterschiede im Ablauf zwischen den einzelnen Semestern. In Kapitel 4 werden Ablauf

¹Erfreulicherweise hat sich an diesem Experiment auch eine Studentin beteiligt, was den Frauenanteil dieser Gruppe sofort weit über das bei Informatik-Lehrveranstaltungen an der ETH sonst leider übliche Mass gehoben hat. Im vorliegenden Bericht wird jedoch für Funktionsbezeichnungen konsequent die kurze (meist männliche) Form verwendet; der Autor ist sich jedoch der damit verbundenen Diskriminierung sehr wohl bewusst.

und aufgetauchte Probleme anhand der einzelnen Projektphasen besprochen. Kapitel 5 zeigt allgemeine Führungs- und Organisationsprobleme, die während allen Phasen aufgetaucht sind. In Kapitel 6 wird auf die Probleme der Betreuung einer solchen Veranstaltung eingegangen, und in Kapitel 7 werden mögliche Konsequenzen und Vorschläge für weiterführende Arbeiten diskutiert.

2. Methodik

2.1. Konkrete Aufgabenstellungen

Eine wesentliche Motivation für das gute Mitmachen der Studenten entstand durch die konsequente Verwendung echter Aufgabenstellungen Dritter als Themen der einzelnen Projekte.

2.2. Gliederung in Phasen

Um die in Kapitel 1 genannten Ziele erreichen zu können, wurde den Gruppen eine einheitliche methodische Vorgehensweise vorgegeben. Für die Projektstrukturierung wurde verlangt, dass konsequent nach dem Phasenmodell gearbeitet wurde, das den Studenten in der Vorlesung Informatik-Projektentwicklung erläutert worden war (Fig. 1). Die Projekte mussten (angepasst an die jeweiligen Aufgabenstellungen) nach folgenden Phasen bearbeitet werden:

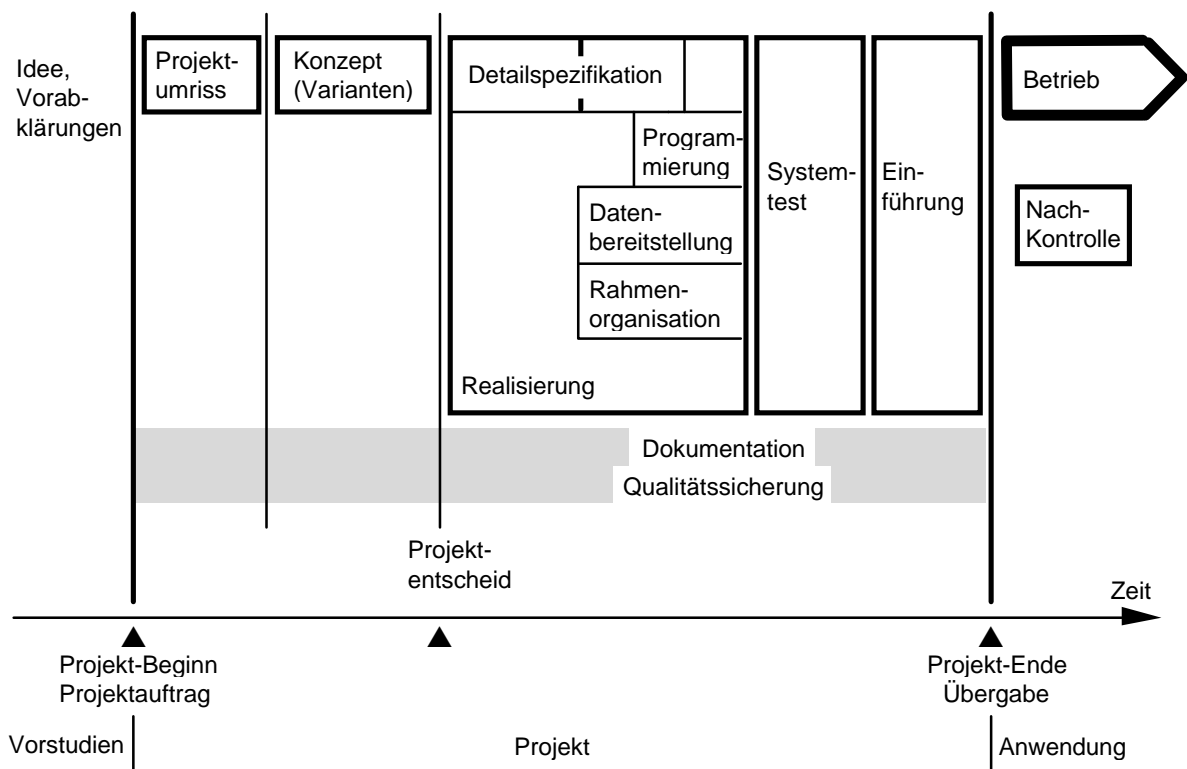


Fig. 1: Projektphasenmodell [Zehnder 91]

Diese Vorgabe bot einerseits Gewähr vergleichbarer Vorgehensweisen der einzelnen Gruppen, andererseits war damit bereits eine stützende Struktur für die Studenten vor-

gegeben. Zu jeder der Phasen gehören spezifische Arbeitsergebnisse (zum Teil in Form von Dokumenten), womit gleich zu Beginn auch eine Grobzeitplanung der zu leistenden Arbeiten möglich war.

2.3. Gruppengrösse und -struktur

Basierend auf einschlägigen Erfahrungen von F. Brooks [Brooks 91], der an der University of North Carolina at Chapel Hill diese Art der Ausbildung seit vielen Jahren durchführt, wurde eine einheitliche Gruppengrösse von vier Personen vorgegeben. Innerhalb der einzelnen Gruppen mussten folgende vier „Funktionen“ besetzt werden:

- **Produzent:** Organisatorischer Verantwortlicher, der die Voraussetzungen für die Arbeit der Gruppe schafft. Mögliche Aufgaben: Mittelbeschaffung, Gesprächsleitung, org. Abklärungen, Kontakte zu Benutzern, Zeit- und Terminplanung. Kann auch Arbeiten als Ingenieur übernehmen.
- **Chefingenieur:** Technischer Leiter des Projektes. Er ist für ein gutes technisches Gesamtergebnis des Projektes zuständig. Kann auch Arbeiten als Ingenieur übernehmen.
- **Ingenieure A & B:** Projektmitarbeiter in allen Phasen, die Teilaufgaben selbständig betreuen.

Die Besetzung der Funktionen wurde den einzelnen Gruppen überlassen. Eine Gruppengrösse von vier Personen hat sich bei Brooks als günstigste Grösse erwiesen. Kleinere Gruppen bieten zuwenig Beziehungsgeflechte innerhalb der Gruppe; es sollte ja auch so etwas wie Teamgeist entstehen können, grössere Gruppen verursachen andererseits zuviel an internem Kommunikations- und Steuerungsaufwand. Grössere Gruppen sind zudem schwieriger methodisch zu führen. Eine gerade Anzahl Teilnehmer bietet zudem Gewähr, dass wirkliche Entscheidungsfindungsprozesse stattfinden müssen, einfache Mehrheitsentscheide sind nicht so leicht möglich; auch werden Einzelne weniger leicht ausgegrenzt.

2.4. Betreuung, Mittel

Die Betreuung der einzelnen Gruppen erfolgte in regelmässigen Besprechungen der einzelnen Teams mit den Betreuern (Professor / Assistent). Dabei waren die Betreuer im WS 91/92 meist beide gemeinsam anwesend und nahmen dabei sowohl die methodische als auch die anwendungsbezogene Betreuung wahr. Diese Doppelfunktion (Methodikvermittler, Anwender/Auftraggeber) erwies sich als ungünstig, da die Studenten sehr schlecht in der Lage waren, diese beiden Aspekte sauber zu trennen. In den weiteren Veranstaltungen (SS 93 bzw. SS 94) wurden deshalb zusätzliche Personen für die anwendungsseitige Betreuung beigezogen, um dieses Problem zu entschärfen. Mehr zu diesem Aspekt findet sich im Kapitel 5.

An Produktionsmitteln für die Programmentwicklung standen den Teams im wesentlichen die im Departement vorhandene Hard- und Software zur Verfügung, sie durften jedoch auch (realistische!) Beschaffungsanträge stellen, falls dies für die konkreten Anwendungsbedürfnisse sinnvoll war, und die für ihre Arbeit angefallenen Spesen (Büromaterialien, Kopierkosten usw.) wurden ihnen selbstverständlich zurückerstattet.

Die Betreuer bemühten sich ganz allgemein, ein möglichst praxisnahes Umfeld für die Arbeit der einzelnen Gruppen zu schaffen.

3. Organisation

3.1. Vorkenntnisse

Um die Lehrveranstaltung durchführen zu können, wurden die AGP öffentlich für mindestens eine, besser aber zwei oder gar drei Gruppen ausgeschrieben. Damit das Feld potentieller Teilnehmer nicht gar zu stark eingeschränkt wurde, durften keine zu restriktiven Anforderungen an Vorkenntnisse der Teilnehmer gestellt werden. Als einzige Vorbedingung für eine Teilnahme wurde deshalb ein bestandenes zweites Vordiplom verlangt, was im WS 91/92 entsprechende Studenten im 5. und 7. Semester zuließ. Bei den Aufgabenstellungen musste Rücksicht genommen werden auf die vorhandenen fachlichen Vorkenntnisse, da die für die Projektarbeit zur Verfügung stehende Zeit nicht zuließ, dass sich die Studenten in ein völlig neues technisches Gebiet einarbeiten. An technischen Kenntnissen wurde deshalb lediglich Programmiererfahrung vorausgesetzt. Weitergehende theoretische Kenntnisse oder gar Erfahrungen auf dem Gebiet der Projektführung, insbesondere auch mit modernen Projektführungstechniken [Boehm 86, Hallmann 90, Schulz 89, Oertly 91], wurden selbstverständlich nicht vorausgesetzt. Es zeigte sich schon bei der ersten Durchführung, dass das 5. Semester eher zu früh liegt. Daher wurden ab 1993 die AGP auf das Sommersemester verlegt. Damit sollten primär Studenten im sechsten Semester angesprochen werden.

3.2. Gruppenbildung

WS 91/92

Die Veranstaltung wurde einerseits im Vorlesungsverzeichnis unter der Rubrik „Selbständige und Prüfungsarbeiten, Informatik“ (37-570A) angekündigt, andererseits wurde per Anschlag in den Schaukästen darauf hingewiesen. Es wurden nicht einzelne Studenten gesucht, sondern Gruppen mit genau vier Personen. Der erste gruppendynamische Prozess hatte deshalb schon vor Beginn der Veranstaltung stattzufinden, indem sich mögliche Interessenten Partner suchen mussten. Im Rahmen einer kurzen Vorstellung wurde unverbindlich über Form, Ablauf und Inhalt (Aufgabenstellungen) orientiert.

Es meldeten sich vorerst ein komplettes Team (4 Studenten) sowie ein einzelner Interessent. Anlässlich der Vorstellung der Veranstaltung entschied sich das Viererteam umgehend für die Teilnahme; dem einzelnen Studenten, der nach wie vor Interesse bekundete, wurde geraten, sich nach weiteren Teilnehmern umzusehen. Innerhalb einer Woche konnte so auch das zweite Team komplettiert werden.

Von den acht Teilnehmern standen sieben im fünften und einer im siebten Semester.

SS 93

Für die nächste Durchführung wurde bewusst ein Sommersemester (6. oder 8. Semester der Studenten) gewählt. So konnte die Veranstaltung AGP bereits im vorangehenden Wintersemester in der für Studenten im fünften Semester obligatorischen Vorlesung „Informatik-Projektentwicklung“ von Prof. Zehnder kurz vorgestellt werden. Es meldeten umgehend mehrere Gruppen ihr Interesse an. Die ersten drei wurden berücksichtigt. Weiterführende Publizität war deshalb nicht nötig.

SS 94

Auch für diese Veranstaltung wurde Ende des Wintersemesters, nach entsprechender Ankündigung in der Vorlesung „Informatik-Projektentwicklung“, eine Vorbesprechung durchgeführt. Es meldete sich diesmal nur eine Gruppe an.

3.3. Aufgabenstellungen

Mit jedem AGP sollten konkrete und „nützliche“ (d.h. von jemandem formulierte und erhoffte) Ergebnisse produziert werden. Damit die gestellten Ziele in der verfügbaren Zeit erreicht werden konnten, mussten die Aufgabenstellungen entsprechend gewählt werden. Es wurden durchwegs Aufgaben aus dem Umfeld der Hochschule gesucht, die nach Einschätzung der Betreuer durch eine Gruppe von vier Studenten in der zur Verfügung stehenden Zeit gelöst werden konnten. Die Aufgaben wurden nach Möglichkeit so gewählt, dass sie auch eine Softwarekomponente enthielten, so dass etwas konkret implementiert werden musste.

