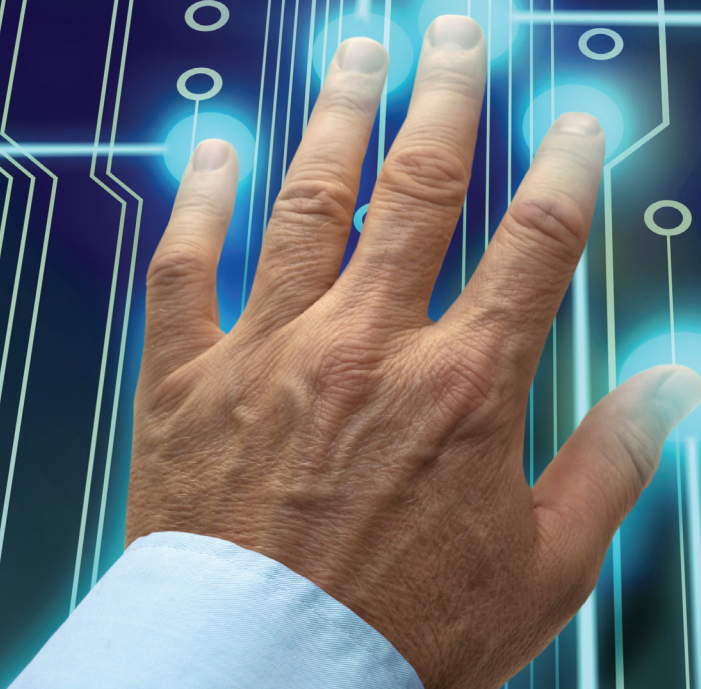


Computer System Reliability

Safety and Usability



B. S. DHILLON



CRC Press
Taylor & Francis Group

Computer System Reliability

Safety and Usability

Computer System Reliability

Safety and Usability

B. S. DHILLON



CRC Press

Taylor & Francis Group

Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

© 2013 by Taylor & Francis Group, LLC
CRC Press is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works
Version Date: 20130410

International Standard Book Number-13: 978-1-4665-7313-0 (eBook - PDF)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access www.copyright.com (<http://www.copyright.com/>) or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Visit the Taylor & Francis Web site at
<http://www.taylorandfrancis.com>

and the CRC Press Web site at
<http://www.crcpress.com>

*This book is affectionately dedicated to all my Scythian ancestors
for their firm belief in number 40. (This is my 40th book!)*

