Software Engineering

The Basics of Software Development

Acknowledgments

I would like to express my deepest appreciation to the Almighty God for his everlasting love and blessings. This book would have remained a dream had it not been for him.

I cannot find words to express my gratitude to all my best friends who have always supported me, and incented me to strive towards my goal.

In addition, a special thanks to **Rev. Prof. Jones Kaleli (Vice Chancellor, Kabarak University)** for his guidance and motivational support which has inspired me to achieve this productive result.

Dedication

This book is dedicated to My Dearest Late Mother who has been a great rock of stability throughout all these years of my life.

Abstract

This book is an introduction to the art of software engineering. It is intended to be used as a textbook for an undergraduate level course.

When teaching software engineering courses, it is highly important to have good text books that are well-founded, up-to-date, and easily accessible for students. However currently, the available text books in the market are either very broad or highly specialized, making it hard for students to select appropriate books for specific software engineering courses.

Software engineering is also about communication. Teams do not consist only of developers, but also of testers, architects, system engineers, customers, and project managers. Software projects can be so large that we have to do careful planning.

Implementation is no longer just about writing codes, but it is also about following guidelines, writing documentation and writing unit tests. All of these different pieces have to fit together and we have to be able to spot problematic areas using metrics.

This will enable us to know whether the codes follow certain standards.

This is reflected from the fact that once we have finished the coding, we must complete other tasks before concluding the project.

A clear example is large projects maintaining software which can keep many people on their toes. There are so many factors influencing the success or failure of a project, thus we need to acquire knowledge about good management skills. And last but not the least, a good software engineer, like any engineer, needs tools which are gained through knowledge.

In this book, it briefly explains and discusses an approach of using a web-based system for creating collaborative and peer-reviewed text books that can be customized individually for specific courses.

Contexts **Page No**

Chapter 1 **1**

1.1 Introduction to Software Engineering 2

1.2 Software Crisis 2

1.3 Various Contributing Factors/Reasons Responsible for Software Crisis 3

1.4 Differences between Software Engineering and Traditional Engineering. 4

1.5 Goals/ Objective of Software Engineering 4

1.6 Principles of Software Engineering 5

1.7 Process of Software System Development 8

1.7.1 Phased Development Process 8

1.7.2 Software Development Life Cycle 8

1.8 Software Engineering Models/Paradigm 15

1.8.1 Waterfall model 15

1.8.2 Prototype Model 17

1.8.3 Iterative Enhancement Model 19

1.8.4 Spiral Model 21

Chapter 2 **24**

2.1 Introduction 25

2.2 Planning a Software Project 25

2.3 Software Cost Estimation 26

2.4 COCOMO Model 30

2.5 Project Scheduling 33

2.5.1 Basic Principles of Software Project Scheduling 33

2.5.2 Project Scheduling Activities 34

2.5.3 Software Project Scheduling Techniques	34
2.6 Personnel Planning.	38
2.7 Team Structures	38
2.8 Software Configuration Management	41
2.9 Software Quality and Quality Assurance	43
2.10 Risk Management	45
Chapter 3	**49**
3.1 Introduction	50
3.2 Software Requirement Analysis	50
3.3 Structured Analysis	51
3.3.1 Data Flow Diagram (DFD)	52
3.3.1.1 Symbols used for Constructing DFDs	53
3.3.1.2 Guidelines for developing DFD	55
3.3.1.3 Context Diagram	55
3.3.1.4 Detailed level DFD	55
3.3.2 Data Dictionary	56
3.3.2.1 Need for a Data Dictionary	56
3.3.2.2 Components of Data Dictionary	57
3.3.2.3 Advantage of data dictionary	59
3.3.2.4 Disadvantages of Data Dictionary	59
3.4 Object Oriented Analysis	60
3.5 Software Requirement Specification (SRS)	62
3.6 SRS Validation	65
Chapter 4	**67**
4.1 Introduction	68
4.2 Software Design Fundamentals	68
4.3 Software Design Principles	69

