
Contents

INTRODUCTION	ix
CHAPTER 1. SAFETY MANAGEMENT	1
1.1. Introduction.	1
1.2. Dependability	1
1.2.1. Introduction	1
1.2.2. Obstacles to dependability	3
1.2.3. Obstacles to dependability: case study	6
1.2.4. Safety demonstration.	7
1.2.5. Summary.	8
1.3. Conclusion	8
1.4. Bibliography	8
CHAPTER 2. FROM SYSTEM TO SOFTWARE	9
2.1. Introduction.	9
2.2. Systems of command and control.	10
2.3. System	13
2.4. Software implementation.	14
2.4.1. What is software?.	14
2.4.2. Types of software.	15
2.4.3. Software implementation in its context	15
2.5. Conclusion	16
2.6. Bibliography	17
2.7. Glossary.	17
CHAPTER 3. CERTIFIABLE SYSTEMS	19
3.1. Introduction.	19

3.2. Normative context	20
3.2.1. Generic standards	20
3.2.2. Railway sector	23
3.2.3. Automotive sector	33
3.2.4. Aviation sector	35
3.2.5. Aerospace	37
3.3. Conclusion	37
3.4. Bibliography	38
3.5. Glossary	41
CHAPTER 4. RISK AND SAFETY LEVELS	43
4.1. Introduction.	43
4.2. Basic definitions	43
4.3. Safety implementation	48
4.3.1. What is safety?	48
4.3.2. Safety management.	50
4.3.3. Safety integrity	57
4.3.4. Determining the SIL	59
4.3.5. SIL table	64
4.3.6. Allocating SIL.	65
4.3.7. SIL management	66
4.3.8. Software SIL.	67
4.3.9. Iterative process.	68
4.3.10. Identifying safety requirements.	68
4.4. In standards IEC 61508 and IEC 61511	70
4.4.1. Risk diagram.	71
4.4.2. LOPA.	73
4.4.3. Summary	74
4.5. Conclusions.	74
4.6. Bibliography	74
4.7. Acronyms	77
CHAPTER 5. PRINCIPLES OF HARDWARE SAFETY	79
5.1. Introduction.	79
5.2. Safe and/or available hardware	79
5.3. Reset of a processing unit	80
5.4. Presentation of safety control techniques	81
5.4.1. Error detection	81
5.4.2. Diversification.	88
5.4.3. Redundancy	88
5.4.4. Retrieval through error or error recovery	115
5.4.5 Partitioning	116

5.5. Conclusion	117
5.6. Bibliography	118
5.7. Glossary	119
CHAPTER 6. PRINCIPLES OF SOFTWARE SAFETY	121
6.1. Introduction.	121
6.2. Techniques to make software application safe.	121
6.2.1. Error management	122
6.2.2. Error recovery	124
6.2.3. Defensive programming	131
6.2.4. Double execution of the software application	138
6.2.5. Data redundancy	145
6.3. Other forms of diversification	149
6.3.1. Temporal diversity	149
6.3.2. Diversity in memory assignment.	149
6.4. Overall summary	150
6.5. Quality management	150
6.5.1. Introduction	150
6.5.2. Completing the software application	151
6.5.3. Completion cycle	152
6.6. Conclusion	155
6.7. Bibliography	156
6.8. Glossary	157
CHAPTER 7. CERTIFICATION	159
7.1. Introduction	159
7.2. Independent assessment	159
7.3. Certification	160
7.4. Certification in the rail sector.	161
7.4.1. Obligations.	161
7.4.2. Needs	162
7.4.3. Applying certification	162
7.4.4. Implementation	163
7.4.5. Upgrading management	164
7.4.6. Cross-acceptance	169
7.4.7. Tests	170
7.5. Automatic systems	171
7.6. Aircraft	171
7.7. Nuclear.	171
7.8. Automotive	172
7.9. Spacecraft	172
7.10. Safety case	172

7.11. Conclusion	173
7.12. Bibliography	174
7.13. Glossary	176
CONCLUSION	177
INDEX	179