Contents

Preface.....xv

About the Author xvii

1. Introduction.....1

1.1 Background.....1

1.2 Facts, Figures, and Examples1

1.3 Terms and Definitions3

1.4 Sources To Obtain Information Related to Computer System
Reliability, Safety, and Usability4

1.4.1 Journals and Magazines5

1.4.2 Conference Proceedings5

1.4.3 Standards and Handbooks.....5

1.4.4 Data Sources6

1.4.5 Books.....7

1.5 Scope of the Book.....8

References9

2. Basic Mathematical Concepts13

2.1 Introduction13

2.2 Arithmetic Mean and Mean Deviation.....13

2.2.1 Arithmetic Mean.....14

2.2.2 Mean Deviation.....14

2.3 Boolean Algebra Laws.....15

2.4 Probability Definition and Properties.....16

2.5 Probability-Related Definitions.....17

2.5.1 Cumulative Distribution Function17

2.5.2 Probability Density Function.....18

2.5.3 Expected Value.....19

2.6 Probability Distributions19

2.6.1 Exponential Distribution20

2.6.2 Rayleigh Distribution21

2.6.3 Weibull Distribution.....21

2.6.4 Bathtub Hazard Rate Curve Distribution22

2.7 Laplace Transform Definition, Common Laplace
Transforms, and Final-Value Theorem’s Laplace Transform22

2.7.1 Laplace Transform: Final-Value Theorem23

2.8 Laplace Transforms’ Application in Solving First-Order
Differential Equations24

References26

- 3. Reliability, Safety, and Usability Basics 29
 - 3.1 Introduction 29
 - 3.2 Bathtub Hazard Rate Curve 29
 - 3.3 General Reliability-Related Formulas 31
 - 3.3.1 Failure (or Probability) Density Function..... 31
 - 3.3.2 Hazard Rate (or Time-Dependent Failure Rate) Function 31
 - 3.3.3 General Reliability Function 32
 - 3.3.4 Mean Time to Failure 33
 - 3.4 Reliability Configurations 34
 - 3.4.1 Series Network 34
 - 3.4.2 Parallel Network 37
 - 3.4.3 *k*-out-of-*m* Network..... 39
 - 3.4.4 Standby Redundancy 41
 - 3.4.5 Bridge Network 42
 - 3.5 Reliability Evaluation Methods 44
 - 3.5.1 Failure Modes and Effect Analysis (FMEA) 44
 - 3.5.2 Fault Tree Analysis (FTA) 45
 - 3.5.2.1 Common Fault Tree Symbols 45
 - 3.5.2.2 Steps Involved in Developing a Fault Tree 46
 - 3.5.2.3 Probability Evaluation of Fault Trees 46
 - 3.5.3 Markov Method 48
 - 3.6 Need for Safety and the Role of Engineers in Designing for Safety 51
 - 3.7 Classifications of Product Hazards 52
 - 3.8 Human Factors Basics for Engineering Usability 53
 - 3.8.1 Comparisons of Humans’ and Machines’ Capabilities/Limitations..... 54
 - 3.8.2 Typical Human Behaviors and Their Corresponding Design Considerations 55
 - 3.8.3 Human Sensory Capacities 55
 - 3.8.3.1 Touch..... 56
 - 3.8.3.2 Sight..... 56
 - 3.8.3.3 Vibration 57
 - 3.8.3.4 Noise 58
 - References 59
- 4. Computer System Reliability Basics..... 61
 - 4.1 Introduction 61
 - 4.2 Hardware Reliability Versus Software Reliability 61
 - 4.3 Major Sources of Computer Failures and Issues in Computer System Reliability 62
 - 4.4 Fault Classifications and Computer Reliability Measures 64
 - 4.5 Fault Masking 65

- 4.5.1 Triple Modular Redundancy (TMR)65
 - 4.5.1.1 TMR System Maximum Reliability with Perfect Voter 67
 - 4.5.1.2 TMR System Time-Dependent Reliability and Mean Time to Failure68
 - 4.5.1.3 Reliability Analysis of TMR System with Perfect Voter and Repair 70
 - 4.5.2 N-Modular Redundancy (NMR)72
 - 4.6 Reliability Analysis of Redundant Computer Systems with Common-Cause Failures73
 - 4.6.1 Model I..... 74
 - 4.6.2 Model II78
 - References 81
- 5. Software Reliability Assessment and Improvement Methods.....83**
 - 5.1 Introduction83
 - 5.2 Software Reliability Assessment Methods.....83
 - 5.2.1 Software Metrics84
 - 5.2.1.1 Metric I: Code and Unit Test Phase Measure84
 - 5.2.1.2 Metric II: Design Phase Measure84
 - 5.2.2 Analytical Methods85
 - 5.2.3 Software Reliability Models.....85
 - 5.2.3.1 Air Force Model.....86
 - 5.2.3.2 Musa Model88
 - 5.2.3.3 Mills Model89
 - 5.2.3.4 Power Model91
 - 5.2.3.5 Shooman Model.....92
 - 5.3 Software Reliability Improvement Methods.....93
 - 5.3.1 Fault-Tolerant Software Design Methods94
 - 5.3.2 Reliable Software Design Methods95
 - 5.3.3 Testing96
 - 5.3.4 Formal Methods97
 - References 98
- 6. Software Quality 101**
 - 6.1 Introduction 101
 - 6.2 Software Quality Factors and Their Classifications 101
 - 6.2.1 Product Revision Factors 102
 - 6.2.2 Product Operation Factors.....102
 - 6.2.3 Product Transition Factors.....103
 - 6.3 Quality Methods for Use during Software Development 103
 - 6.3.1 Pareto Diagram103
 - 6.3.2 Checklists.....104
 - 6.3.3 Run Charts.....105

