
Softwareentwicklung II

Der Turm von Babel

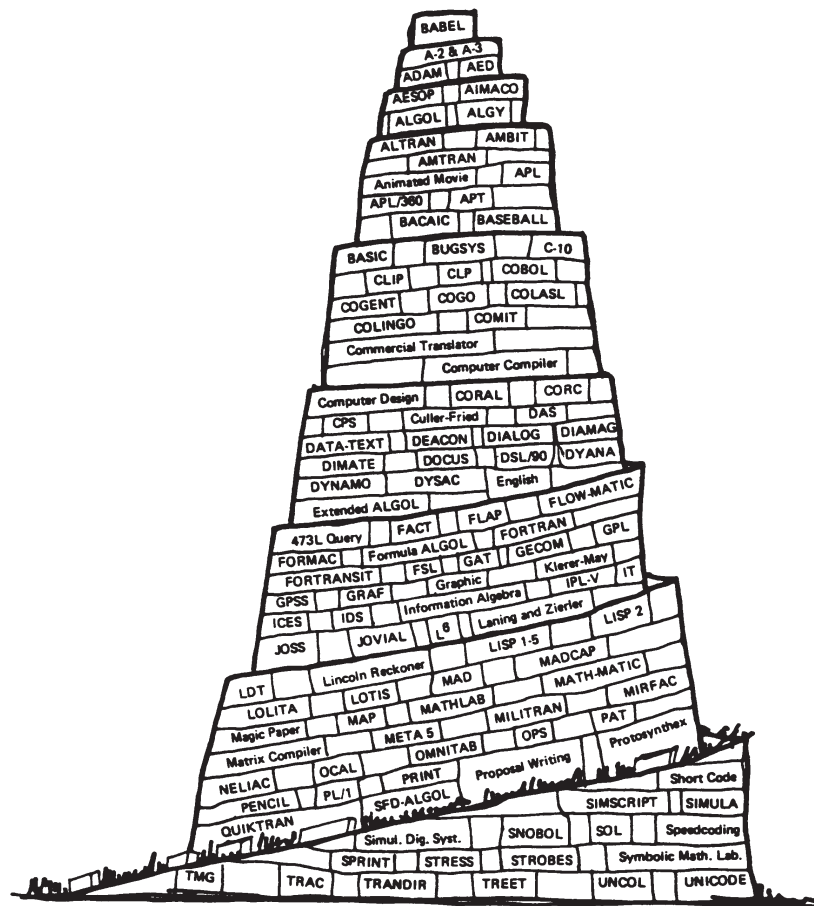
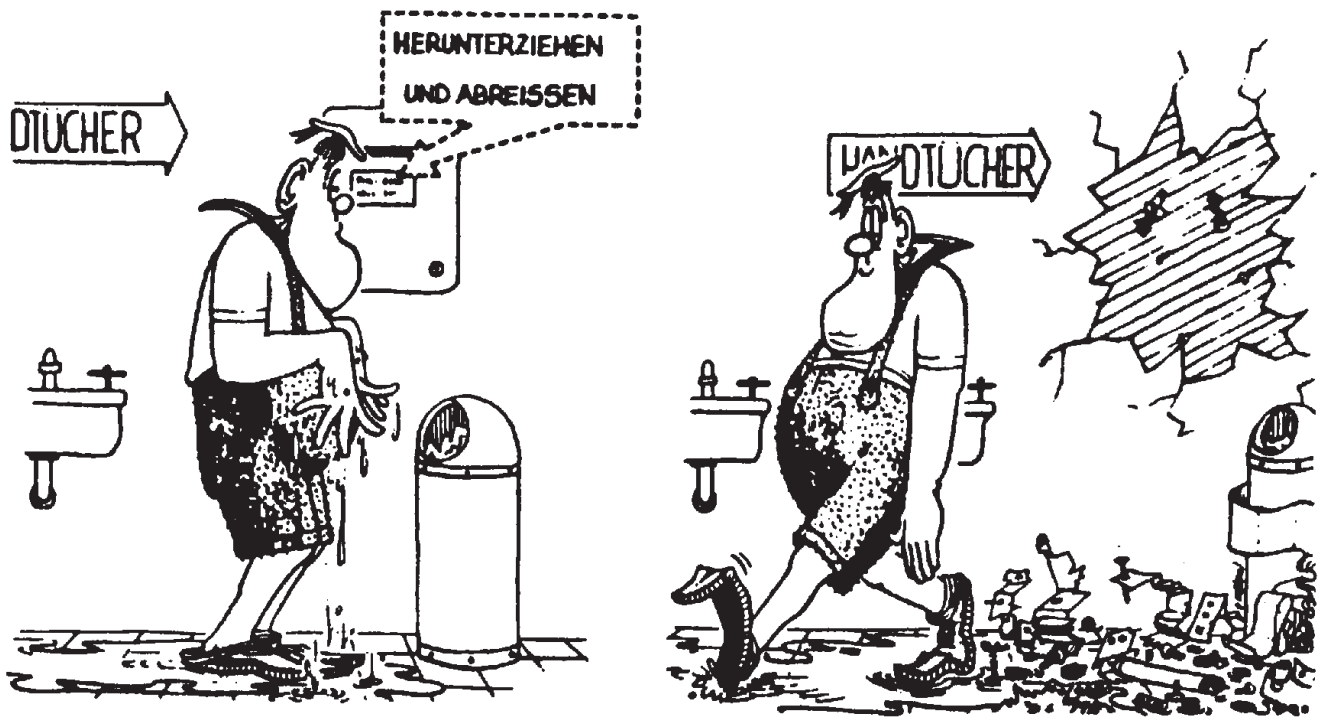


