

Eberhard Lehmann: Projekte im Informatik-Unterricht – Software Engineering,  
Ferd. Dümmlers Verlag, Bonn 1995

## Inhaltsverzeichnis



<b>Vorwort</b>	<b>5</b>
□	
<b>1. Komplexe Software - Projekte - Software-Engineering</b>	<b>7</b>
□	
1.1 Komplexe Softwareprodukte in der Schule	7
1.2 Was versteht man unter Software-Engineering ?	10
1.3 Projektarbeit im Informatik-Unterricht	13
<b>2. Wege zur erfolgreichen Konstruktion komplexer Software</b>	<b>17</b>
2.1 Organisation der Arbeit	17
2.1.1 Ein Software-Life-Cycle	17
2.1.2 Prototyping	22
2.1.3 Projektmanagement und Teamarbeit	25
2.2 Konzepte für die Konstruktion komplexer Software	29
□	
2.2.1 Systemdenken, Modellbildung	29
□	
2.2.2 Strukturiertes Programmieren, Top-Down und Bottom-Up	34
2.2.3 Das Prozedurkonzept	38
2.2.4 Das Modulkonzept	41
2.2.5 Graphische Darstellungsmittel	49
2.2.6 Oberflächendesign - Masken	61
2.2.7 Baustein-Bibliotheken	66
2.2.8 Qualitätsanforderungen an Software aus Benutzer- und Programmierersicht	71
2.2.9 Das Testen von Programmen - Testmethoden	74
2.3 Projekt-Dokumentation - Handbücher, Dokumentationshilfsmittel	79
2.4 Projektkontrolle	85
2.4.1 Warnungen	85
2.4.2 Checklisten für die Softwareproduktion	86
<b>3. Das Projekt TRABRENNBAHN (Entwurfsdokumentation)</b>	<b>93</b>
3.1 Sammeln und Auswerten von Informationen, Problemanalyse - Anforderungsdefinition	95
3.2 Funktionelle Spezifikation und Pflichtenheft	98

3.3 Entwurf, Modularisierung - Entwurfsspezifikation	104
3.4 Modulprogrammierung - Modultests	112
3.5 Systemintegration	119
3.6 Installierung des Systems, Betrieb und Wartung	123
<b>4. Softwarewartung - Reengineering</b>	<b>124</b>
4.1 Grundlagen	124
4.2 Beispielablauf eines Wartungsprojekts	129
4.3 Weitere Software zum Zweck der Wartung	132
<b>5. Aufgaben zur Projektarbeit (zu den Kapiteln 1 bis 4)</b>	<b>137</b>
<b>6. Glossar - wichtige Begriffe des Software-Engineering</b>	<b>156</b>
<b>7. Disketten</b>	<b>166</b>
7.1 Trabrennbahn, SOFTWARE-ENGINEER	166
7.2 Oberflächen und Bausteine, LEH-TOOLS	167
7.3 Spiel Gobang und Räuber-Beute-Systems, MAUS-WIESEL	168
<b>Sachverzeichnis</b>	<b>169</b>
<b>Aufgabenverzeichnis</b>	<b>171</b>

## Vorwort



Die Arbeit an komplexen Systemen mit den Methoden der Projektarbeit gehört zu den Inhalten der Schulinformatik, die besonders praxisnah sind. Überall werden umfangreiche Softwaresysteme benutzt und mindestens soweit analysiert, daß man ihre Funktionen erfaßt und sinnvoll für die eigenen Probleme einsetzen kann. Doch wie entstehen derartige Systeme, welche Konstruktionsverfahren setzt man ein, welche Kenntnisse muß man haben, um komplexe Probleme soweit zu modellieren, daß sie einen brauchbaren Ausschnitt aus der vorliegenden Realität darstellen? Gerade hiermit beschäftigt sich das vorliegende Buch.

**Das Buch ist als ein Schülerbuch konzipiert, das in die Methoden zur Bearbeitung von Softwareprojekten (Software-Engineering) einführt und zahlreiche Übungsaufgaben zur Verfügung stellt. Es ist unabhängig von den anderen schon erschienenen Bänden zur Reihe "Analyse, Konstruktion, Wartung komplexer Software", setzt diese Bände also nicht voraus!**

Kapitel 1 beschäftigt sich mit dem Umfeld der Konstruktion komplexer Softwareprodukte, insbesondere mit der **Projektmethode** und dem Begriff des Software-Engineering. Dieses Hintergrundwissen ist auch für Schüler wichtig, soll diesen bewußt gemacht werden und trägt wesentlich zum Verständnis der Zusammenhänge bei.