Die Aufgabenstellungen im WS 91/92 entstammten alle der Hochschulverwaltung. Sie wurden mit dem Leiter der Administrativen Informatik vorbesprochen, dann aber von den Betreuern konkret formuliert. Zur Auswahl standen:

1. Entwicklung eines Informationssystems (IS) mit allgemeinen Informationen über die ETH, im Sinne eines Auskunftssystems (z. B. in der Eingangshalle des Hauptgebäudes).
2. Entwicklung eines IS für on-line-Abfragen des Semesterprogrammes der ETH.
3. Entwicklung eines IS für on-line-Abfragen über die Inhalte der Lehrveranstaltungen aller Abteilungen der ETH („Katalog der Lehrveranstaltungen“).

Die Wahl der Aufgabe wurde den Teams überlassen. Da zu Beginn erst ein komplettes Team zur Verfügung stand, stellte die Verteilung der Aufgaben kein Problem dar; es wählte die zweite Aufgabe. Nach Komplettierung wählte das zweite Team die Aufgabe drei.

Für die Veranstaltung im SS 93 wurde vorerst innerhalb des Departementes Informatik nach möglichen anwenderseitigen Betreuern und entsprechenden Projekten gesucht. Es meldete sich ein einziger Professor aus der Datenbankgruppe. Durch gezielte Anfragen konnten in den Informatikdiensten der Hochschule beim Leiter des Computer-Service sowie dem Leiter der Administrativen Informatik zwei weitere Aufgaben gefunden werden. Damit standen folgende Projekte zur Verfügung:

1. Entwicklung einer Komponente für ein adaptives Datenverwaltungssystem (Komponente eines Forschungsprojektes).
2. Entwicklung eines IS mit allgemeinen Informationen über die ETH (im wesentlichen die Aufgabe, die im WS 91/92 übrig geblieben war).
3. Entwicklung eines IS für die Abwicklung des Rapportwesens des Computer-Service.

Die Verteilung der Aufgaben gestaltete sich diesmal wesentlich schwieriger. Im Anschluss an eine allgemeine (unverbindliche) Vorstellung der Themenbereiche mussten sich die drei Gruppen auf die Verteilung der Aufgaben einigen. Interessanterweise wünschten alle drei Gruppen auf Anhieb die Aufgabe eins aus der Datenbankforschung,

von der eine relativ konkrete Umschreibung vorlag und die im wesentlichen eine relativ klare Entwurfs- und Programmierarbeit darstellte und deshalb aus Projektentwicklungssicht eher wenig hergab. Die Aufgaben wurden deshalb mittels Los verteilt.

Da sich für das SS 94 nur eine einzige komplette Gruppe meldete, gestaltete sich die Themenvergabe einfach. Zur Verfügung standen folgende Aufgaben:

1. Entwicklung eines Literaturverwaltungssystems.
2. Entwicklung eines on-line-IS mit allgemeinen Informationen über das Department Informatik.

Die Studenten wählten die zweite Aufgabe.

4. Ablauf

Im folgenden wird für jede Phase gemäss Phasenmodell (Fig. 1) jeweils kurz beschrieben, was darin zu leisten ist. Darauf folgt die Beschreibung der Probleme, die sich für die Studenten ergeben haben.

4.1. Projektumriss

Zweck und Ziel

Umschreibung und Abgrenzung des Problembereiches. Problemanalyse, Skizzieren erster Lösungsideen. Formulierung eines Pflichtenheftes. Klarheit über den Rahmen der zu bearbeitenden Aufgabe gewinnen. Reduktion auf Machbares.

Erfahrungen

Der Start verlief bei allen Teams und jedes Jahr wieder erwartungsgemäss harzig. Hier zeigte sich die mangelnde Erfahrung mit grösseren Aufgaben deutlich. Keines der Teams war im Stande, selbständig eine saubere Auftragsanalyse („Was wird von uns erwartet, wie sind die Zeitverhältnisse, welche Mittel stehen zur Verfügung, welches sind die Rahmenbedingungen“ usf.) durchzuführen. Mit wenigen Hinweisen konnte jedoch anlässlich der Projektsitzungen manches rasch korrigiert werden. In den ersten paar Wochen ging damit jedoch wertvolle Zeit verloren, insbesondere auch, weil keine Delegation stattfand und sich alle Teilnehmer gleichzeitig mit der Aufgabe auseinandersetzten.

Bereits in dieser Phase trat ein Phänomen deutlich zu Tage, das auch im weiteren Projektverlauf immer wieder anzutreffen war: Die Studenten bekundeten erhebliche Mühe, ihre Aufgabestellung konstruktiv mitzugestalten und sich dabei die längerfristigen Konsequenzen ihrer Entscheide vorzustellen. Dies verwundert nicht, sind Studenten doch gewohnt, im Uebungsbetrieb die Aufgabenstellungen fertig geliefert zu bekommen. Sobald sie nicht einfach eine Lösung für eine Aufgabe liefern, sondern zuerst die Aufgabenstellung selber erarbeiten müssen, sind sie überfordert. Es zeigte sich bei den Besprechungen sehr oft eine standardisierte studentische Denkweise, wie sie vom sonst üblichen Uebungsbetrieb geprägt wird:

Neues Thema auf Grund einer schriftlichen Aufgabenstellung
⇒ Lösen der Aufgabe ⇒ Abgeben ⇒ Korrektur durch den Assistenten ⇒ ev. Besprechung des Resultates ⇒ Aufgabenstellung vergessen ⇒ Neues Thema usw.

Alle Studenten hatten zu Beginn Mühe, den Aufbaucharakter des Projektes richtig zu verstehen („Was wir heute tun, hat Konsequenzen für morgen“). Es dauerte deshalb auch viel zu lange, bis ein brauchbares Pflichtenheft vorlag (einzelne Gruppen waren gar nicht im Stande, ein solches selbständig zu erstellen). Zum Teil wurde sofort mit grossem Engagement an Realisierungen gearbeitet, während die Studenten noch ohne klare Vorstellungen über Form und Inhalt („was ist ein Pflichtenheft“, „wozu soll es dienen“) waren.

In dieser Phase ging es seitens der Betreuer vor allem auch darum, die Bäume nicht in den Himmel wachsen zu lassen und sich auf *das Machbare* zu beschränken. Die meisten Gruppen schlugen sich allzulange mit unrealistischen, zu umfangreichen und zu aufwendigen Varianten herum.

Als weiteres wesentliches Problem zeigte sich auch bereits in dieser frühen Phase, dass die Studenten nicht gewohnt sind, im Gespräch ein Problem zu lokalisieren und zu lösen, sei es intern im Team, sei es mit externen Personen („Anwendern“). Sie versuchten anlässlich von Sitzungen oft vor allem sich zu rechtfertigen, also die von *ihnen* gewählte Lösung zu verteidigen, statt offen mit den Anwendern *gemeinsam* eine gute Lösung zu erarbeiten. Dieses Phänomen hielt sich leider hartnäckig oft bis zum Schluss. Zum Teil rührte das Problem aus der Doppelfunktion der Betreuer her, die namentlich im WS 91/92 gleichzeitig methodische und anwendungsspezifische Aspekte vertraten. Durch die „Semesterarbeit“ war die übliche Distanz zwischen (benotendem!) Professor und Studenten gegeben; im Projekt aber sollte gemeinsam eine Lösung entworfen werden (siehe auch 5.3).

4.2. Konzepterarbeitung, Variantenentscheid

Zweck und Ziel

Grobentwurf von *machbaren* Lösungsvarianten. Basierend auf dem Pflichtenheft sollen zwei bis drei Lösungsvarianten mit Vor- und Nachteilen, auch unter Berücksichtigung wirtschaftlicher Aspekte erarbeitet werden (im vorliegenden Fall war die zur Verfügung stehende Zeit die knappste Ressource). Für den Variantenentscheid sind Grundlagen vorzubereiten.

Erfahrungen

Beim Ausarbeiten von Varianten zeigte sich zu Beginn wiederum die mangelnde Klarheit über Sinn und Zweck des Vorgehens. Die Studenten wussten zwar, dass die Betreuer von ihnen Variantenvorschläge erwarteten. Viel Zeit wurde aber auf das Ausarbeiten von nicht realisierbaren Varianten verwendet. Zum Teil wurden auch zu viele Freiheitsgrade eingeführt, was dann zu einer allzu grossen Zahl von möglichen Varianten (Kombinationen aller Möglichkeiten) führte. Anlässlich der Sitzungen fiel auf, dass die Studenten von den Anwendern sehr wenig Fachinformation verlangten (so dass oft

nicht klar wurde, ob gar keine Probleme vorliegen oder die Aufgabe nicht richtig verstanden war). Als wichtigstes Problem erwies sich wiederum die mangelnde Erfahrung im Einteilen und Abschätzen der zur Verfügung stehenden Zeit. So musste immer wieder eindringlich auf ein vorausschauendes Planen auch der Varianten hingewiesen werden (z. B. Beschaffungszeiten für Software). Auch in dieser Phase wurde meist mit der Ressource Arbeitszeit zu wenig haushälterisch umgegangen. Die verfügbaren Kapazitäten (z. B. der Ingenieure A & B) wurde viel zu wenig für das parallele Durchführen von Machbarkeitsstudien, Abklären der Verfügbarkeit von Mitteln, Einarbeiten in Hilfsmittel (Software) usf. genutzt.

Bei einzelnen Gruppen war trotz Instruktion das typisch studentische Verhalten der raschen Lösungsidee zu beobachten: Diese Studenten hatten grosse Mühe einzusehen, weshalb sie nun für *ein* Problem *mehrere* Lösungsvorschläge erarbeiten sollten. Aus dem normalen Übungsbetrieb, etwa im Programmieren, kannten sie diese Forderung nicht.

Variantenentscheid

Um die Bedeutung des Variantenentscheides zu unterstreichen, mussten die Gruppen im Wintersemester 91/92 dazu anstatt der normalen Projektsitzung eine eigentliche Präsentation („Vortrag“) durchführen. Die Teams mussten dabei einen Raum organisieren und ihre Varianten mit Vor- und Nachteilen relativ formell präsentieren. Dazu wurde (in einem Team sofort, im anderen nach anfänglichem Widerstand) vereinbart, Schriftdeutsch zu sprechen. Zu einer Präsentation war ein Gast eingeladen, der im Projekt eine beratende und unterstützende Funktion hatte.

Im Sommersemester 93 mussten die Teams den Variantenentscheid selbständig mit ihren anwenderseitigen Betreuern durchführen. Die methodischen Betreuer waren (mit einer Ausnahme) bei diesen Präsentationen nicht dabei. Im SS 94 waren alle Beteiligten anwesend.

Es zeigte sich eine grosse Unsicherheit über die beste Form für Inhalt und Ablauf des Variantenentscheides (was muss präsentiert werden, in welcher Form, wer entscheidet usf.). Allzu häufig biss man sich dabei fest in Detailproblemen („auf welchem Drucker wird gedruckt“); deshalb entstanden oft zu viele „Detailvarianten“ statt Gesamtlösungen. Oft herrschte nach wie vor Unklarheit darüber, was nun als Variante zu gelten hatte; ein Team beschränkte sich auf eine sehr ausgiebige Diskussion über das für die Implementation zu verwendende Werkzeug, statt einen Lösungsweg für das Problem zu präsentieren! Diese Probleme konnten erst im Rahmen weiterer Projektsitzungen aufgearbeitet werden.

Die Präsentation der einzelnen Varianten erfolgte unterschiedlich: Ein Team erkor einen Sprecher, der die gesamte Präsentation bestritt; die anderen Teammitglieder standen lediglich für ergänzende Fragen zur Verfügung. Ein anderes Team gliederte die Präsentation in mehrere Blöcke, die jeweils von einem Teammitglied präsentiert wurden. Die gezeigten Varianten waren jeweils unterschiedlich seriös vorbereitet, so dass daraus bei allen Teams dessen Wunschvariante in der Regel leicht ersichtlich war (auch besteht der Verdacht, dass nicht alle Varianten eine sinnvolle Lösung ergeben hätten). Um die Arbeit nicht unnötig zu verzögern und da in allen Fällen, allenfalls nach entsprechenden Korrekturen durch die Betreuer, durchaus machbare Varianten beantragt wurden, ent-

schieden sich auch die Betreuer für die jeweils von den Teams beantragte Wunschvariante.