6.4	Quality Measures during the Software Development Life Cycle (SDLC)	105
6.5	Software Quality-Related Metrics	106
6.5.1	Metric I	107
6.5.2	Metric II	107
6.5.3	Metric III	108
6.5.4	Metric IV	108
6.5.5	Metric V	108
6.5.6	Metric VI	109
6.5.7	Metric VII	109
6.5.8	Metric VIII	109
6.5.9	Metric IX	109
6.5.10	Metric X	109
6.6	Software Quality Assurance Manager's Responsibilities and Elements of a Successful Software Quality Assurance Program	110
6.7	Software Quality Assurance Standards and Advantages	111
	References	113
7.	Human Error and Software Bugs in Computer Systems	115
7.1	Introduction	115
7.2	Facts, Figures, and Examples	115
7.3	Factors Affecting the Occurrence of Human Error in Computer Systems	116
7.4	Computer Failure Categories and Hardware and Software Error Sources	118
7.5	Common Software Errors in Programming	119
7.6	Factors Causing Human Errors during Software Development and Maintenance	119
7.6.1	Classification I Factors	119
7.6.2	Classification II Factors	120
7.7	Methods To Prevent Programmers from Inadvertently Introducing Bugs during the Software Writing Process	121
7.8	Software Error-Related Metrics	122
7.8.1	Metric I: Defect Density	122
7.8.2	Metric II: Cumulative Failure Profile	122
7.8.3	Metric III: Fault Density	123
7.8.4	Metric IV: Defect Indices	123
	References	125
8.	Software Safety and Internet Reliability	127
8.1	Introduction	127
8.2	Software Safety Classifications and Potential Hazards	127
8.3	Software Risk Classifications	128
8.4	Basic Software System Safety-Related Tasks	129

- 8.5 Software Safety Assurance Program and Software Quality Assurance Organization’s Role in Regard to Software Safety130
- 8.6 Software Hazard Analysis Methods..... 131
 - 8.6.1 Code Walk-Through 132
 - 8.6.2 Software Fault Tree Analysis (SFTA) 132
 - 8.6.3 Software Sneak Circuit Analysis 133
 - 8.6.4 Proof of Correctness 134
 - 8.6.5 Failure Modes and Effect Analysis (FMEA) 134
- 8.7 Software Standards and Useful Software Safety Design-Related Guidelines 134
- 8.8 Internet Facts, Figures, and Failure Examples; Benefits and Impediments of Internet Electronic Commerce; and Internet Reliability-Related Observations 135
- 8.9 Classifications of Internet Outages 137
- 8.10 An Approach for Automating Fault Detection in Internet Services and Models for Performing Internet Reliability and Availability Analysis 138
 - 8.10.1 Model I..... 139
 - 8.10.2 Model II 140
- References 143
- 9. Software Usability 147**
 - 9.1 Introduction 147
 - 9.2 Need for Considering Usability during the Software Development Phase and Basic Principles of the Human-Computer Interface 147
 - 9.3 Software Usability Engineering Process 148
 - 9.4 Steps for Improving Software Product Usability 150
 - 9.5 Software Usability Inspection Methods and Considerations for Their Selection 151
 - 9.6 Software Usability Testing Methods and Important Factors in Regard to Such Methods 153
 - 9.7 Guidelines for Conducting Software Usability Testing 155
 - References 156
- 10. Web Usability 159**
 - 10.1 Introduction 159
 - 10.2 Web Usability Facts and Figures 160
 - 10.3 Common Web Design Errors 160
 - 10.4 Web Page Design..... 161
 - 10.4.1 Image Usage..... 162
 - 10.4.2 Page Size..... 163
 - 10.4.3 Textual Element Usage 163
 - 10.4.4 Help for Users..... 164
 - 10.4.5 Font Usage..... 164