Abb. 1.1: Der Turm zu Babel aus Programmiersprachen
(aus Sammet, 1969)

WWZ-Produkte

- Bibliotheksverwaltung
- Trouble Shooter
- Studienberater
- CBT - Lineare Optimierung
- CBT - Netzplantechnik ...
- Internet-basierte Evaluation

Lernziele und Projektorganisation



Lernziele

- Phasen der Softwareentwicklung selbst erleben
- Gruppenarbeit

Projektorganisation

- Terminplan studieren → Merkblatt
- Projektaufgaben identifizieren → Folie/Merkblatt
- Betreuer vorstellen → Merkblatt
- Projektgruppen bilden → Folie

Projektaufgaben und Phasendokumente

1 Spezifikation

A11 Grobspezifikation (vorgegeben)

A12 Phasendokument **Feinspezifikation**

2 Entwurf

A21 Dialogentwurf

A22 Modulhierarchie

A23 Modulbeschreibungen

A24 Pseudocode

A25 Phasendokument **Feinentwurf**

3 Implementation

A31 Modulprogrammierung und -test

A32 Systemintegration

A33 Systemtest

A34 Phasendokument **System- und Benutzerdokumentation**

“Web Quiz” - Grobspezifikation

1999 erscheint beim Springer Verlag unser *Data Warehousing und Data Mining - Eine Einführung in Entscheidungsunterstützende Systeme*. Das Lehrbuch enthält eine CD ROM mit einer interaktiven Foliensammlung, Demonstrationssoftware sowie Beispielen und Übungen.

Spezifizieren, entwerfen und implementieren Sie eine Anwendung “Web Quiz”. Die Anwendung soll als Teil der CD dem Autor bzw. Leser die folgenden Möglichkeiten bieten:

- ✓ “Web Quiz” erlaubt dem Autor, die **Erstellung** objektiv korrigierbaren Aufgaben (Mehrfachwahl-, Zuordnungs- und Wahr/Falsch-Aufgaben)
- ✓ “Web Quiz” erlaubt ausserdem die interaktive **Bearbeitung** der Aufgaben auf dem World Wide Web. Der Benutzer (Leser) ...
 - kann jederzeit den Quiz (das Kapitel) wechseln
 - kann sich jederzeit über den Bearbeitungsstand orientieren
 - erhält nach jeder Antwort eine begründete Korrektur
- ✓ Die Anwendung benötigt nur einen verbreiteten Browser (MS Internet Explorer oder Netscape). Zusätzliche Software ist nicht erforderlich.

Verwenden Sie zur Implementation Software, welche die Mehrheit ihrer Gruppe bereits kennt. Wenn Sie über Programmiererfahrungen mit Visual Basic verfügen, verwenden Sie HTML und Visual Basic Script/Internet Explorer. Setzen Sie Java Script/Netscape nur dann ein, wenn Sie Java Script bzw. Java *gut* kennen.

Gruppenbildung und Aufgabe

Gruppenbildung

a) höchstens 5 Mitglieder

b) In jeder Gruppe ...