4.3. Realisierung

Zweck und Ziel

Das Ziel dieser Phase ist die vollständige Entwicklung der zukünftigen Problemlösung. Basiert wird auf dem Grobkonzept (gewählte Variante). Die Realisierung umfasst eine Unterphase Detailspezifikation, die wiederum in eine fachliche und technische Teilphase aufgegliedert werden soll, und die je nach Bedarf anschließenden Unterphasen Programmierung, Datenbereitstellung und Rahmenorganisation.

Erfahrungen

Diese Phase liess nebst einem Mangel an Erfahrung oft auch einen Mangel an Ausbildung erkennen. So verfügten die Studenten zwar über grundlegende Kenntnisse in Programmierung, zum Teil auch in Datenmodellierung [Zehnder 89]; der Gesamtentwurf eines Informationssystems, bestehend aus einem Daten- einem Prozessmodell sowie einer Benutzerschnittstelle, ist ihnen jedoch fremd. Es fehlten ihnen Kenntnisse, um eine saubere Zustandsanalyse durchzuführen und mit geeigneten Hilfsmitteln darzustellen [Yourdon 92, Oesterle 92], beziehungsweise einen Entwurf einer ganzen Anwendung zu spezifizieren. Ihre diesbezüglichen Anstrengungen erschöpften sich in der Regel in einem Datenmodell und einigen Handskizzen von Bildschirmmasken. Es fehlten allgemeine Kenntnisse in Software Engineering „at large“ [Denert 92]; aber auch spezielle Techniken, z. B. für ein effizientes Benutzerschnittstellenprototyping [Pomberger 91], waren ihnen unbekannt. Hingegen wollten sie endlich „loslassen“ und programmieren, weil sie das gelernt hatten. Diese Mängel liessen sich durch die Betreuer nicht kurzfristig beheben!

Diese Mängel führten bei fast allen Teams zu einer Vermischung von Analyse, Design und Programmierung und damit zu suboptimalen Lösungen, was auch eine vernünftige Zeitplanung sehr erschwerte. Alle Teams hatten zudem in dieser Phase Mühe, die künftigen Anwender sinnvoll in die Entwurfsarbeit einzubeziehen (Motto: Wir sind die Informatiker, wir wissen schon, was der Kunde braucht).

In dieser Phase wurde von allen Teams als wichtiger Punkt die Abklärung über den Lieferumfang erkannt („Was ist am Ende wem in welcher Form abzugeben?“).

Die Programmierung war gekennzeichnet durch weitgehend selbständiges Arbeiten der Teams. Es schien, dass sie sich dabei nicht „in die Karten schauen“ lassen wollten. So wurden von einzelnen Teams recht kurzfristig Sitzungen mit den Betreuern/Anwendern abgesagt, vielleicht um störende Einflüsse seitens der Anwender zu vermeiden und Tatsachen zu schaffen. In den Teams jedoch herrschte reges Treiben und es wurde in der Regel fleissig gearbeitet. Methodische Begleitung war in dieser Phase kaum mehr gefragt und auch nicht mehr möglich. In dieser Phase rächten sich nun allerdings die Versäumnisse der Phase Variantenausarbeitung, wo manche Abklärungen über die verfügbaren Werkzeuge und Hilfsmittel allzu oberflächlich geblieben waren, so dass sich jetzt damit unerwartet grosse Probleme ergeben konnten, die im Rahmen einer Abklärung / Machbarkeitsstudie früher hätten erkannt und behoben oder umgangen werden können.

Das Team im SS 94 hatte keine eigentliche Programmieraufgabe zu lösen. Es ging vielmehr darum, ein bestehendes Produkt entsprechend zu konfigurieren und eine geeignete Rahmenorganisation und Datenbereitstellung zu organisieren. Auch hier machte sich jedoch der Mangel an entsprechenden Erfahrung stark bemerkbar. Durch häufige, gezielte Interventionen seitens der Betreuer/Anwender konnte jedoch manches noch auf den richtigen Weg gebracht werden.

4.4. Test / Uebergabe

Alle Gruppen erkannten die offenen Probleme bei Installation/Einführung/Schulung frühzeitig und trafen entsprechende Vorkehrungen. Es klaffte jedoch eine Lücke zwischen dem 'Wie' der Uebergabe und dem 'Was'. So wurde zum Teil bereits zu einem Zeitpunkt über die Uebergabe diskutiert, als über den Umfang des Produktes noch keine abschliessende Information vorlag.

Fast alle Teams präsentierten ihre (fast fertigen) Arbeiten kurz vor Projektendtermin. Dabei fiel auf, dass sie dieser Uebergabepäsentation zu wenig Bedeutung beimassen (was auch in den eigentlichen Schlusspräsentationen nochmals verstärkt zum Ausdruck kam). Durch eine seriöse Vorbereitung, worunter auch das Bereitstellen geeigneter Test- und Präsentationsdaten zu zählen ist, hätte sich die geleistete Arbeit oft mit geringem Aufwand wesentlich besser darstellen („verkaufen“) lassen. Viele kleine Mängel (falsch beschriftete Menüpunkte, unkorrekt geschriebene Namen in Testdaten usw.) wurden mit dem für Studenten einleuchtenden Argument entschuldigt, dass deren Behebung eine Kleinigkeit sei. Präzisere Vorarbeiten hätten aber zu einem deutlich besseren Eindruck geführt.

Leider wurden diese kleinen Mängel zum Teil auch für die ansonsten recht gelungenen Schlusspräsentationen nicht rechtzeitig behoben. Hier hätte ein bisschen mehr „Show“ nicht geschadet. Die abgelieferten Unterlagen waren meist sehr umfangreich (allenfalls zu umfangreich).

4.5. Zusammenfassung

Mit einer einzigen Ausnahme gelang es allen Teams, die gestellte Aufgabe in der vorgegebenen Zeit weitgehend abzuschliessen. In zwei Fällen konnte gar ein Praxisbetrieb aufgenommen werden. Die Motivation und das Interesse an dieser Form von Arbeit war bei den Studenten gross, was auch in den entsprechenden Schlussberichten zum Ausdruck gebracht wurde.

5. Führungs- und Organisationsaspekte, Arbeitstechnik

5.1. Gesprächsführung

Für die in der Regel wöchentlich stattfindenden Sitzungen wurden Gesprächsführung und Inhalt den einzelnen Gruppen überlassen. Die Studenten sollten damit gezwungen werden, die Sitzungen seriös vorzubereiten und im Gespräch zügig die Punkte zu berechnen, die für sie wichtig waren. Diese Besprechungen förderten rasch und bei allen Gruppen, wenn auch in unterschiedlicher Ausprägung, eine Reihe von allgemeinen arbeitstechnischen Problemen zutage:

- Die Teilnehmer verfügten über keine Erfahrung in Gesprächsführung. Gesprächsführungshilfsmittel wie z. B. eine Traktandenliste (u. a. zur Vorbereitung der Gesprächspartner) waren ihnen zu Beginn fremd.
- Die Gespräche waren (mindestens zu Beginn) oft unstrukturiert und drohten zum „Geplauder“ zu werden. Hier war stetiger Einfluss der Betreuer nötig.
- Das Erstellen eines konzisen Gesprächsprotokolles verursachte Schwierigkeiten. Mehr als einmal wurden Aussagen falsch protokolliert.
- Die Gesprächsführung wurde (zu Beginn) in der Regel an den „Produzenten“ delegiert. Eine vernünftige Stellvertreterregelung bestand zu Beginn nicht (was sich sehr schön zeigte, als ein Produzent einmal zu einer Sitzung zu spät kam).
- Es wurde zu Beginn (aufgrund der verständlichen Unsicherheit) zu viel Zeit für das Erstellen von (zum Teil in dieser Form gar nicht benötigten) Dokumenten verwendet. Ein klares Konzept über die zu erstellenden Unterlagen bestand nicht.

Alle diese Effekte waren aufgrund von Vorbildung und Erfahrung der Teilnehmer erwartet worden. Eine Verbesserung konnte im Projektverlauf erzielt werden („learning by doing“). Es ist jedoch festzuhalten, dass recht viel der zur Verfügung stehenden Besprechungszeiten aufgrund solcher Schwierigkeiten verlorenging und die Betreuer gelegentlich starken Einfluss nehmen mussten, um diese Mängel zu korrigieren.

5.2. Zeitplanung

Ein Thema, das die Gemüter während allen Projektphasen bewegte (und zum Teil auch erhitzte!), war die angemessene Zeitplanung. Es dauerte sehr lange, den Studenten die enorme Bedeutung einer frühen, nicht übertriebenen, aber genügend genauen Zeitplanung für ihre Arbeit klarzumachen. Dabei waren sowohl formale („Wie stellt man einen Zeitplan dar?“) als auch inhaltliche Fragen („Wie führt man eine vernünftige Zeitschätzung durch?“) zu klären. Leider erwies sich der Lernfortschritt auf diesem Gebiet als sehr gering. Die Studenten zeigten sich weitgehend ausserstande, eine vernünftige, rollende Zeitplanung zu machen. Sie verschwendeten zu viel Zeit auf Aspekte der Darstellung (Konzentration auf Form statt Inhalt) und machten sich zu wenig Gedanken über die zeitlichen Konsequenzen ihrer Entscheide. Auch dies ist ein Resultat des üblichen Übungsbetriebes, der die Studenten nie zwingt, sich über längerfristige Konsequenzen von Entscheiden Gedanken zu machen (herkömmliche Übungen sind immer so angelegt, dass sie in einem fest vorgegebenen Zeitraum von ein oder zwei Wochen lösbar sind!).

Um dieses Problem ein wenig zu entschärfen, wurde im Sommersemester 92 im Rahmen einer Semesterarbeit ein computergestütztes Werkzeug für Gruppenprojekte entwickelt, das die Darstellung eines Zeitplanes als Balkendiagramm erlaubt. Das Werkzeug sollte zudem die Möglichkeit bieten, zu den einzelnen Phasen auch Dokumente („Dokumentenbudget“) und Mitarbeiter verwalten zu können. Es zeigte sich jedoch nach der Fertigstellung, dass aufgrund der grossen Funktionalität die Benützung dieses Werkzeuges zu kompliziert und zu aufwendig wurde. Wichtig war vor allem, rasch nur einen Zeitplan erstellen zu können (eine Aufgabe, die typischerweise auch von allen Diplomanden geleistet werden muss). Daher wurde eine einfachere Version erstellt (Fig. 2). Dieses Werkzeug ist inzwischen in einer auf Mac und einer auf DOS/Windows laufenden Version vorhanden. Details hierzu sind im Teil 4 zu finden.

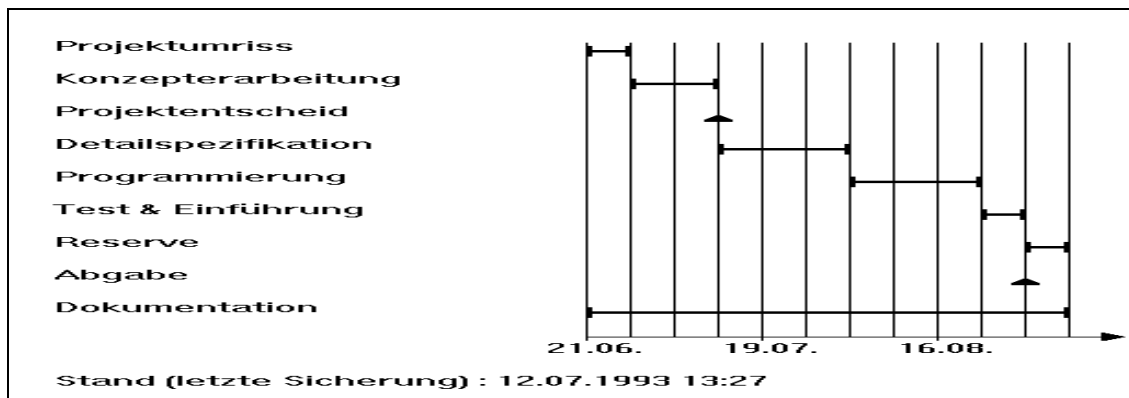


Fig. 2: Zeitplan

5.3. Umgang mit den Betreuern

Das Aufbauen einer tragfähigen Zusammenarbeit mit fremden Personen, ist zwar eines der mit den AGP zu erreichenden Ziele, bildete aber für alle Gruppen ein Problem. Es gelang den Studenten in der Zeit ihrer Projektarbeit nicht, ihre „Verteidigungshaltung“ gegenüber den Betreuern völlig abzulegen. Auch dies ist eine direkte Folge des üblichen Übungs- und Prüfungsbetriebes, der die Studenten zwingt, jeweils ihre eigene Lösung eines Problems möglichst gut zu vertreten. Trotz erheblicher Anstrengungen gelang es deshalb nur beschränkt, eine konstruktive Zusammenarbeit der Gruppe mit den Betreuern zu etablieren; die Distanz Professor - Student blieb zu gross. Um diese Situation etwas zu verbessern, wurden im SS 93 bzw. SS 94 zusätzlich Personen mit der anwenderseitigen Betreuung beauftragt. Dies half zwar, eine saubere Trennung zwischen methodischen und anwendungsbezogenen Fragestellungen durchzusetzen; trotzdem entstand keine völlig offene Zusammenarbeit mit den Anwendern (im Sinne einer gemeinsamen Denkweise und der Fähigkeit, Ideen der „anderen“ unmittelbar aufzunehmen). Die Studenten beharrten nach wie vor allzuoft auf der Rechtfertigung ihrer eigenen Lösungen. Es ist offenbar für unerfahrene Studenten nicht einfach, einen vernünftigen Mittelweg zwischen Vertreten der eigenen Positionen (was auch wichtig ist) und konstruktiver Zusammenarbeit mit den Anwendern zu finden (was manchmal auch Nachgeben bedeutet).