- 10.5 Website Design 165
 - 10.5.1 Shared Elements of Site Pages 165
 - 10.5.2 Site Testing and Maintenance 166
 - 10.5.3 Site Organization 166
- 10.6 Navigation Aids 167
 - 10.6.1 Menus and Menu Bar Usage 167
 - 10.6.2 Navigation Bar Usage 168
 - 10.6.3 Link Usage 169
- 10.7 Tools for Evaluating Web Usability 170
 - 10.7.1 NetRaker 170
 - 10.7.2 Web SAT 170
 - 10.7.3 Lift 171
 - 10.7.4 Max 171
- 10.8 Questions to Evaluate Effectiveness of Website Message Communication 172
 - 10.8.1 Text 172
 - 10.8.2 Concept 173
 - 10.8.3 Content 173
 - 10.8.4 Mechanics 173
 - 10.8.5 Navigation 174
 - 10.8.6 Design 174
- References 175
- 11. Computer System Life-Cycle Costing 177**
 - 11.1 Introduction 177
 - 11.2 Models for Estimating Computer System Life-Cycle Cost 177
 - 11.2.1 Model I 177
 - 11.2.2 Model II 180
 - 11.3 Models for Estimating Computer System Servicing-Labor Cost and Maintenance Cost 181
 - 11.3.1 Model I: Computer System Servicing-Labor Cost 181
 - 11.3.2 Model II: Computer System Maintenance Cost 182
 - 11.4 Software Costing and Associated Problems and Steps 183
 - 11.5 Model for Estimating Software Life-Cycle Cost and Influencing Factors 185
 - 11.6 Models for Estimating Software Costs 187
 - 11.6.1 Model I 187
 - 11.6.2 Model II 188
 - 11.6.3 Model III 188
 - 11.6.4 Model IV 189
 - 11.6.5 Model V 189
 - 11.7 Models for Estimating Software Costs 190
 - 11.7.1 Multiplicative Models 191
 - 11.7.2 Analytic Models 191
 - 11.7.3 Linear Models 191

11.7.4	Tabular Models.....	192
11.7.5	Composite Models	192
	References	193
Appendix: Bibliography of Literature on Computer System		
	Reliability, Safety, and Usability	195

Preface

Computer systems are increasingly being used at an alarming rate for various purposes. They have become an important element of the world economy because billions of dollars are spent each year to develop, manufacture, operate, and maintain various types of computer systems around the globe. Their reliability, safety, and usability have become an important concern because of problems such as high cost, wrong decisions and actions, and accidental deaths. For example, a study performed by the National Institute of Standards in 2002 found that software defects alone cost the United States economy about \$59 billion annually, i.e., around 0.6% of its gross domestic product (GDP).

Computer system reliability, safety, and usability have become more important than ever before. In response, a large number of journal and conference proceedings' articles on various aspects of computer system reliability, safety, and usability have been published over the years. However, to the best of the author's knowledge, there is no specific book on the topic. This causes a great deal of difficulty for information seekers because they have to consult many different and diverse sources.

Thus, the main objective of this book is to combine computer system reliability, safety, usability, and other related topics into a single volume and to eliminate the need to consult many different and diverse sources to obtain desired information. The book contains a chapter on mathematical concepts considered necessary to understand the material presented in subsequent chapters.

The topics covered in the volume are treated in such a manner that the reader will require no previous specialized knowledge to understand the contents. At appropriate places, the book contains examples along with their solutions, and at the end of each chapter there are numerous problems to test the reader's comprehension. The sources of most of the materials presented are given in the reference section at the end of each chapter. An extensive list of publications dating from 1967 to 2011—directly or indirectly related to computer system reliability, safety, and usability—is provided at the end of this book to give readers a view of the intensity of the developments in this area.

