SafeScrum® – Agile Development of Safety-Critical Software

Geir Kjetil Hanssen • Tor Stålhane •
Thor Myklebust

SafeScrum® – Agile Development of Safety-Critical Software

Springer

Geir Kjetil Hanssen
Software Engineering, Safety and Security
SINTEF Digital
Trondheim, Norway

Tor Stålhane
NTNU
Trondheim, Norway

Thor Myklebust
Software Engineering, Safety and Security
SINTEF Digital
Trondheim, Norway

ISBN 978-3-319-99333-1 ISBN 978-3-319-99334-8 (eBook)
https://doi.org/10.1007/978-3-319-99334-8

Library of Congress Control Number: 2018954543

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This book addresses the development of safety-critical software and proposes the SafeScrum® methodology. SafeScrum® is—as the name indicates—inspired by the agile method Scrum, which is extensively used in large parts of the software industry. Scrum is, however, not intended or made for safety-critical systems, hence we have proposed guidelines and additions to make it both practically useful and compliant with the additional requirements found in mandatory safety standards. We have specifically addressed the generic IEC 61508:2010 standard, part 3 (the software part), but this book will also apply to other, related domain-specific standards. Just like Scrum, SafeScrum® is to be considered a framework and not a fully detailed process suitable for all projects. This means that each case needs to consider adaptations of the framework to make it work optimally. The ideas and descriptions in this book are based on collaboration with industry, through discussions with assessment organizations, general discussions within the research fields of safety and software, but also on the authors' own judgements and ideas. Hence, SafeScrum® and this book do not necessarily represent the view or liability of any specific organization or individual.

Safety-critical systems are increasingly based on software, while established practice is often directed towards design and development of mainly hardware-based systems. While hardware-based systems call for a high level of details early in development since hardware is costly to alter, software can be managed more flexibly throughout development. This calls for new ideas on how software should be developed efficiently and how compliance with safety standards should be managed; we believe that agile methods will offer new opportunities to a domain facing new challenges.

This book provides basic knowledge on safety-critical systems with an emphasis on software. It provides an overview of agile software development, and how it may be related to safety, and it explains how to interpret and relate to safety standards. SafeScrum® is described in detail as a useful approach to gain the benefits of agile methods and is indented as a set of ideas and a basis for adaptation and adoption in industry projects. This covers roles, processes and process artefacts, and documentation.

We look into how standard software process tools may be taken into use. We provide insights into some relevant research in this new and emerging field and also provide some real-world examples.

Trondheim, Norway
June 2018

Geir Kjetil Hanssen
Tor Stålhane
Thor Myklebust

Acknowledgements

We would like to thank the Research Council of Norway for co-funding the work leading to this book, through the SUSS research project (#228431 Smidig Utvikling av Sikkerhetskritisk Software—Agile Development of Safetycritical Software). In collaboration with the authors, Børge Haugset has contributed with developing the SafeScrum® idea, and in particular with Chap. 10. We would also like to thank our project partners, Autronica Fire & Security and Kongsberg Maritime, that have contributed considerably to the shaping of SafeScrum®. We also want to thank several assessment organizations for taking part in discussions, in particular on how to interpret the IEC 61508:2010 requirements and guidelines. The International Electrotechnical Commission (IEC) has granted us re-print of important tables and details from the IEC 61508:2010 standard. Finally—and in particular—we are grateful for the support and valuable contributions by Ingar Kulbrandstad, Frank Aakvik, Jan-Arne Eriksen, Ommund Øgaard, Erik Korssjøen and Lars Meskestad.