6. Bemerkungen zur Betreuung

Eine Veranstaltung wie die AGP, die gleichzeitig eine Vielzahl von Fertigkeiten und Techniken schulen und anwenden will, stellt auch an die Betreuer höhere Anforderungen als der gewöhnliche Übungsbetrieb.

Die zeitliche Belastung für die Betreuung dieser Veranstaltung ist auch für einen erfahrenen Projektleiter recht hoch. So ist (ohne die Projektvorbereitung) während des laufenden Projektes etwa mit einer bis zwei Stunden pro Woche und Gruppe zu rechnen. Da mit den einzelnen Gruppen in aller Regel individuelle Probleme besprochen werden müssen, können die Gruppen nicht zusammengelegt werden. Eine Ausnahme bilden höchstens eine allgemeine Einführung zu Beginn sowie allenfalls eine gemeinsame Schlussbesprechung. Assistenten, die AGP betreuen, leisten damit einen Beitrag zum Unterricht, wie dies andere in Übungen tun.

Die bisherigen Erfahrungen haben aber gezeigt, dass die Betreuung von AGP nicht von allen Assistenten geleistet werden kann. Dazu muss ein genügend grosser, eigener Wissens- und vor allem auch Erfahrungsvorsprung aus Projekten vorhanden sein, um die Tätigkeiten der Studenten aus der nötigen Distanz beurteilen zu können. Einige Jahre praktische Erfahrung in den Bereichen Anwendungsentwicklung und/oder Projektleitung sind dabei von wesentlichem Vorteil. Diese Voraussetzungen sind normalerweise nur bei wenigen Assistenten erfüllt, da die meisten erst vor kurzem ihr eigenes Diplomstudium abgeschlossen haben. Neben der Projekterfahrung sind vor allem auch Erfahrungen und Fähigkeiten von Nutzen, die nicht auf Informatik beschränkt sind (z. B. Gesprächsführung, Arbeitstechnik) und die an beliebigen Orten erworben sein können (Verein, Politik, Militär u. a.). Ein Betreuer für AGP sollte als minimale Voraussetzung zumindest bereits einige Studentenarbeiten (Diplom- und Semesterarbeiten) betreut haben. Vorschläge zur Ausbildung der Assistenten finden sich in Abschnitt 7.2.

7. Konsequenzen - zukünftige Arbeiten

7.1. Allgemeine Eindrücke

Das grosse Interesse und die überwiegend positiven Kommentare der Teilnehmer lassen darauf schliessen, dass ein Bedarf nach einem Ausbildungsangebot vorhanden ist, das die Themen Projektführung und Gruppenarbeit abdeckt. Die bisher gemachten Erfahrungen mit AGP müssen jedoch aus organisatorischer und inhaltlicher Sicht auch kritisch betrachtet werden:

- Der Betreuungsaufwand für solche Lehrveranstaltungen ist recht hoch (Gruppenbetreuung statt Semesterklassen). Andererseits sind für die Betreuer eine Fülle interessanter Beobachtungen zu machen und Erfahrungen zu gewinnen, so dass die Betreuungsarbeit an sich als sehr attraktiv empfunden wurde.
- Die Studenten verfügen in der Regel auch im oberen Teil des Diplomstudiums (5.-8. Semester) weder über genügend breite technische Kenntnisse, insbesondere was die Detailspezifikation einer Anwendung anlangt, noch über organisatorische Erfahrungen (Gesprächsführung, Zeitplanung usw.), um ein Informatikprojekt weitgehend selbständig durchführen zu können.
- Auf der Betreuerseite stellt die Lehrveranstaltung Anforderungen und verlangt Vorkenntnisse, die auch bei älteren Assistenten nicht selbstverständlich erwartet werden können. Eine gezielte Ausbildung der Betreuer ist nötig.
- Da eine erfolgreiche Teilnahme an einem AGP als Semesterarbeit nach Studienplan anerkannt wird, die Arbeit jedoch in Vierergruppen zu leisten ist, wird den anderen Fachgruppen mit dieser Veranstaltung eine stattliche Anzahl Semesterarbeiten „weggenommen“.
- Die Bereitstellung geeigneter Aufgabenstellungen ist nicht einfach. Erwünscht sind Aufgaben, welche wichtige Weichenstellungen für die Realisierung noch offen lassen. Reine Programmierarbeiten, wie sie auch von einer einzelnen Person geleistet werden können, sind ungeeignet.
- Die bei der Einführung vorgegebene Gruppenstruktur (siehe Abschnitt 2.3.) bringt nicht nur Vorteile. Insbesondere lässt sich die Funktion des Chefingenieurs nur schwer sinnvoll besetzen. Die Struktur einzelner Gruppen wurde im Verlaufe

der Projekte geändert. Hier sind Überlegungen nötig, ob andere Funktionen vorgegeben werden sollen oder ob die interne Struktur den einzelnen Gruppen überlassen werden soll. Es ist jedoch wichtig, von Beginn weg unterschiedliche Tätigkeitsbereiche zu identifizieren und innerhalb der Gruppe durch verschiedene Personen besetzen zu lassen.

Die Erfahrungen mit den bisherigen Gruppen führen zu einer Reihe von Konsequenzen, die eine künftige Organisation und Durchführung von Ausbildungsangeboten dieser Art vereinfachen können.

7.2. Ausbildung der Betreuer

Wie bereits im Kapitel 6 angedeutet, kann eine solche Veranstaltung nicht mit unerfahrenen Betreuern („Informatiker mit Diplom, aber ohne Erfahrung“) durchgeführt werden. Entsprechende Vorkenntnisse und Erfahrungen sind unabdingbar. Andererseits dürfte auch von Seiten vieler Assistenten ein Bedürfnis bestehen, Projekt- und Gruppenerfahrungen sammeln zu können. Diese könnten zum Beispiel in folgendem mehrstufigen Verfahren erworben werden:

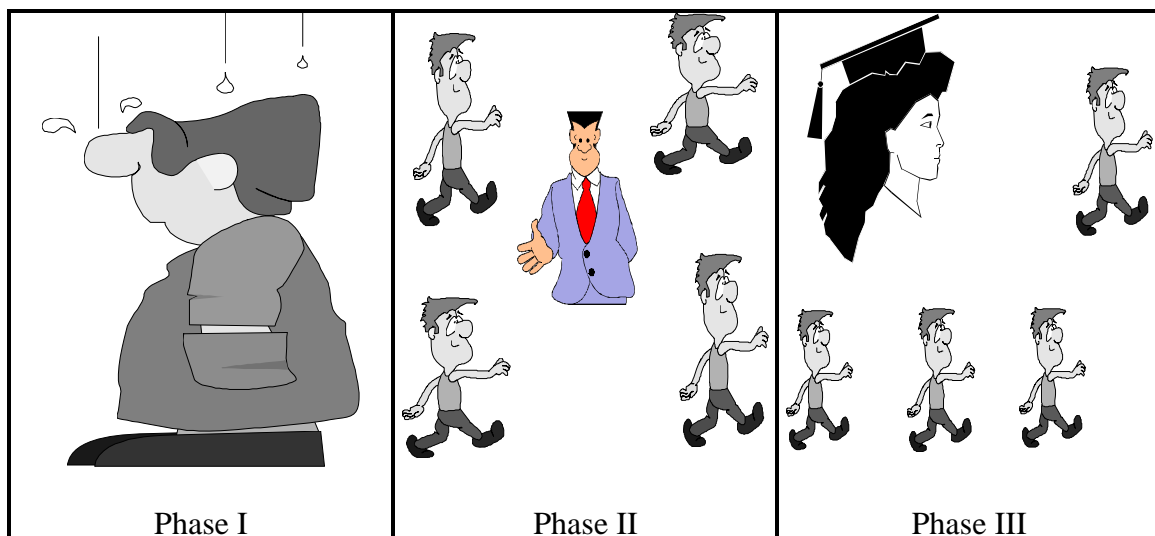


Fig. 3: Ausbildung der Betreuer

- Phase I: Der junge Assistent führt selbst ein kleines Projekt durch, allenfalls unter direkter Leitung des Gruppenchefs (Dozent). Diese Art Einstieg wird in manchen Fachgruppen bereits praktiziert. Allenfalls kann ein erfahrener Assistent oder Oberassistent hier eine beratende Funktion übernehmen. Gleichzeitig betreut der Assistent erste Semester- und Diplomarbeiten.
- Phase II: Der Assistent führt eine kleine Projektgruppe. Dazu können sich auch AGP eignen, indem bei einer etwas anspruchsvolleren Aufgabe der Assistent mit mehreren Studenten zusammen das Projektteam bildet. Er übernimmt dabei eine Führungsaufgabe und erwirbt so selbst wesentliche Erfahrungen im Führen einer Gruppe. Dabei stehen ihm die AGP-Betreuer hilfreich zur Seite.
- Phase III: Der Assistent betreut ein AGP, nunmehr als Mentor, die Gruppe von Studenten arbeitet als Team selbständig.

Ein solcher „Ausbildungsgang“ eines angehenden Betreuers dauert dann je nach Situation mindestens drei Semester und muss unter der Leitung eines Professors oder eines erfahrenen (Ober-)Assistenten stehen.

7.3. Ausbildung der Studenten

Wie bereits dargelegt, verfügen die Studenten vor allem für die Phase Detailspezifikation meist nicht über genügend praktisches Fachwissen. Es ist daher zu diskutieren, wie diese Lücken geschlossen werden können, ohne die Anforderungen zu hoch anzusetzen (so sollten keine allzu arbeitsintensiven Vorarbeiten gefordert werden). Als Möglichkeit steht hier z. B. die Veranstaltungsform *Seminar* zur Verfügung. Ein Seminar, das mit wissenschaftlichen Methoden beispielsweise kommerziell erhältlichen Software-Engineering-Werkzeuge untersucht, könnte die nötigen Grundlagenkenntnisse vermitteln, die dann im Rahmen der Projektarbeit praktisch angewendet werden könnten [Sommerville 89, Gane 89, Schönthaler 90, Vetter 90, Bischofberger 89]. Ein erster Schritt in diese Richtung wurde an der ETH Zürich im Wintersemester 93/94 mit dem Seminar „Projekte, Methoden, Werkzeuge“ bereits unternommen.

Was die mangelnde Erfahrung und Sachkenntnis in Gesprächsführung und Zeitplanung anlangt, wäre ein Einführungskurs (z. B. in der letzten Woche des Vorsemesters) denkbar, der einige grundlegende Arbeitstechniken vermittelt. Im Rahmen eines solchen Kurses könnte man sich auf einige wesentliche Hilfsmittel, deren Anwendung aber grossen Nutzen bringen (Traktandenliste, Dokumentenbudget, Zeitplan) konzentrieren, so dass im Verlaufe der eigentlichen Projektarbeit aufgrund solcher Probleme weniger Zeit verlorenggeht, als dies bisher der Fall war. Es bleibt auch zu überdenken, ob den Studenten allenfalls striktere Vorgaben für die Organisation ihrer Arbeit gemacht werden sollen (z. B. keine Besprechung ohne Traktandenliste, Vorgabe der Darstellung der Zeitpläne usw.).