- mind. 1 mit vertieften Kenntnissen jedes Werkzeugs
- mind. mit aktivem Besuch der Einführung Glinz
- je 1 Verantwortlicher für jedes Phasendokument

Aufgabe für die Gruppensitzungen der 2. Woche

- Jede Gruppe teilt dem Betreuer je einen *Verantwortlichen* für jedes Phasendokument mit
- Jede Gruppe erstellt zuhanden ihrer Mitglieder und zuhanden des Betreuers eine Adress- und Telefonliste
- Jedes Gruppenmitglied studiert die Grobspezifikation und bringt einige Ideen zur Feinspezifikation mit

Software Development Cycle

Specification (WHAT?)

- ⇒ Current System (Ist-Analyse)
- ⇒ Proposed System (Soll-Analyse)

Realisation (HOW?)

- ⇒ Design
 - Preliminary Design (Grobentwurf)
 - User interface Design (Dialogentwurf)
 - Preliminary Module Design (Grobentwurf)
 - Detailed Design (Feinentwurf)
- ⇒ Implementation
 - Coding
 - Unit Testing
 - Integration
- ⇒ System Testing
- ⇒ Installation/Maintenance

Stage Independent Activities

Project Management (*Merkblatt*)

⇒ Products

⇒ Activities

⇒ Resources

- Time
- Money
- Personnel
- Hardware/Software

Quality Control (Evaluation)

⇒ Specification Walk Through

⇒ Design Walk Through

⇒ Code Walk Through

⇒ Test Walk Through

Meeting Management (M41)

- Group Leader
- Formal Communication Network (*Teilnehmerliste*)
e.g. telephone chain
- Regular Meeting Times (*Terminplanungsblatt*)
- Location
- Recorder (*Protokollführer fallweise*)

Specification Management (M89)

Sponsor Meetings

⇒ Questionnaire (*Fragebogen*)

- Who
- What
- When
- Where
- Why
- How

⇒ Documents Collection

Documentation Tools and Standards

Preliminary Specification (*Grobspezifikation*)

Goal Description (e.g. M397)

Nonfunctional Requirements (e.g. M402-403)

- ⇒ User Interface
 - Hardware/Software
- ⇒ Documentation
 - User -/Developer -
- ⇒ Performance
- ⇒ System Interface
 - Hardware/Software
- ⇒ Security

Detailed Specification (*Feinspezifikation*)

- ✓ Data Flow Diagram (*Datenflussdiagramm*)
- ✓ *Data Dictionary*
- ✓ Activity Hierarchy (*Tätigkeitshierarchie*)
- Feasibility Study (*Machbarkeitsstudie*)
- ✓ Sponsor's Sign Off



Aufgabe

- a) Bis zum 19.11. vervielfältigen
- b) Am 19.11. präsentieren

Eine einfache PDF-Lösung

Funktionsumfang

- ✓ Fragen auf PDF-Seiten
- ✓ Richtige Antwort durch einen Klick auf ein Note-Symbol
- ✓ Begründung durch Hyperlink auf den Folientext
- ✓ Speicherbare Annotationsmöglichkeit

Hauptnachteile

- Testaufgaben des Lehrbuches nicht integriert
- Geringe Interaktivität
- Keine Bewertungsbuchhaltung
- ...

Eine Lösung mit HTML und eingebetteten Scripts

Funktionsumfang

- ✓ Kapitelmenü
- ✓ Beantwortung durch Klick auf die gewählte Alternative
- ✓ Richtige Antwort in einem getrennten Frame
- ✓ Begründung (noch) nicht vorhanden
- ✓ Keine Annotationsmöglichkeit

Hauptnachteile

- Testaufgaben des Lehrbuches nicht integriert
- Nur Mehrfachwahlaufgaben implementiert
- Benutzerunfreundliche Autorenkomponente
- Keine Links zur Foliensammlung
- ...

Grundsätze zur Feinspezifikation

- Fahrtüchtiger VW statt Porsche ohne Fahrwerk
(nur *Minimalspezifikation* entwerfen und implementieren)
- Primat der Auskünfte des Lehrbuchautors
- *Einfrieren* einer *gemeinsamen* Feinspezifikation
- HELP statt ausgebaute Benutzerdokumentation

Preliminary Design (*Grobentwurf*)

User Interface Design (Dialogentwurf, M137)

- ✓ Current System Evaluation
- ✓ Potential User Description (e.g. novice/expert)
- ✓ Similar Systems Evaluation
- ✓ User Interface Prototype
- ✓ User Aids (Help, On Line/Off Line Tutorials)
- ✓ User Manual Draft (M Appendix B)

Preliminary Module Design (M188)

- ✓ Module Hierarchy (List or Chart)
- ✓ Module Headers (Modulköpfe)