4.4 Modularity 72
4.5 Structured Design Methodology 78
4.5.1 Structured Chart 79
4.6 Object Oriented Design Methodology 82
4.7 Design Verification 83
Chapter 5 **85**
5.1 Introduction 86
5.2 Programming Style 86
5.3 Structured Programming 90
5.4 Documentation 93
5.5 Verification and Validation (V & V) 95
5.6 Monitoring and Control 98
Chapter 6 **100**
6.1 Introduction 101
6.2 Need of Software Metrics 101
6.3 Benefits of Software Metrics 102
6.4 Size Metrics 102
6.5 Control Complexity Metrics 104
6.6 Object Oriented Metrics 106
Chapter 7 **108**
7.1 Software Reliability 109
7.2 Error, Faults and Failures 110
7.3 Software Reliability Metrics 110
7.4 Fault Avoidance 111
7.5 Fault Tolerance 112
7.6 Exception Handling 113

Chapter 8	**114**
8.1 Introduction	115
8.2 Objectives of Software Testing	116
8.3 Testing Principles	116
8.4 White box and Black box testing techniques	117
8.5 Software Testing Strategies	126
8.6 Unit Testing	128
8.7 Integration Testing	130
8.8 Validation Testing	132
8.9 System Testing	133
8.10 Regression Testing	134
8.11 Alpha and Beta Testing	135
Chapter 9	**137**
9.1 Introduction	138
9.2 Aim of Software Maintenance	139
9.3 Types of Software Maintenance	140
9.4 Maintainability	142
9.5 Maintenance Tasks	144
9.6 Maintenance Side Effects	145
Chapter 10	**147**
10.1 Overview of Case	148
10.2 Case Tools	148
10.3 Types of CASE Tools	149
Reference Material	153

List of Figures **Page No.**

Figure 1: Software Development Life Cycle	9
Figure 2: Waterfall Model	16
Figure 3: Prototype Model	18
Figure 4: Iterative Enhancement Model	20
Figure 5: Spiral Model	21
Figure 6: Activity Networks	35
Figure 7: Gantt Charts	37
Figure 8: Democratic Team	38
Figure 9: Structure of a Chief Programmer Team	40
Figure 10: Hierarchical Team Structure	41
Figure 11: Risk Management	46
Figure 12: Requirement Phase Activities	50
Figure 13: Context Diagram	55
Figure 14: Detailed Level DFD	56
Figure 15: Coupling	74
Figure 16: Highly Coupled	74
Figure 17: Content Coupling	75
Figure 18: Common Coupling	76
Figure 19: Communicational Cohesion	77
Figure 20: Sequential Cohesion	78
Figure 21: Arrow with a Circular Tail	80
Figure 22: Arrow with a Shaded Circular Tail	80
Figure 23: Arrow with Diamond	81
Figure 24: Arc with Arrowhead	81
Figure 25: Sequence structure	91
Figure 26: Decision Structure	91

Figure 27: While Structure 92
Figure 28: Do-While Structure 92
Figure 29: Control Complexity Metrics 106
Figure 30(i): Flow Graphs 119
Figure 30(ii): Flow Graphs 121
Figure 31: Connection Matrix 121
Figure 32: Black Box Testing 122
Figure 33: Cause Effect 'Identity' Graph 124
Figure 34: Cause Effect 'NOT' Graph 124
Figure 35: Cause Effect 'OR' Graph 125
Figure 36: Cause Effect 'AND' Graph 125
Figure 37: Cause Effect Mixed Graph 126
Figure 38: Software Testing Strategies 127
Figure 39: Unit Testing procedure 130
Figure 40: Top Down Integration 131
Figure 41: Bottom up Integration 132

The Book is available at the following link:

https://www.morebooks.de/store/gb/book/software-engineering/isbn/978-3-659-55566-4