7.4. Weiterführende Ideen

Trotz der bereits sehr umfangreichen Zielsetzungen, die mit den AGP erreicht werden sollten, bleiben einige Wünsche offen. So wäre es insbesondere sehr wünschenswert, die Studenten für Fragestellungen zu sensibilisieren, die im Zusammenhang mit der Wartung [Lehner 91] und der Wiederverwendung [Krueger 92] von Komponenten einer Informatik-Anwendungen auftauchen. Ein weiteres, inskünftig immer wichtiger werdendes Gebiet der Informatikpraxis betrifft die Systemablösung; solche Fragestellungen, unter Einschluss von Reverse- und Reengineering-Aspekten [Arnold 92], konnten im Rahmen der AGP (bisher) nicht vermittelt werden. Es wäre allerdings denkbar, dass im Rahmen eines AGP nicht nur eine neue Anwendung entwickelt, sondern an einer existierenden Informatiklösung Änderungen / Erweiterungen vorgenommen oder aber sie ganz oder teilweise abgelöst oder auf eine andere Umgebung portiert würde. Hier wären allerdings die Betreuer herausgefordert, eine entsprechende Aufgabenstellung attraktiv zu präsentieren, um geeignete Interessenten zum Mitmachen zu motivieren (bei Studenten ist es eher unbeliebt, die Arbeit anderer weiterzuführen; sie entwickeln lieber alles selbst von vorn...).

Es stellt sich auch grundsätzlich die Frage, ob die angestrebten Ziele allenfalls im Rahmen einer Lehrveranstaltung für *alle* Studenten erreicht werden können. Entsprechende Erfahrungen anderer Hochschulen (die Universität Wien führt z. B. solche Lehrgänge

mit 450 Studenten durch [Biffel 93]) sind hier seriös zu studieren. In einem solchen Falle müssten alle Gruppen dieselbe Aufgabe bearbeiten, einerseits um vergleichbare Ergebnisse zu erhalten, andererseits aber auch um den Betreuungsaufwand in Grenzen zu halten. Auch in diesem Falle wäre eine gründliche Ausbildung der Betreuer natürlich unabdingbar.

Literatur

[Arnold 92]

Robert Arnold. Re-Engineering. *IEEE Computer Society Press*, Los Alamitos, California, 1992.

[Biffel 93]

Stefan Biffel und Thomas Grechenig. Erfahrungen mit Entwicklung in der Lehre und deren Konsequenzen. In H. C. Mayr und R. Wagner, Hrsg.: *Methoden für Informationssysteme*. Tagung, Klagenfurt, Juni 1993. pp 123–134

[Bischofberger 89]

W. Bischofberger und R. Keller. Enhancing the Software Life Cycle by Prototyping. *Structured Programming*. No 1, 1989, pp 47–59

[Boehm 86]

Barry W. Boehm. A Spiral Model of Software Development and Enhancement. *ACM SIGSOFT*, Vol 11, No. 4, 1986, pp 22–42

[Brooks 91]

Persönliche Gespräche zwischen F. P. Brooks und C. A. Zehnder, Juli 1991.

[Denert 92]

Ernst Denert. *Software-Engineering*. Springer, Berlin, 2. Auflage, 1992.

[ETH 89]

Studienplan 1989 der Abteilung für Informatik. Eidgenössische Technische Hochschule Zürich, September 1989.

[ETH 91]

Wegleitung für die Studierenden der Abteilung für Informatik. Eidgenössische Technische Hochschule Zürich, Sommer 1991.

[Gane 89]

Chris Gane. *Rapid System-Development*. Prentice Hall, Englewood Cliffs, 1989.

[Hallmann 90]

Matthias Hallmann. *Prototyping komplexer Softwaresysteme*. B. G. Teubner, Stuttgart, 1990.

[Krueger 92]

Charles W. Krueger. Software Reuse. *Computing Surveys*, Vol 24, No 2, 1992, pp 131–183.

[Lehner 91]

Franz Lehner. *Softwarewartung*. Hanser, München, 1991.

[Oertly91]

Walter Frederic Oertly. *Evolutionäre Prototypenbildung für Datenbank-Anwendungskomplexe*. Dissertation, ETH Zürich, Institut für Informationssysteme, 1991.

[Oesterle 92]

H. Oesterle und T. Gutzwiller. *Konzepte angewandter Analyse- und Design-Methoden*. AIT, Hallbergmoos, 1992.

[Pomberger 91]

W. Bischofberger und G. Pomberger. *Software Development*. Springer, Berlin, 1991.

[Schulz 89]

Arno Schulz. Software-Lifecycle- und Vorgehensmodelle. *Angewandte Informatik*, April 1989, pp 137–142.

[Schönthaler 90]

Schönthaler/Németh. *Software-Entwicklungswerkzeuge: Methodische Grundlagen*. B. G. Teubner, Stuttgart, 1990.

[Sommerville 89]

I. Sommerville. *Software-Engineering*. Addison-Wesley, Reading Massachusetts, 3. Auflage, 1989.

[Vetter 90]

Max Vetter. *Strategie der Anwendungssoftware-Entwicklung*. B. G. Teubner, Stuttgart, 2. Auflage, 1990.

[Yourdon 92]

Edward Yourdon. *Moderne Strukturierte Analyse*. Prentice Hall, Englewood Cliffs NJ, 1992.

[Zehnder 89]

C. A. Zehnder. *Informationssysteme und Datenbanken*. Verlag der Fachvereine, Zürich und B.G. Teubner, Stuttgart, 5. Auflage, 1989.

[Zehnder 91]

C. A. Zehnder. *Informatik-Projektentwicklung*. Verlag der Fachvereine, Zürich und B.G. Teubner, Stuttgart, 2. Auflage, 1991.

Teil 3

Zur Arbeit in Projektgruppen

C. A. Zehnder

1. Der Aspekt Zeit

In der Projektarbeit geht es grundsätzlich darum, mit beschränktem *Aufwand* und innert nützlicher *Zeit* (Termin) ein bestimmtes Ergebnis zu erreichen. Bei vielen Projekten und namentlich im Software-Bereich bildet die menschliche Arbeit über eine bestimmte *Zeitdauer* die wichtigste Aufwandskomponente. Das Stichwort *Zeit* tritt somit bei der Projektarbeit in zwei unterschiedlichen Rollen auf:

- *Termine*: Ein Projekt durchläuft im Laufe der *Zeit* verschiedene charakteristische Termine (Beginn, Ende, Meilensteine usw.). *Masseinheit*: Datum, allenfalls Tageszeit.
- *Zeitaufwand*: Von verschiedenen beteiligten Personen wird ein persönlicher Arbeits-einsatz geleistet, der gesamthaft als *Zeitaufwand* bezeichnet wird. *Masseinheit*: Personenstunden (bzw. Personentage, -wochen, -monate, -jahre). In kommerziellen Situationen lässt sich der *Zeitaufwand* auch in Geldwerten (Lohn und Arbeitsplatzkosten) ausdrücken.

Termineinhaltung und Aufwandbeschränkung spielen nicht in allen Projekten die gleich zentrale Rolle. In Entwicklungsabteilungen von Firmen, die ihre Produkte (= Ergebnisse der Projektarbeit) auf dem Markt anbieten, wird mit genauen Termin- und Kostenvorgaben gearbeitet, während in der Grundlagenforschung derartige Vorgaben selten möglich sind und sich der Erfolg oft erst nach geduldigem und wiederholtem Suchen einstellt. Die nachfolgenden Betrachtungen beziehen sich primär auf *Entwicklungsprojekte* (nicht auf Forschungsprojekte) und damit auf die Vorbereitung von Studierenden auf die künftige Industriepraxis (viele der Überlegungen gelten dennoch für jede Art von Projektarbeit).

Studierende neigen dazu, sich ihren persönlichen Zeitplan von aussen (Schule, Professor, Übungsaufgabe) vorgeben zu lassen; sie sind es auch gewohnt, klassenweise mit vielen Kolleginnen und Kollegen parallel und trotzdem allein die *gleichen* Übungsaufgaben zu lösen. Diese beiden Erfahrungen führen bei vielen Studierenden zu einer äusserst fatalistischen Haltung gegenüber ihrer persönlichen Zeitplanung: Sie warten auf Anstösse von aussen. Die wenigen selbständigen Arbeiten und Studienprojekte, denen ein „Standardstudent“ zwischen Einschulung und Hochschulabschluss begegnet, ändern an dieser Haltung im allgemeinen wenig. Im späteren Berufsleben von Informatikern bilden aber Projekte eine zentrale Form der Berufstätigkeit. Dabei ist eine *aktive* Termin- und Aufwandplanung nötig. Um das Termin- und Aufwandbewusstsein zu fördern, werden die Informatikstudierenden an der ETH Zürich im Rahmen von zwei Semesterarbeiten (je 150 Arbeitsstunden, davon eine projektorientiert in Informatik) und einer Diplomarbeit (Vollzeittätigkeit über ganze 4 Monate) dazu angehalten, ihre *Zeit* haushälterisch einzusetzen.

Noch wichtiger sind Termin- und Aufwandsplanung aber bei *Gruppenprojekten*, wie sie bei uns für Vierergruppen systematisch angeboten werden. Absolviert nämlich eine

Gruppe von vier Studierenden eine Semesterarbeit gemeinsam, so stehen für das Projekt rein rechnerisch 4×150 Stunden, gesamthaft also 600 Personenstunden zur Verfügung. Diese 600 Personenstunden sind aber etwas völlig anderes, als wenn eine einzige Person 600 Arbeitsstunden für ein Projekt einsetzen könnte. Die Gruppenarbeit konsumiert nämlich einen nicht zu vernachlässigenden Teil dieser Bruttokapazität für die interne Information und Koordination. Koordinationsbesprechungen sind eine aufwendige Sache. Fig. 1 zeigt das unmittelbar:

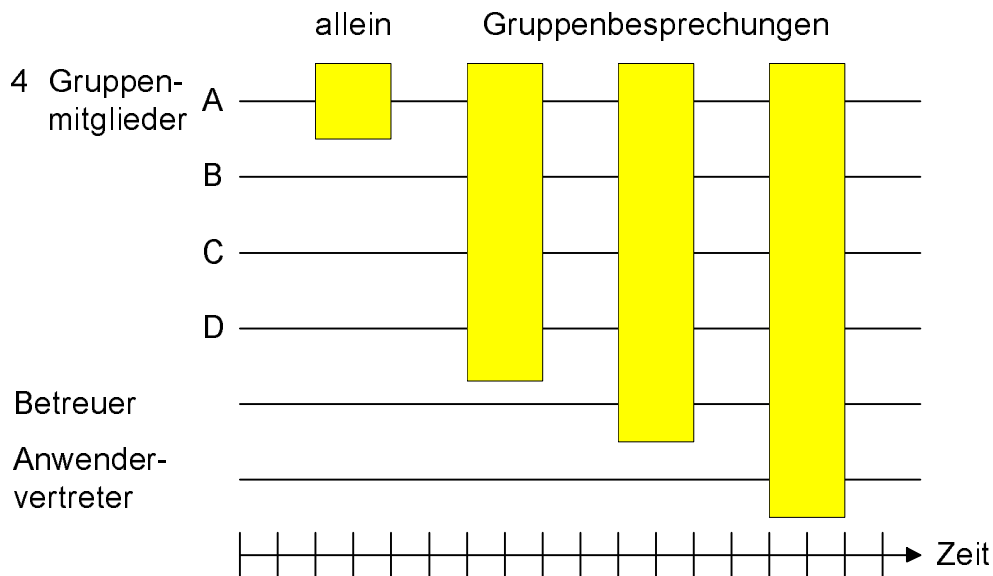


Fig. 1: Jede gemeinsame Sitzung kostet ein Vielfaches an Zeitaufwand

Jede Besprechung zur gruppeninternen Koordination erfordert hier mehrfachen Zeitaufwand; sie muss daher jedesmal sorgfältig vorbereitet, geleitet und bewusst kurz gehalten werden (zum Problem der Leitung siehe Ziffer 2).

Wer zu Sitzungen noch weitere Leute einlädt (Fig. 1 rechts: mit Betreuer, Anwender), belastet eine Sitzung zusätzlich: Auch diese benötigen Sprechzeit; dazu belasten sie irgendwo eine Lohnabrechnung! Daher Zurückhaltung mit Sitzungen und Teilnehmerlisten!