Aufgabe: Mit dem Feinentwurf ...
a) bis zum 17.12. vervielfältigen
b) am 17.12. präsentieren

UI - User Interface Development Tools

Dialog-Simulatoren (Bsp. Dan Bricklin's Demo)

Programm- und Template-Generatoren

a) spezialisiert (Bsp. Paradox Personal Programmer)

Beschränkung auf eine Zielsprache

b) allgemein

Auswahl aus mehreren Zielsprachen

Programm-Bibliotheken (Toolboxes)

- Grafik
- Dialog (text- oder grafikorientiert)
- [Speicherverwaltung]
- [Datenverwaltung]
- [Dokumentation]

Beispiele

Object Professional, Curses, Vermont Views

Funktionsumfang

- Fenster
- Menüs
- Eingabeedition und -formatierung (inkl. Masken)
- Hilfe
- Fehlerbehandlung
- Maus
- Konfiguration
- Data Dictionary
- Automatische Dokumentation

Form

- horizontal
- Draw Down
- Pop Up
- Trennlinien

Wahl

- Mehrfach-/Einfach-
- Anfangszeichen
- beliebiges Zeichen
- Cursor/Maus

Hilfe

- Hilfe für Menü
- Hilfe für Menüpunkt
- Sperrung von Menüpunkten

UI - Eingabeedition und -formatierung

- Bündigkeit
- teilweise/vollständige Muss- oder Kannfelder
- Gross/Klein
- Feldsperrung/Feldpasswort
- Formatmasken
- Gültigkeitsprüfung während/nach Eingabe

M - Module Hierarchy List

Bibliotheksverwaltung 1

Ausleihverkehr buchen 11

Stammdaten verwalten 12

 Dokumente 121

 Benutzer 122

Statistiken erstellen 13

 Dokumente 131

 monatliche Ausleihe 1311

 ...

 Benutzer 132

M - Modulbegriff (M144)

Allgemeine Definition

Möglichst selbständiger Blackbox-Baustein mit

- maximaler intramodularer Bindung
- minimaler intermodularer Koppelung
- Blackbox-Schnittstelle (vgl. ADT)

Weite programmiersprachliche Definition

Einheit von

- Bezeichner
- Anweisungen

Enge programmiersprachliche Definition

Einheit von

- extern aufrufbarem Bezeichner
- getrennt übersetzbaren Anweisungen (vgl. Unit)

M - Preliminary Module Header

MODULE
PURPOSE
IN
OUT
INOUT
AUTHOR
MODIFIED

e.g.

| | |
|----------|---|
| MODULE | AusleiheHistogramm 13111 |
| PURPOSE | Histogramm der Ausleihehäufigkeiten in den Fachgebieten <i>BWL, VWL, Recht, Soziologie/Psychologie und Uebrige</i> am Bildschirm ausgeben |
| IN | <i>BWLAusleihehaeufigkeit, VWLAusleihehaeufigkeit,</i> <i>RechtAusleihehaeufigkeit, SozPsychAusleihehaeufigkeit,</i> <i>UebrigeAusleihehaeufigkeit, Ausleihemonat</i> |
| AUTHOR | mlu |
| MODIFIED | 21.8.1998 |

M - Maximiere **intra**modulare Bindung

Strebe nach ...

- hoher Kontextunabhängigkeit
- Wiederverwendbarkeit
- Unifunktionalität

Vermeide also ...

- zufällige Bindung
z.B. Modularisierung aus Speichergründen
- rein zeitliche Bindung
z.B. Modularisierung, nur weil Funktionen hintereinander stehen
- rein kommunikative Bindung
z.B. Modularisierung, nur weil Funktionen die gleichen Daten benutzen

M - Minimiere **inter**modulare Kopplung

Vermeide **globale** Variablen

- ✓ höhere Wiederverwendbarkeit
 - ✓ bessere Verständlichkeit
- ⇒ Vermeide zum Beispiel parameterlose Unterprogramme
- Umschreiben eines Moduls berührt andere Module
 - Änderung gemeinsamer Daten während der Laufzeit durch einen Modul berührt andere Module

Entwerfe ein Modul als **Black Box**

Minimiere die **Schnittstellenbreite**

Vermeide Module, deren Wirkung vom vorangehenden Aufruf abhängt

M - Weitere Modularisierungs-Grundsätze

Vermeide zu starke Verallgemeinerung

Vermeide zu starke Spezialisierung

Vermeide grosse Module

- ✓ Modul besser testbar
- ✓ Modul übersichtlicher
- ✓ Modul weniger kontextabhängig