The book is composed of 11 chapters. Chapter 1 presents various introductory aspects of computer system reliability including safety; usability-related facts, figures, terms, and definitions; and sources for obtaining useful information on computer system reliability, safety, and usability. Chapter 2 reviews mathematical concepts considered useful to understanding subsequent chapters. Some of the topics covered in the chapter are arithmetic mean and mean deviation, Boolean algebra laws, probability properties, probability distributions, and useful definitions.

Chapter 3 presents various introductory aspects of reliability, safety, and usability. Chapter 4 presents computer system reliability basics. Some of the topics covered in this chapter are hardware reliability versus software reliability, major sources of computer failures, issues in computer system reliability, fault classifications, fault masking, and computer reliability measures. Chapter 5 is devoted to software reliability assessment and improvement methods. A number of methods grouped under seven categories are presented in this chapter.

Chapters 6 and 7 present various important aspects of software quality and human error and software bugs in computer systems, respectively. Chapter 8 is devoted to software safety and Internet reliability. It covers topics such as software safety classifications; potential software hazards; software safety assurance programs; software hazard analysis methods; Internet facts, figures, and examples; Internet outage classifications; and models for performing Internet reliability and availability analysis.

Chapter 9 covers various important aspects of software usability, including the need to consider usability during the software development phase, the software usability engineering process, software usability inspection methods, software usability test methods, and guidelines for conducting software usability testing. Chapter 10 is devoted to web usability. Some of the topics covered in the chapter are web usability facts and figures, common web design errors, web page design, tools for evaluating web usability, and questions to evaluate the effectiveness of website message communications. Finally, Chapter 11 presents various important aspects of computer system life-cycle costing.

This book will be useful to many individuals, including computer engineers, software engineers, design engineers, system engineers, human factors engineers, and other professionals involved with computers and the Internet—engineering managers and administrators; reliability and other engineers-at-large; researchers and instructors involved with computer systems; and graduate and senior undergraduate students in computer engineering, software engineering, system engineering, computer science, etc.

The author is deeply indebted to many individuals, including family members, friends, colleagues, and students, for their invisible input. The unseen contributions of my children also are appreciated. Last but not least, I thank my wife, Rosy, my other half and friend, for typing this entire book and her timely help in proofreading.

B. S. Dhillon
Ottawa, Ontario

About the Author

Dr. B. S. Dhillon is a professor of engineering management in the Department of Mechanical Engineering at the University of Ottawa. He has served as a chairman/director of the Mechanical Engineering Department/Engineering Management Program for over ten years at the same institution. He is the founder of the probability distribution named the “Dhillon Distribution/Law/Model” by statistical researchers in their publications around the world. He has published over 364 (i.e., 217 [70 single authored and 147 coauthored] journal and 147 conference proceedings) articles on reliability engineering, maintainability, safety, engineering management, etc. He is or has been on the editorial boards of eleven international scientific journals. In addition, Dr. Dhillon has written forty books on various aspects of health care, engineering management, design, reliability, safety, and quality published by Wiley (1981), Van Nostrand (1982), Butterworth (1983), Marcel Dekker (1984), Pergamon (1986), etc. His books are being used in over one hundred countries, and many of them are translated into languages such as German, Russian, Chinese, and Persian (Iranian).

He has served as general chairman of two international conferences on reliability and quality control held in Los Angeles and Paris in 1987. Professor Dhillon has also served as a consultant to various organizations and bodies and has many years of experience in the industrial sector. At the University of Ottawa, he has been teaching reliability, quality, engineering management, design, and related areas for over thirty-three years, and he has also lectured in over fifty countries, including keynote addresses at various international scientific conferences held in North America, Europe, Asia, and Africa. In March 2004, Dr. Dhillon was a distinguished speaker at the Conference/Workshop on Surgical Errors (sponsored by White House Health and Safety Committee and Pentagon) held on Capitol Hill (One Constitution Avenue, Washington, DC).

Professor Dhillon attended the University of Wales, where he received a BS in electrical and electronic engineering and an MS in mechanical engineering. He received a PhD in industrial engineering from the University of Windsor.