2. Die interne Struktur der Projektgruppe

In professionellen Entwicklungsabteilungen von Informatikbetrieben werden Projektgruppen normalerweise so zusammengesetzt, dass zum mindesten einzelne Gruppenmitglieder über *Projekterfahrung* verfügen; die Gruppenleitertaufgabe übernimmt eine dafür kompetente Person im Auftrag oder kraft natürlicher Autorität. Studentengruppen verfügen aber typischerweise *nicht* über derartige Chefpersönlichkeiten. Ihre Gruppenmitglieder waren ja bisher „gleichgeschaltet“; sie hatten alle den gleichen Stundenplan, lösten die gleichen Aufgaben, waren vor allem Zuhörer.

Diese über viele Jahre eingefahrene Rolle ist den meisten Studierenden so selbstverständlich, und jede Streberei ist hierzulande derart verpönt, dass zu Beginn einer Gruppenarbeit die meisten Studentengruppen sich sehr passiv verhalten und dabei viel Zeit verlieren. Niemand ergreift von selbst die Initiative; man macht anfänglich alles gemeinsam und wartet lieber ab, bis sich im gruppenspezifischen Spiel ein „Chef“ her-

ausbildet, der anschliessend die Leitungsaufgaben übernimmt. Dieses Spiel kann aber viele Stunden - mal vier - beanspruchen, die später bei der Projektarbeit fehlen. Daher muss durch geeignete Massnahmen *von aussen* dafür gesorgt werden, dass *sehr rasch* nach Projektbeginn die notwendigen Führungsaufgaben erkannt und durch einzelne Gruppenmitglieder übernommen werden und dass dann die übrigen Gruppenmitglieder diesen Führungsanspruch respektieren.

Der schnellste und eindeutigste Weg zur Projektleiterbestimmung ist die direkte Ernennung durch die Betreuer des Studentenprojektes (also durch Professor/Assistent); diese Form wäre aber für die Studierenden einmal mehr fremdbestimmt und daher unerwünscht. Eine weniger autoritäre Form könnte darin bestehen, dass bloss die rasche Bezeichnung eines Chefs verlangt, die Wahl aber der Gruppe überlassen würde. Auch diese Lösung würde aber den Gruppenmitgliedern den falschen Eindruck vermitteln, dass in einer Projektgruppe grundsätzlich straff hierarchisch geführt würde. Dabei weiss jedermann mit Projekterfahrung, dass in einer guten Projektgruppe nicht nur „von oben“ befohlen wird, sondern dass alle Gruppenmitglieder ihre spezifischen Beiträge an Wissen, Erfahrung und Persönlichkeit einbringen können und müssen, weil dies zu echten Synergien führt.

Erwünscht ist daher eine offenere Arbeitsstruktur, die zwar Effizienz vermittelt, aber auch die Initiative aller anregt. Dazu soll von den Gruppen nicht die Bezeichnung *eines* Chefs verlangt werden, sondern *mehrerer* Funktionsträger, die alle *auch* Chefaufgaben erhalten. Dieses Vorgehen passt übrigens sehr gut zur Situation der vier bisher gleichgestellten Studierenden.

Wie lassen sich aber vier sinnvolle, *unterschiedliche* Funktionen für die Mitglieder einer Studenten-Projektgruppe definieren? Der Autor dieses Texts verdankt die nachstehend (Fig. 2) skizzierte Aufgabengliederung einer persönlichen Mitteilung von Frederick Brooks, der damit selber an der University of North Carolina in Chapel Hill schon vieljährige gute Erfahrungen gemacht hat. Bei diesem Funktionsstrukturierungsansatz wird davon ausgegangen, dass sich in jedem Projekt zwei selbständige Führungsaufgaben dadurch ergeben, dass

- ein *Produzent* für die Bereitstellung aller notwendigen Rahmenbedingungen und Mittel sorgt, und
- ein *Chefingenieur* für die technische und anwendungsbezogene Qualität des zu entwickelnden Produktes geradesteht.

(Brooks vergleicht diese Funktionen mit der Herstellung eines Kinofilms: Dort stellt der Produzent den Rahmen bereit, und der Regisseur ist für das künstlerische Werk verantwortlich). Selbstverständlich sind die zwei Gruppenmitglieder, welche die zwei genannten Funktionen übernehmen, damit noch nicht ausgelastet. Sie setzen den noch verfügbaren Teil ihrer Kapazitäten zusammen mit den übrigen Gruppenmitgliedern als Entwicklungsingenieure ein. Damit haben wir die restlichen Funktionen: Ingenieure.

- Jeder *Ingenieur* übernimmt selbständig Teilaufgaben im Projekt. Beispiele: Abklärungen, Konzeption, Programmierung, Tests, Dokumentation (dabei ist es wichtig, dass eine Komponente nicht durch die gleiche Person konstruiert und getestet wird). Dazu kommen allenfalls Koordinationsaufgaben für das Gesamtprojekt (z. B. Controlling gemäss Ziffer 4).

Führungsaufgaben für das Gesamtprojekt

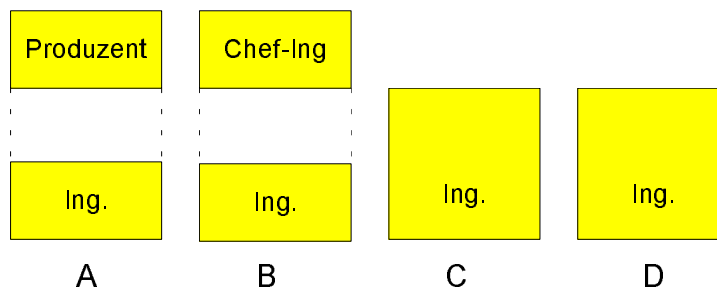


Fig. 2: Mögliche Strukturierung einer Projektgruppe aus vier Mitgliedern gleicher Ausbildungsstufe.

Wenn ein Betreuer der Gruppe diese mögliche Aufgabenverteilung vor Projektbeginn vorstellt und „etwas Derartiges von der Gruppe rasch erwartet“, so wird die Gruppe dadurch im Normalfall genügend angeregt, aktiv zu werden, sich zu strukturieren und damit zur konstruktiven Projektarbeit überzugehen. Im Hintergrund lässt sich dazu durch die Betreuer auch Hilfe (nicht Besserwisserei!) anbieten, und wenn die Gruppe aus irgendwelchen Gründen selber andere Funktionen und Strukturen oder im Laufe der Projektarbeiten personelle Umbesetzungen vorschlägt, so kann dem im allgemeinen problemlos entsprochen werden.

3. Die Projektgruppe und ihre Partner

Wichtig ist, dass die Projektgruppe jederzeit spürt (notfalls zu spüren bekommt), dass einzig sie für das Durchziehen des Projektes voll verantwortlich ist, während alle übrigen vom Projekt tangierten Personen (hier „Partner“ genannt) sich nur subsidiär mit dem Projekt befassen und sonst mit anderen Aufgaben beschäftigt sind.

- Aktives Element: Projektgruppe
- Passive Elemente: verschiedene Partner, namentlich Auftraggeber, Anwendervertreter, Betreuer.

Die Zusammenarbeit der Projektgruppe mit ihren Partnern ist nicht immer problemlos; bei Studentenprojekten kommt hinzu, dass die Partner meist mehr Erfahrung und Sachkenntnis als die Projektmitarbeiter aufweisen. Die Projektgruppe fühlt sich daher von ihren Partnern bald einmal kontrolliert oder bedrängt; sie empfindet die Meinungsäußerungen von Partnern als Bremsklötze oder Vorwürfe und antwortet darauf - je nach Persönlichkeitsstruktur und Stimmungslage der Gruppe - mit Nichtzurkenntnisnehmen, Rechtfertigung oder gar Trotzreaktionen. Diese Reaktionen sind bei Gruppenprojekten meist stärker als bei Einzelprojekten, weil die Gruppenmitglieder sich gegenseitig in ihrer Haltung unterstützen. Solche Reaktionen entstehen sehr direkt aus typischen Verhaltensstrukturen von Studierenden im Grundstudium; sie belasten gelegentlich die Projektarbeit wesentlich.

Zur Betreuung von Projektarbeiten von Studierenden gehört daher auch die Anleitung zum positiven Umgang mit Meinungsäußerungen und Beiträgen von Partnern. Diese sollten grundsätzlich *konstruktiv* verwertet werden:

- Einwände von Partnern dienen als Prüfmarken für das in Vorbereitung befindliche Projektergebnis (Produkt). Solche Prüfmarken helfen, wesentliche Fehler zu vermeiden, besonders wenn sie schon früh zur Kenntnis genommen werden können (weil die Projektgruppe darauf zu hören bereit ist).
- Alternativvorschläge von Partnern können die eigene Variantensuche (siehe Ziffer 4) wesentlich erleichtern. Sie zeigen aber auch, was der Partner eigentlich anstrebt, und sollten daher ebenfalls aktiv zur Kenntnis genommen werden (auch wenn am Schluss nicht alles berücksichtigt werden kann).

Zur Art des Umganges mit den Partnern gehören auch die äusseren Umgangsformen; namentlich eine sachliche Höflichkeit. Nicht alle Studierenden haben darin Erfahrung (besonders nicht schriftlich!). Auch hier können Betreuer aus dem Hintergrund eine nützliche Hilfe leisten, die meist sofort und gerne angenommen wird.

4. Variantendenken und Konzentration auf das Wesentliche

Der Autor dieses Textes ermuntert seine Studierenden regelmässig, sich bei der Projektarbeit nicht zu früh auf eine einzige Lösungsmöglichkeit zu beschränken, sondern namentlich in der Konzeptphase bewusst mehrere sinnvolle Varianten einzubeziehen und dann erst die Bestgeeignete auszuwählen und weiterzuverfolgen [Zehnder 91, Phase „Konzept (mit Varianten)“].

Andererseits gilt aber ständig auch die Forderung nach der Konzentration der Projektarbeit auf das Wesentliche [Zehnder 91, „80-20-Regel“]. Das ist schon im Normalfall wichtig, besonders aber unter Zeitdruck; Termin- oder Aufwandprobleme müssen primär durch Verzicht auf Kompliziertes und Zusätzliches gelöst werden.

Die Projektarbeit zwischen Variantendenken und ständigem Suchen nach Vereinfachungen ist eine Gratwanderung. Diese wird noch schwieriger, wenn nicht eine Einzelperson, sondern eine ganze Gruppe in der Arbeit involviert ist, weil oft die verschiedenen Varianten durch verschiedene Gruppenmitglieder ausgearbeitet werden. Ihre Ausarbeitung erfolgt leicht allzu fantasie reich und ihr Vergleich wird erschwert,.

Es ist somit gerade bei Gruppenprojekten eine zentrale Aufgabe, ständig und konsequent den Projektrahmen im Auge zu behalten und alle Auswüchse rasch zu beenden. Diese Aufgabe kann der Chefingenieur oder aber ein eigentlicher *Controller* wahrnehmen. So könnte neben Produzent und Chefingenieur ein drittes Gruppenmitglied mit einer globalen Projektaufgabe betraut werden.

5. Als Zusammenfassung: Einige Organisationsregeln

- Alle Besprechungen sind durch einen Verantwortlichen sorgfältig vorzubereiten.
- Interne Koordinationssitzungen sind periodisch und bei Entscheiden nötig; gerade sie erfordern aber einen thematisch und zeitlich straffen Arbeitsstil.
- Zu Besprechungen sind nur die wirklich nötigen Personen beizuziehen. Verlangen verschiedene Themen unterschiedliche Teilnehmer, so sind Besprechungen entsprechend aufzuteilen.
- Alle Einzelpersonen in einer Projektgruppe sollen individuelle Teilarbeiten übernehmen.

- Aufgaben, die von Einzelnen erledigt werden können, sollen rasch zugeteilt werden.
- Kommentare und Einwände von externen Partnern (Auftraggeber, Anwender, Betreuer) zur Projektarbeit sind vorurteilslos aufzunehmen und zu verarbeiten. Oft können sie nutzbringend in der Projektarbeit verwertet werden.
- Das Ziel eines Projektes darf auch in der Hitze des Gefechtes nie aus den Augen verloren werden: Konzentration auf das Nötige und Machbare.

Literatur

[Zehnder 91]

C. A. Zehnder. *Informatik-Projektentwicklung*. Verlag der Fachvereine, Zürich und B.G. Teubner, Stuttgart, 2. Auflage, 1991.

Teil 4

Das Programm „Zeitplan“ - Benutzerdokumentation

D. Aebi

1. Einführung

Wer professionelle Projektarbeit leistet, sollte jederzeit über folgende Werte den Ueberblick haben:

- *Aufwand* dieser Projektarbeit, gemessen in Personenstunden (oder Personentagen, Personenwochen, Personenmonaten, Personenjahren).
- *Termine*, angegeben als Datum (allenfalls auch Tageszeit).