Bündle und isoliere systemabhängige Operationen

- ✓ Modul leichter änderbar

Detailed Design (*Feinentwurf*, M234)

detailed module header :=

- ✓ preliminary module header +
- ✓ pseudocode

Pseudocode :=

procedural refinement of the declarative part of a module header

- ✓ Take care not to refine down to code level.
- ✓ The pseudocode should be an abstraction of the real code



Aufgabe: Mit dem Grobentwurf ...
a) bis zum 17.12. vervielfältigen
b) am 17.12. präsentieren

Pseudocode-Kontrollstrukturen

IF THEN

<actions>

ELSE

<actions>

CASE <condition variable>

<condition>: <action1>

...

<conditionN>: <action N>

WHILE <condition>

<actions>

REPEAT

<actions>

UNTIL <condition>

FOR ALL <count variable>

<actions>

M - Detailed Module Header

| | |
|----------|--|
| MODULE | AusleiheHistogramm 13111 |
| PURPOSE | Histogramm der Ausleihhäufigkeiten in BWL, VWL, Recht, Soziologie/Psychologie und Uebrige am Bildschirm ausgeben |
| IN | Ausleihhäufigkeiten BWL, VWL, Recht, Psychologie, Uebrige Ausleihemonat |
| OUT | Histogramm |
| AUTHOR | mlu |
| MODIFIED | 10.12.1998 |

Daten lesen

- Programmtitel ausgeben
- FOR ALL Ausleihhäufigkeiten:
 - Prompt "Ausleihhäufigkeit"
 - Ausleihhäufigkeit lesen
 - Ausleihhäufigkeit aggregieren
 - Prompt "Ausleihemonat"
 - Monatszahl lesen
 - Monatszahl prüfen

Histogramm vorbereiten

- Maximale Häufigkeit ermitteln

Histogramm ausgeben

- Monatszahl in Monatsname umwandeln
- Histogrammkopf ausgeben
- FOR ALL Ausgabehäufigkeiten
 - Balkenbezeichner ausgeben
 - Balkenlängen berechnen
 - Balken ausgeben

M - Pseudocode (Detailed Module Headers)

Ziel

Detailed Module Headers mit

- Pseudocode
- Datenstrukturen

Lösungshinweise

1. Einschränkungen des Programmierwerkzeugs beachten (Manuals/Beispiele)
2. Preliminary Module Headers in der Gruppe verteilen
3. Gemäss Modulhierarchie vorgehen (top down)
4. Pseudocode-Standard beachten
5. Koordinieren
6. Design Walkthrough

Organisatorische Rahmenbedingungen

- Standards des Auftraggebers
- Vorkenntnisse und Präferenzen der Entwickler

Qualität von Sprachen

- Programming in the small
- Programming in the large

Qualität von Compilern

- Effizienz/Zuverlässigkeit des Compilers
- Effizienz/Zuverlässigkeit des erzeugten Code
- Qualität der Fehlermeldungen
- Portabilität des erzeugten Code

Qualität von Entwicklungsumgebungen

- Editor, Debugger, Profiler, Linker, Cross-Referencer, Dokumentationshilfen, Hilfesystem

Funktionsumfang (Mächtigkeit)

- COBOL, C, Lisp, Prolog, PAL/Clipper, C++, Java
Viertgenerationssprachen?

Einfachheit

- Sparsamkeit des Vokabulars
- Maschinenunabhängigkeit
- PL/1, COBOL, Modula-2, C, Ada, Java

Syntax

- begin/end vs. if/endif,
- Formatfreiheit (Gegenbsp. COBOL, FORTRAN)
- Bezeichnerlänge (Gegenbsp. BASIC)

Kontrollstrukturen

- allgemeines CASE
- default label in CASE
- allgemeine Zählschleife
- geschützte Zählvariable und Anfangswert

Implementation - **Programmiersprache III**

- exit
- exception handling

Datentypen

- static vs. dynamic/strong vs. weak typing
- statische und dynamische Datenstrukturen
- benutzerdefinierte Datentypen (Bsp. Bereich)
- berechnete Konstanten

Sichtbarkeit (information hiding)

- lokale Namen (Gegenbsp. COBOL)

Modularität und Objektorientierung

- getrennte Übersetzbarkeit
- Information Hiding (Spezifikations- und Implementationsteil)
- Vererbung
- Polymorphie
- Generische Objekte (Bsp. generischer Stapel)