Kapitel 2.1 zeigt **Wege zur erfolgreichen Konstruktion von Softwaresystemen** auf und betont dabei die Bedeutung des Projektmanagements und des Prototyping. Kapitel 2.2 beschreibt die **Konzepte**, die man beherrschen muß, um komplexe Datenverarbeitungsprobleme zu bewältigen. Programmiersprachen spielen hier nur eine untergeordnete Rolle, die Konzepte sind programmiersprachenunabhängig. Wegen der besonderen Bedeutung der **Projektdokumentation** für spätere Wartungsarbeiten an der Software erhält diese ein eigenes Kapitel (2.3). Dabei werden die Grundlagen einer Programmiersprache vorausgesetzt, hier PASCAL. Schul-Softwareprojekte scheitern nicht selten an diversen Unzulänglichkeiten. Die **Warnungen** in Kapitel 2.4 entspringen meiner längjährigen Erfahrung in Projektarbeit. Die angegebenen **Checklisten** können helfen, Projektarbeit erfolgreich abzuschließen!

Kapitel 3 bringt die **Beschreibung eines realen Projektablaufs in Form einer Entwicklungsdokumentation**. Das gewählte Beispiel "Simulation einer Trabrennbahn" wurde mit einer Schülergruppe in Klasse 12/13 durchgeführt. Es kann als Muster für andere Projekte dienen.

Grundzüge der Projektarbeit kann man auch bei der **Wartung von Software** (Reengineering) lernen. Grundlegende Ausführungen hierzu bringt Kapitel 4, als

Beispiel wird erneut das Trabrennbahn-Programm herangezogen, von dem hier eine zweite Version entsteht. In dem Kapitel werden auch Hinweise auf weitere Softwaresysteme des Autors gegeben, die für Wartungsprojekte geeignet sind.

Es gibt in der Literatur kaum **Aufgabenstellungen zu Informatik-Projektarbeit!** Diese Lücke soll mit den **Übungsaufgaben** in Kapitel 5 geschlossen werden. Diese beschäftigen sich mit Fragen aus verschiedenen Bereichen des Software-Engineering. Es sind insbesondere Fragestellungen zu den Konzepten und dem Bewußtmachen von Zusammenhängen.

Kapitel 6 ist ein **Glossar**, in dem weitere Begriffe im Zusammenhang mit Software-Engineering erläutert werden, soweit sie nicht schon im Buchtext näher betrachtet wurden.

Die meisten Darstellungen im Buch sind programmiersprachenunabhängig. Die Programmiersprache PASCAL (Turbo-PASCAL 6) wird wichtig beim Angebot an Hilfsprogrammen und wiederverwendbaren Bausteinen (Kapitel 2.2.6, 2.2.7) und beim Trabrennbahnprojekt in Kapitel 3. Die MS-DOS-**Begleitdiskette** SOFTWARE-ENGINEER (Turbo-PASCAL 6) enthält nicht nur Quellprogramm und kompiliertes Programm zum Trabrennbahnprojekt (das hiermit auch für Wartungsarbeiten zur Verfügung steht), sondern auch eine zweite Version des Trabrennbahn-Programms nach einem Wartungsprojekt, siehe Kapitel 4.

Kapitel 7 verweist auf weitere Bücher und Disketten in Zusammenhang mit Projektarbeit.

***Zum Adressatenkreis des Buches gehören Schüler, Lehrer, Studenten, Dozenten und alle, die sich mit der Konstruktion von Software beschäftigen.***

***An Schüler und Studenten:*** Projektarbeit ist die Arbeitsform im Informatikunterricht, die wegen ihrer Praxisnähe besonders wichtig ist. In der beruflichen Praxis werden nicht nur im Bereich der Datenverarbeitung komplexe Probleme mit dieser Arbeitsform gelöst, so daß viele der hier vermittelten Methoden über die Informatik hinausgehen (z.B. Teamarbeit). Nach allen Erfahrungen macht diese Arbeitsform zudem viel Spaß. Ich hoffe, daß auch Sie diese Freude empfinden, wenn Sie Ihr Projektthema schrittweise einer Lösung zuführen und schließlich das fertige Produkt vor sich haben!

***An Lehrer und Dozenten:*** Projektartiger Unterricht erfordert einige Übung. Sie finden in dem Buch viele Hinweise zur Projektmethode und zur Unterrichtspraxis.

***Für private Softwarekonstruktoren*** bietet das Buch Grundlagen und Methoden, die zur erfolgreichen Konstruktion größerer Softwareprodukte führen.

Berlin, Januar 1995

Eberhard Lehmann