Beide Arten von Werten werden zu Beginn der Projektarbeit geschätzt (*SOLL-Werte*), während des Projektfortschrittes aber laufend aufgezeichnet (*IST-Werte*) und dabei auch mit den SOLL-Werten verglichen. Die Aufzeichnung der IST-Werte ist bei professioneller Projektarbeit zwingend, sie liefert die Grundlagen für Fakturierung und Kostenrechnung.

Der SOLL-IST-Vergleich dient:

- der frühzeitigen Orientierung aller Interessierten über allfällige Abweichungen während der Projektarbeit;
- der Selbsterkenntnis von Projektleitung und Projektmitarbeitern betreffend ihre Fähigkeiten zu angemessenen Schätzungen.

Studierende sollten jede Gelegenheit nutzen, erste Erfahrungen in Projektarbeit zu sammeln. Dazu eignen sich namentlich auch ihre Semester- und Diplomarbeiten, wo sie nicht nur technisches Wissen einbringen, sondern auch Aufwand und Termine unter Kontrolle halten müssen. Bei diesen Ueberwachungsaufgaben kann ein einfaches grafisches Werkzeug helfen, wie es das Programm „Zeitplan“ darstellt.

„Zeitplan“ ist auf studentische Arbeiten ausgerichtet:

- Der Aufwand wird in Personenstunden gemessen.
- Termine werden nur wochengenau angegeben.
- Die Gesamtprojektdauer darf 30 Wochen nicht überschreiten.
- Die Zahl der Projektkomponenten (Arbeitspakete und Meilensteine s.u.) darf 20 nicht überschreiten.

In diesem Rahmen produziert das Programm automatisch Zeitpläne, wie sie in Fig. 7, 8 und 9 dargestellt werden. „Zeitplan“ ist erhältlich in Versionen für DOS/Windows und für Macintosh; Studenten können das Programm selbständig vom FTP-Server (<ftp.inf.ethz.ch>) des Departementes kostenlos herunterladen.

2. Begriffe - Konzept

Das Programm Zeitplan verwendet die folgenden Begriffe:

- **Projekt:** Dieses wird in verschiedene Arbeitspakete (AP) aufgeteilt; zusätzlich lassen sich verschiedene Meilensteine (MS) angeben.
- **Stammdaten:** Daten, die ein Projekt als Ganzes betreffen (Projektbezeichnung, Beginn, Ende, Anzahl Arbeitsstunden).
- **Arbeitspaket (AP):** Teil eines Projektes („Aufgabe“), der eine Einheit bildet und einer Gruppe von Personen oder einer Einzelperson zur Bearbeitung zugeteilt wird.
- **Meilenstein (MS):** Zeitpunkt, zu dem eine bestimmte Handlung (Dokument, Aufgabe) fertiggestellt sein muss.
- **Projektkomponente:** Oberbegriff für AP und MS zusammen.

Zu jeder Projektkomponente werden SOLL- und IST-Daten verwaltet. Die grafische Darstellung der Zeitverhältnisse kann auf drei Arten erfolgen:

1. nur SOLL-Daten (Beispiel: Fig. 7)
2. SOLL- und IST-Daten, nur zeitlich (Beispiel: Fig. 8)
3. SOLL- und IST-Daten, mit Aufwandsangaben (Beispiel: Fig. 9)

Das Programm dient nur der Darstellung des zeitlichen Ablaufes eines Projektes (Balkendiagramm). Es werden keinerlei Abhängigkeiten zwischen einzelnen Projektkomponenten verwaltet. Das Programm ermöglicht es, in einer ersten Phase ein Projekt zu planen und in einer zweiten Phase die effektiven Aufwandsangaben (und damit den Projektstand) wochenbezogen festzuhalten. Dazu werden zuerst alle Projektkomponenten mit ihren SOLL-Daten erfasst („Planung“). Im Laufe des Projektes wird dann der effektive Zeitaufwand eingetragen (IST-Daten) ebenso die effektiven Termine der MS.

3. Funktionen des Programmes (entsprechend Menustruktur)

Hinweis: Es werden nur diejenigen „Zeitplan“-Funktionen erklärt, deren Bedienung nicht analog zu bekannten Standardfunktionen (Datei öffnen, Speichern, Drucken usw.) erfolgt. Das Aufrufen der einzelnen Programmfunktionen erfolgt über ein Menü, dessen Darstellung sich nach den üblichen Regeln der entsprechenden Plattform richtet. Die folgenden Ausführungen befassen sich vor allem mit Aspekten der Dateneingabe.

Projektdaten → Stammdaten

Über diese Dialogbox werden die Stammdaten eines Projektes eingegeben und allenfalls geändert.

Bezeichnung	Tutorial		
Zeit zur Verfügung (in h)	300	Zeit verbraucht (in h)	0
Projekt-Beginn	1.1.1994	Projekt-Ende	30.3.1994

Buttons: OK, Abbrechen

Fig. 1: Eingabe der Projektstammdaten

Die eingegebenen Stammdaten werden wie folgt überprüft:

- Die Bezeichnung (Projektname) darf nicht leer gelassen werden.
- Es werden nur gültige Datumswerte angenommen; Format (T)T.(M)M.(JJ)JJ. Ein Projektbeginn an einem Wochende wird automatisch auf den darauffolgenden Montag, ein Projektende an einem Wochenende auf den Freitag vorher verschoben.
- Die verfügbare Arbeitszeit muss grösser als Null sein.
- Die Gesamtdauer darf dreissig Wochen nicht übersteigen.

Die eingegebenen Stammdaten lassen sich nachträglich wie folgt ändern:

- Die Bezeichnung kann jederzeit geändert werden, ebenso die Arbeitszeit, die zur Verfügung steht.
- Der Projekt-Beginn kann nicht mehr verändert werden.
- Das Projekt-Ende kann vor- oder nachverlegt werden. Wird allerdings das Projektende vorverlegt, nachdem bereits Projektkomponenten eingefügt wurden, muss der Benutzer selber dafür sorgen, dass keine Inkonsistenzen entstehen (das Programm prüft dies nicht).

Projektdaten → Arbeitspakete SOLL...

Mit dieser Dialogbox werden die Soll-Daten zu einem Arbeitspaket eingegeben.

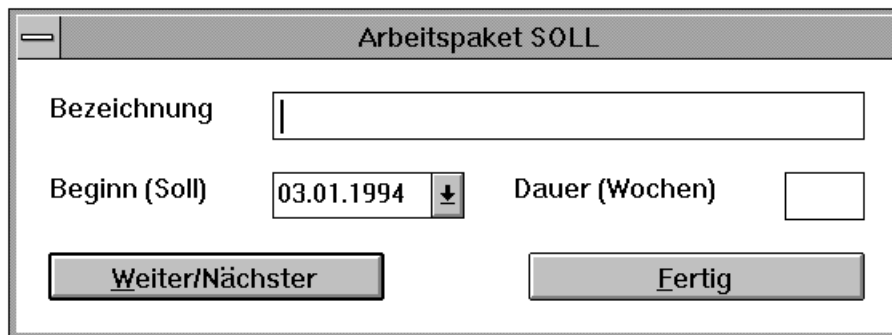


Fig. 2: Eingabe der SOLL-Daten zu einem AP

- Jedes Arbeitspaket muss einen eindeutigen Namen erhalten (wird bei der Eingabe geprüft).
- Der Beginn (Soll) muss aus einer Liste (bestehend aus Montagen, Ausnahme erste Projektwoche: effektiver Beginn) ausgewählt werden. Das Programm schlägt als Ersatzwert jenen Wochenbeginn vor, der direkt an das bisher späteste Ende eines schon eingegebenen AP anschliesst.
- Die Dauer (Wochen) muss grösser als Null sein.
- Auf Anklicken der Schaltfläche „Weiter/Nächster“ wird das beschriebene AP eingefügt; die Dialogbox ist wieder bereit zur Datenaufnahme für das nächste AP. Nach der letzten Eingabe kann die Dialogbox mit „Fertig“ verlassen werden.

Projektdaten → Meilensteine SOLL...

Mit dieser Dialogbox werden die Soll-Daten zu einem Meilenstein eingegeben.

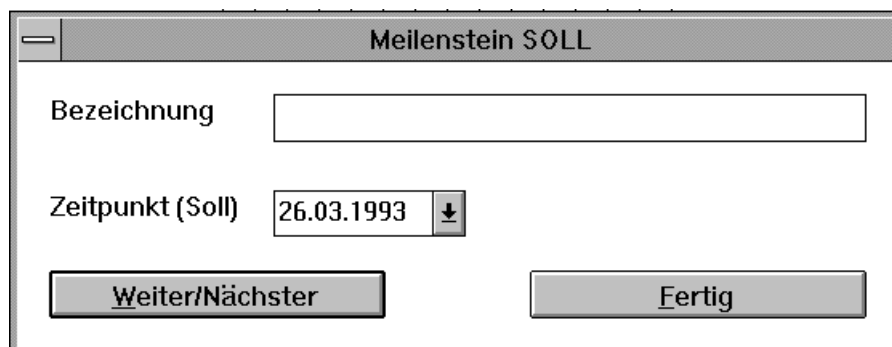


Fig. 3: Eingabe der SOLL-Daten zu einem MS

- Jeder Meilenstein muss einen eindeutigen Namen erhalten (wird bei der Eingabe geprüft).
- Der Zeitpunkt (Soll) muss aus einer Liste (bestehend aus Freitagen, Ausnahme letzte Projektwoche: effektives Ende) ausgewählt werden. Das Programm schlägt als Er-

satzwert jenen Freitag vor, der unmittelbar auf den spätesten bisher eingegebenen folgt.

- Auf Anklicken der Schaltfläche „Weiter/Nächster“ wird der beschriebene MS eingefügt. Die Dialogbox ist wieder bereit zur Datenaufnahme. Nach der letzten Eingabe kann die Dialogbox mit „Fertig“ verlassen werden.

Projektdaten → Arbeitspakete IST...

Über diese Dialogbox kann zu einem bereits existierenden SOLL-Arbeitspaket die zugehörige IST-Arbeitsleistung während einer bestimmten Kalenderwoche eingegeben werden. Die Verarbeitung erfolgt bewusst nach Wochen (und nicht nach AP), damit sich der Benutzer am Ende einer Woche Rechenschaft darüber abgibt, was er in dieser Woche für sein Projekt wirklich getan hat.

Woche	Arbeitspaket	Anzahl Personen-Stunden
01.01.1993	Projektidee	10
04.01.1993	Projektumriss	
11.01.1993	Konzept	
18.01.1993	Realisierung	
25.01.1993	Systemtest	

Fig. 4: Eingabe der IST-Daten zu einem AP

- Die Woche muss aus der präsentierten Liste ausgewählt werden.
- Das AP muss aus der präsentierten Liste aller SOLL-AP ausgewählt werden.
- Die Anzahl der effektiv geleisteten Arbeitsstunden ist einzugeben.
- Auf Anklicken der Schaltfläche „Weiter/Nächster“ werden die Daten übernommen, und die Dialogbox ist wieder bereit zur Datenaufnahme. Nach der letzten Eingabe kann die Dialogbox mit „Fertig“ verlassen werden.

Projektdaten → Meilensteine IST...

Mit dieser Dialogbox wird zu einem existierenden SOLL-MS der IST-Zeitpunkt eingegeben.

Meilensteine	Zeitpunkt (Ist)
Projektauftrag	29.01.1993
Projektentscheid	
Projektende	

Zeitpunkt (Soll)
22.01.1993

Weiter/Nächster Fertig

Fig. 5: Eingabe der IST-Daten zu einem AP

- Der MS muss aus der präsentierten Liste der SOLL-MS ausgewählt werden. Die automatische Anzeige des SOLL-Zeitpunktes dient als reine Orientierungshilfe.
- Der effektive Zeitpunkt ist einzugeben.
- Auf Anklicken der Schaltfläche „Weiter/Nächster“ werden die Daten übernommen; die Dialogbox ist wieder bereit zur Datenaufnahme. Nach der letzten Eingabe kann die Dialogbox mit „Fertig“ verlassen werden.

Projektdaten → Reihenfolge...

Mit dieser Dialogbox kann die Darstellungsreihenfolge der einzelnen Projektkomponenten in der Grafik geändert werden. Auch lassen sich damit vorhandene Projektkomponenten löschen.