1

Introduction

1.1 Background

Nowadays, computer systems have become an important element of the world economy, and each year billions of dollars are spent to develop, manufacture, operate, and maintain various types of computer systems around the globe. Their reliability, safety, and usability have become an important concern because of problems such as high cost, wrong decisions and actions, and accidental deaths. For example, a study conducted by the National Institute of Standards and Technology in 2002 found that software defects alone cost the U.S. economy about \$59 billion annually, i.e., around 0.6% of its gross domestic product (GDP) [1].

The history of computer system reliability can be traced back to the late 1940s and 1950s to the works of Shannon [2], Hamming [3], Von Neumann [4], and Moore and Shannon [5]. For example, in 1956 Von Neumann [4] proposed the triple modular redundancy (TMR) scheme (nowadays widely used in computers) to improve system reliability. In 1965, Pierce published a book entitled *Failure Tolerant Design* [6]. This was probably the first book concerned with computer system reliability.

Over the years, a large number of publications related to computer system reliability have appeared. A comprehensive list of useful publications is provided in the appendix to this book.

1.2 Facts, Figures, and Examples

Some facts, figures, and examples concerned with computer system reliability are as follows:

- In 2002, a study commissioned by the National Institute of Standards and Technology (NIST) reported that software errors cost the U.S. economy about \$59 billion annually [1].

- As per Kletz [7] and Herrman [8], the number of people killed due to computer system malfunctions worldwide up to the end of 1992 was somewhere between 1,000 and 3,000.
- The Internet has grown from four hosts in 1969 to over 147 million hosts and thirty-eight sites in 2002, and in 2001 there were over 52,000 Internet-related incidents and failures [9, 10].
- In 2000, the Internet economy generated about \$830 billion in revenues in the United States [9–11].
- National Aeronautics and Space Administration's (NASA's) Saturn V Launch computer (circa 1964) had a mean time to failure (MTTF) of 25,000 hours [12–14].
- As per Landauer [15], an average software program contains about forty design flaws that impair the ability of workers to use it effectively.
- A pilot set the heading in a plane's computer-controlled inertial navigation system as 270° instead of 027°; the plane ran out of fuel and caused twelve fatalities [16].
- In 1966, European Space Agency's US\$1 billion prototype Ariane 5 rocket was destroyed just forty seconds after launch because of a bug in the onboard guidance computer program [17].
- On April 25, 1997, a misconfigured router of a Virginia service provider injected an incorrect map into the global Internet. In turn, this caused network congestion, instability, and overload of Internet router table memory that ultimately shut down the majority of the main Internet backbones for about 2 hours [18].
- In 1963, a software error resulted in the incapacitation of a North American Air Defense Command (NORAD) exercise [19].
- As per Myers and Robson [20], 50% to 80% of all source code development accounts, directly or indirectly, for the user interface.
- A software error in the code controlling the Therac-25 radiation therapy machine caused many deaths in the 1980s [21].
- A computer opened the vent valve on the wrong vessel due to a software error, and fourteen tons of carbon dioxide were vented and lost [16, 22].
- As per Kuhn [23], a number of studies conducted over the years indicate that the reliability of Internet paths falls far short of the 99.999% availability expected in the public-switched telephone network (PSTN).
- In 1981, the launching of the first U.S. space shuttle was postponed for about twenty minutes prior to the scheduled launching time due to a software error [24].
- Some small-scale studies performed in 1994 and 2000 clearly reported that the probability of encountering a major routing pathology along a path (with respect to the Internet) was roughly 1.5% to 3.3% [25, 26].