Contents

1 Why and How You Should Read This Book 1
1.1 The Starting Point 1
1.2 Why Agile Software Development? 2
1.3 Why Should the Industry Consider Agile Methods? 3
1.4 What Do We Have to Offer? 5
1.5 Does It Work? 6
1.6 A Warning 8
1.7 Cooperation with Two TÜV Certification Bodies 8
1.8 What Next? 9
References 10

2 What Is Agile Software Development: A Short Introduction 11
2.1 Agility and Safety 11
2.2 Agile and Scrum in a Nutshell 11
2.3 Scrum and XP Concepts 13
2.4 Scrum Roles 14
2.5 Iterative and Incremental Development 15
References 15

3 What Is Safety-Critical Software? 17
3.1 IEC 61508:2010 17
3.2 On Safety-Critical Systems 18
3.3 RAMS in IEC 61508:2010 19
3.4 Security 23
3.5 Testing 25
3.6 Safety and Resilience 27
3.6.1 What Is Resilience? 27
3.6.2 A Resilient Development Process 28
3.6.3 A Resilient Organization 29
References 29

4 Placing Agile in a Safety Context 31
4.1 The Big Picture 31
4.2 Prioritizing 38
4.3 Development of Safety-Critical Software 39
4.4 The Role of Safety Culture 40
4.4.1 Introduction 40
4.4.2 What Is a Safety Culture 41
4.4.3 How to Build and Sustain a Safety Culture 42
4.4.4 A Site Safety Index 43
4.5 Information Items 44
4.6 Preparing for SafeScrum® 53
4.6.1 What Should Be Done 53
4.6.2 Introducing SafeScrum® 53
4.6.3 System Architecture 55
4.6.4 UML in Safety-Critical Software: Two Examples 56
4.6.5 Coding Standards and Quality Metrics 59
4.6.6 Configuration Management (CM) 60
4.6.7 Synchronizing SafeScrum® and a Stage-Gate Process 62
References 64

5 Standards and Certification 65
5.1 The Role and Importance of Standards 65
5.2 What the Standards are Not About 66
5.3 The Process of Product Certification 67
5.4 On Standards for Safety-Critical Software 67
5.5 Development Challenges Related to Safety Standards 69
5.6 The Developers' Responsibility 72
5.7 The Assessor's Responsibility 73
5.8 The Development Organization's Responsibility 73
References 74

6 The SafeScrum® Process 75
6.1 SafeScrum® in Perspective 75
6.2 An Iterative and Incremental Process 77
6.3 SafeScrum® and Associated Roles 77
6.4 Fundamental SafeScrum® Concepts 82
6.5 Preparing a SafeScrum® Development Project 84
6.5.1 Create Initial Documentation and Plans 84
6.5.2 Creating the Initial Product Backlog 88
6.5.3 User and Safety Stories 91
6.5.4 Setting Up the Team and Facilities 93
6.6 SafeScrum® Key Process Elements 94
References 95

7 The SafeScrum® Process: Activities 97
7.1 Sprint Planning Meeting 97
7.1.1 Defining the Sprint Goal 98
7.1.2 Clarifying Team and Commitment for the Sprint 98
7.1.3 Creating the Sprint Backlog 98
7.2 Sprint Workflow 99
7.2.1 Resolving Stories 99
7.2.2 Peer Review of Code (Pull Request) 99
7.2.3 Quality Assurance of the Code 99
7.3 Sprint Review Meeting 100
7.4 Sprint Retrospective 101
7.5 The Daily Stand-Up 102
7.6 Backlog Refinement Meeting 103
7.7 Additional Quality Assurance 103
7.7.1 Coding Standard and Quality Metrics 104
7.7.2 Code Documentation Coverage 106
7.7.3 Unit Test Coverage 107
References 107

8 SafeScrum® Additional Elements 109
8.1 Traceability 109
8.2 Change Impact Analysis 111
8.2.1 Introduction 111
8.2.2 Requirement Changes 112
8.2.3 Design and Code Changes 112
8.2.4 Minor Safety Issues 113
8.2.5 Major Safety Issues 114
8.3 Testing 115
8.3.1 Classes of Tests 115
8.3.2 Unit Testing 115
8.3.3 Software Integration Testing 117
8.3.4 Software Module Testing 118
8.3.5 Safety Testing 118
8.3.6 Back-to-Back Testing 121
8.4 Safety Engineering 123
8.4.1 Safety Analysis 123
8.4.2 Agile Hazard Log 123
8.4.3 Agile Safety Cases 126
8.4.4 Constructing Safety Cases 128
8.5 Managing Releases 131
8.5.1 Introductions 131
8.5.2 Internal Releases 132