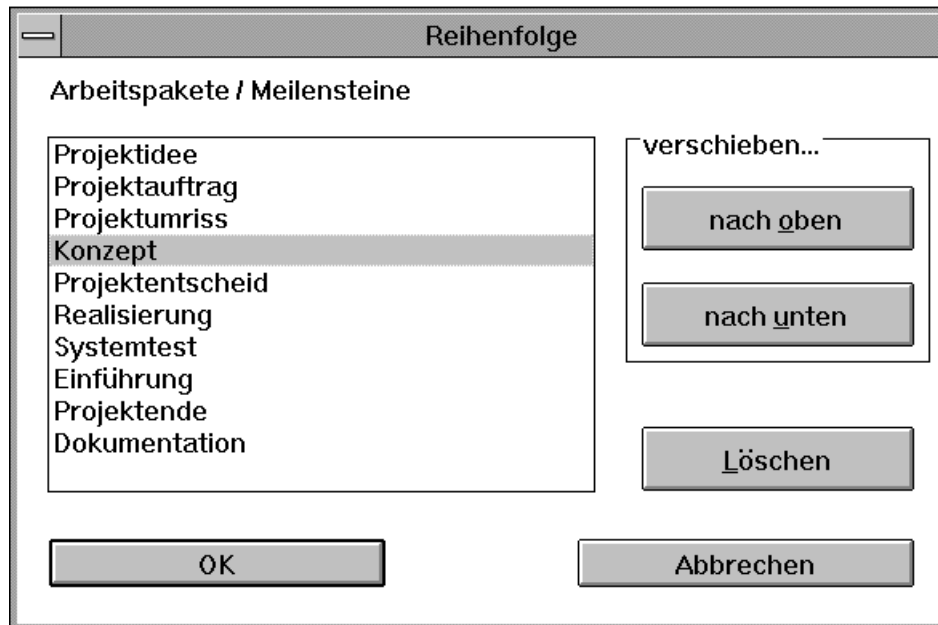


Fig. 6: Festlegen der Reihenfolge der Projektkomponenten

- Zuerst ist die Projektkomponente auszuwählen, die von der Umstellung oder Löschung betroffen ist.
- Zum Verschieben wird die ausgewählte Projektkomponente mit der Schaltfläche „nach oben“ bzw. „nach unten“ an die gewünschte Position verschoben.
- Gelöscht wird eine Projektkomponente durch Anklicken der Schaltfläche „Löschen“. Dieser Befehl löscht sämtliche zu dieser Komponente bereits eingegebenen Daten vollständig, also Bezeichnung, SOLL-Werte und vorhandene IST-Werte. Eine automatische Rekonstruktion dieser Werte ist nicht möglich.

Anzeige → Nur SOLL

Es werden nur die SOLL-Daten angezeigt (Beginn und Ende bei Arbeitspaketen, Zeitpunkt bei Meilensteinen).

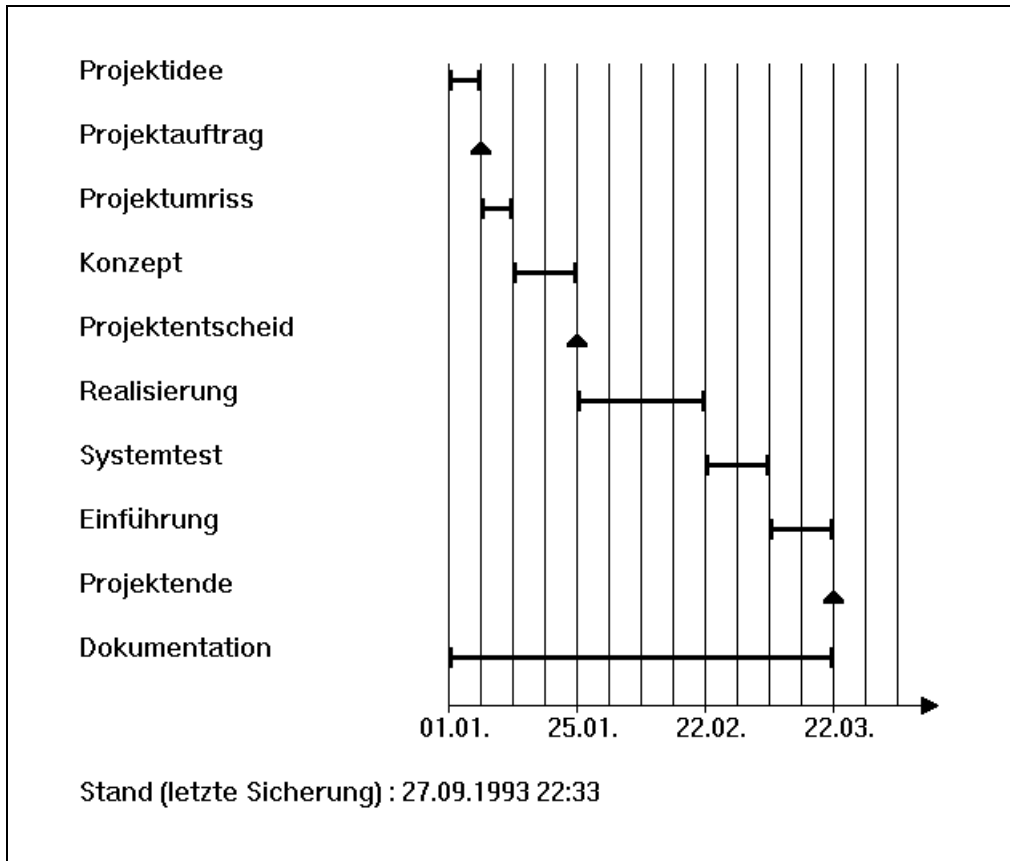


Fig. 7: Zeitplan (Solldaten von Projektkomponenten)

Anzeige → SOLL und IST

Es werden sowohl SOLL- als auch IST-Daten angezeigt, die IST-Werte allerdings nur zeitlich (als Balken), ohne Aufwandsangabe.

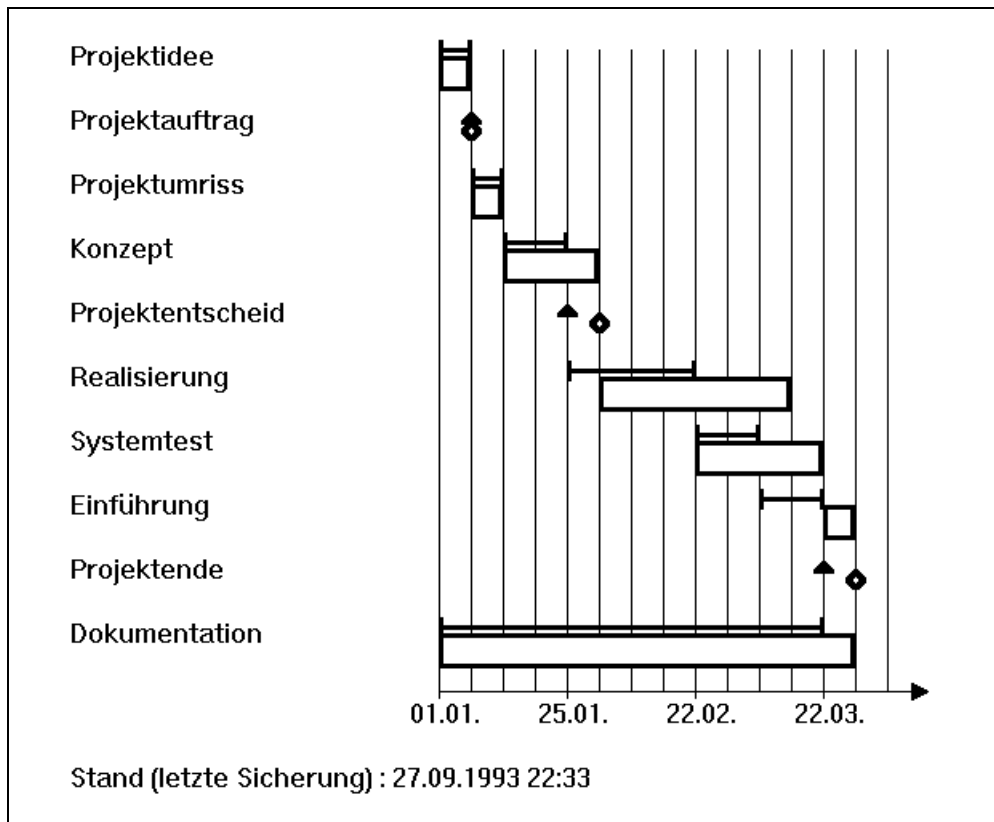


Fig. 8: Zeitplan (SOLL- und IST-Daten von Projektkomponenten)

Anzeige → SOLL und IST, Auslastung

Es werden sowohl SOLL- als auch IST-Daten angezeigt; bei den IST-Daten werden die geleisteten Personenstunden (Aufwand) wochenweise als Säulen dargestellt.

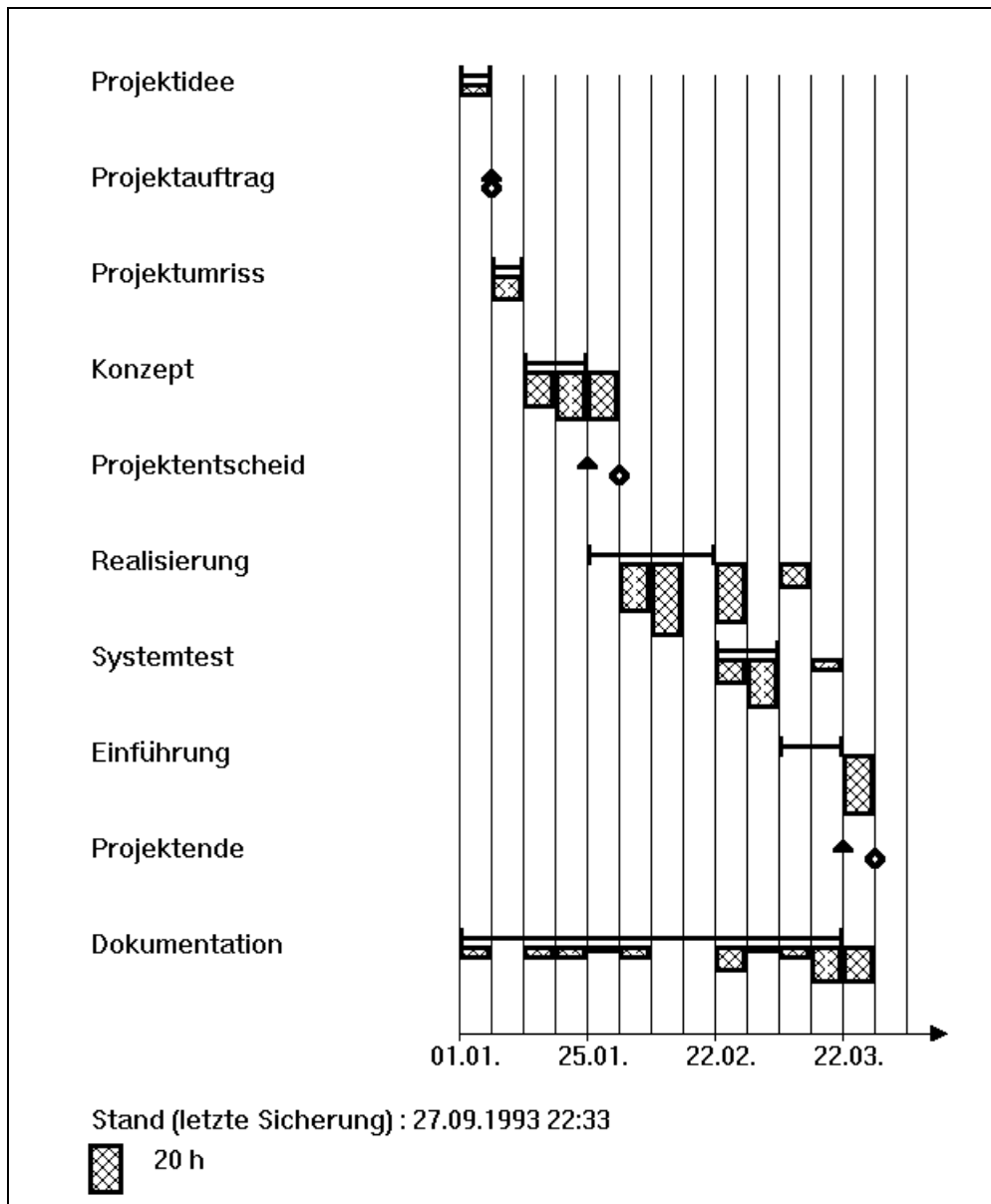


Fig. 9: Zeitplan (SOLL- und IST-Daten inkl. Aufwände)