- In 1991, a software fault caused a MIM-104 Patriot (surface-to-air missile system) to fail to intercept an incoming Iraqi Scud missile that caused twenty-eight American fatalities in Saudi Arabia [27].
- On August 14, 1998, a misconfigured Key Internet Database server mistakenly directed all queries for Internet machines with names ending in “.net” to the wrong secondary database server. This problem resulted in the failure of most connections to “.net” Internet web servers and other end stations for many hours [28].
- A case study performed in regard to Internet outages over a period of one year (November 1997–November 1998) categorized the outages under the following classifications (along with their occurrence percentages in parentheses) [28]:
 - Software problems (1.3%)
 - Malicious attacks (1.5%)
 - Sluggish/congestion (4.6%)
 - Unknown/undetermined/no problem (5.6%)
 - Miscellaneous (5.9%)
 - Routing problem (6.1%)
 - Interface down (6.2%)
 - Hardware problem (6.2%)
 - Unreachable (12.6%)
 - Fiber cut/circuit carrier problem (15.3%)
 - Power outage (16%)
 - Maintenance (16.2%)

1.3 Terms and Definitions

This section presents some useful terms and definitions concerned with various aspects of computer system reliability [14, 29–41].

Accident. An event that involves damage to a certain system/unit that suddenly disrupts the potential or current system/unit output.

Availability. The probability that an item/system is available for use or application when needed.

Debugging. The process of isolating and eradicating errors.

Downtime. The time period during which the item/system is not in a condition to perform its specified mission.

Failure. The inability of an item/system to perform its stated function.

- Fault.** An attribute that adversely affects the reliability of an item.
- Fault-tolerant computing.** The ability to execute specified algorithms successfully, irrespective of computer hardware malfunctions and software errors.
- Hazard.** The source of energy and the physiological and behavioral factors that, when uncontrolled, lead to harmful occurrences.
- Human error.** The failure to carry out a specified task (or the performance of a prohibited action) that could result in disruption of scheduled operations or damage to property.
- Human factors.** A body of scientific facts concerning the human characteristics (the term includes all biomedical and psychosocial considerations).
- Mission time.** The time during which the item/system is carrying out its stated mission.
- Redundancy.** The existence of more than one means to carry out a specified function.
- Reliability.** The probability that a system/unit/item will perform its assigned mission satisfactorily for the stated period of time when used according to the specified conditions.
- Safety.** Conservation of human life and the prevention of damage to systems/units/items as per mission-stated requirements.
- Software error.** A conceptual clerical or syntactic discrepancy that leads to one or more faults in the software.
- Software reliability.** The probability of a given software functioning for a specified time interval, without an error, when used according to the designed conditions on the stated machine.
- Usability.** The quality of an interactive system with respect to factors such as user satisfaction, ease of learning, and ease of use.
- Usability evaluation.** Any analytical or empirical study directed at assessing or understanding the usability of an interactive system/product.
- User-centered design.** An early and continuous involvement of potential users in the product design process.
- User interface.** The physical representations and procedures to view and interact with the product/system functionality.

1.4 Sources To Obtain Information Related to Computer System Reliability, Safety, and Usability

There are many different sources for obtaining information related to computer system reliability, safety, and usability. These include journals and

magazines, conference proceedings, industry standards and handbooks, data sources, and books. Some of these sources are listed in the following subsections [14, 29, 30]:

1.4.1 Journals and Magazines

ACM Transactions on Computer-Human Interaction (TOCHI)

Human-Computer Interaction

IEEE Transactions on Reliability

Interacting with Computers

International Journal of Industrial Ergonomics

International Journal of Man-Machine Studies

International Journal of Reliability, Quality, and Safety Engineering

Journal of Safety Research

National Safety News

Professional Safety

Quality and Reliability Engineering

Reliability Engineering and System Safety

Reliability Review

Safety Science

User Modeling and User-Adopted Interaction (UMUAI)

1.4.2 Conference Proceedings

Proceedings of the Annual Conference on Computer Assurance (USA)

Proceedings of the Annual Reliability and Maintainability Symposium (USA)

Proceedings of the Annual Reliability Engineering Conference for the Electric Power Industry (USA)

Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (USA)

Proceedings of the International Conference on Reliability and Exploitation of Computer Systems (Poland)

Proceedings of the Symposium on Reliability in Electronics (Hungary)

1.4.3 Standards and Handbooks

ANSI/AIAA R-103, *Recommended Practice for Software Reliability*, American National Standards Institute (ANSI), New York.