8.5.3 External Releases: Deployment . . . 132
8.5.4 Release Challenges . . . 133
References . . . 134

9 Documentation and Proof-of-Compliance . . . 135
9.1 Introduction . . . 135
9.2 Trust . . . 136
9.3 Requirements Related to Documentation . . . 137
9.3.1 Reuse and the use of Templates . . . 137
9.3.2 Method When Evaluating IEC 61508-1:2010 Documentation Requirements . . . 138
9.3.3 IEC 61508-1:2010 Walkthrough of Chap. 5 "Documentation" . . . 138
9.3.4 IEC 61508-3:2010 Walkthrough of the Normative Annex A . . . 140
9.4 Classification of the Documentation . . . 141
9.5 Discussion . . . 142
References . . . 144

10 Tools . . . 145
10.1 Introduction . . . 145
10.2 Tool Classification According to IEC 61508:2010 . . . 146
10.3 Tool Chains and Agile Development . . . 147
10.4 Special Considerations for a Safety-Critical Tool Chain . . . 147
10.5 Process Tools . . . 148
10.5.1 Workflow . . . 148
10.5.2 Scrum and Process Traceability . . . 149
10.5.3 Design and Code Documentation . . . 150
10.5.4 UML Models . . . 150
10.6 Test and Analysis Tools . . . 150
10.7 Generic Tools and Their Classification Level . . . 151
Reference . . . 151

11 Adapting SafeScrum® . . . 153
11.1 Adapting SafeScrum® . . . 153
11.2 SafeScrum® for the Process Domain: IEC 61508:2010 . . . 154
11.2.1 The Adaptation . . . 154
11.2.2 The SafeScrum® Approach to IEC 61508:2010 . . . 156
11.3 SafeScrum® for the Avionics Domain: DO 178C:2012 . . . 158
11.4 SafeScrum® for the Railway Domain: EN 50128:2011 . . . 161
11.4.1 Adaptation . . . 161
11.4.2 The SafeScrum® Approach to EN 50128:2011 . . . 161
References . . . 165

12 A Summary of Research . . . 167
12.1 Introduction . . . 167
12.2 Requirements . . . 169
12.3 Testing . . . 171

12.4 Code Refactoring . . . 175
12.5 Continuous Integration and Build . . . 175
12.6 Iterative Process . . . 176
12.7 Customer Involvement . . . 178
12.8 Planning . . . 180
12.9 Traceability . . . 182
12.10 The Near Future: DevOps . . . 184
References . . . 184

13 SafeScrum® in Action: The Real Thing . . . 187
13.1 Introduction . . . 187
13.2 Planning the Work . . . 188
13.3 The Workflow . . . 191
13.4 Sprint Review Meeting . . . 194
References . . . 194

Annexes A–D . . . 195
Annex A: Necessary Documentation . . . 195
Annex B: A Short Introduction to Safety Analysis . . . 199
B.1 Background . . . 199
B.2 Participants . . . 200
B.3 On Safety Analysis in SafeScrum® . . . 200
B.4 Probability and Consequences . . . 204
B.5 Generic Failure Modes and Hazard Lists . . . 205
B.6 PHA: Preliminary Hazard Analysis . . . 205
B.7 FMEA: Failure Mode and Effect Analysis . . . 206
B.8 IF-FMEA: Input Focused FMEA . . . 210
B.9 FFA: Functional Failure Analysis . . . 211
B.10 HazId: Hazard Identification . . . 212
B.11 Hazard Stories . . . 215
B.12 FMEDA: Failure Mode Effect and Diagnostics Analysis . . . 217
B.13 FTA: Fault Tree Analysis . . . 219
B.14 Hazards Under No–Fault Conditions . . . 220
Annex C: Useful UML Diagrams . . . 221
Annex D: Analyses Required by IEC 61508:2010 . . . 225
References . . . 226

Glossary . . . 229

Index . . . 231