Stefan Biffl · Matthias Eckhart
Arndt Lüder · Edgar Weippl *Editors*

Security and Quality in Cyber-Physical Systems Engineering

With Forewords by
Robert M. Lee and Tom Gilb

Springer

Security and Quality in Cyber-Physical Systems Engineering

Stefan Biffl • Matthias Eckhart • Arndt Lüder •
Edgar Weippl
Editors

Security and Quality in Cyber-Physical Systems Engineering

With Forewords by
Robert M. Lee and Tom Gilb

Springer

Editors
Stefan Biffl
Institute of Information Systems Engineering
Technische Universität Wien
Vienna, Austria

Matthias Eckhart
Institute of Information Systems Engineering
Technische Universität Wien
Vienna, Austria

Arndt Lüder
Institute of Ergonomics, Manufacturing Systems and Automation
Otto von Guericke University Magdeburg
Magdeburg, Germany

Edgar Weippl
Institute of Information Systems Engineering
Technische Universität Wien
Vienna, Austria

ISBN 978-3-030-25311-0 ISBN 978-3-030-25312-7 (eBook)
https://doi.org/10.1007/978-3-030-25312-7

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

Foreword on Security in Systems Engineering

Fear less, do more. Four simple words that call into focus a reality of our industrial automation field and its nexus with our cybersecurity community. There are real concerns for the cybersecurity of our infrastructure and the industrial systems that they rely upon. There are risks that have been well discussed for the last few decades that are intrinsic to the systems and their designs without outside influence such as faults, errors in system design, and failure of protective and safety functions when they are needed most. There are also risks from outside influences such as malicious adversaries who seek to abuse the systems and their functionality for nefarious purposes. As we learn more about our systems and their evolution as well as our adversaries and their capabilities, it is natural to fear. The consequence can be enormous even though the frequency of impact seems minimal. The entirety of the modern world seems in the balance. So why not gravitate towards fear? Simply put, because our fields of engineering and cybersecurity have done an amazing job and we must appreciate the situation we are in and the advancements that are being made. Yet, we still must realize that the risk is growing, and we must do more to protect our systems. To play to those four opening words I will briefly go through a few case-studies relevant to this collection of manuscripts to ideally set the tone and importance of the works contained in this book.

BTC Pipeline Explosion

In 2008, a Russian cyber-attack pivoted through Internet-connected camera systems running along the Baku-Tbilisi-Ceyhan (BTC) gas pipeline to shut down alarms and over pressurize the pipeline, which resulted in a massive explosion.[1] This story was revealed by Bloomberg news in December 2014 and was on its path of being the first

[1] https://www.bloomberg.com/news/articles/2014-12-10/mysterious-08-turkey-pipeline-blast-opened-new-cyberwar

ever confirmed case of a cyber-attack causing damage or destruction on industrial control systems (ICS). Except, it was not confirmed, nor did it hold up to scrutiny. Almost immediately after the story was published, I started to suspect elements of the story as did a few others in the information security community while the story took on its own life and spread throughout the community.[2] Eventually, Michael Assante, Tim Conway, and I wrote a whitepaper at the SANS Institute noting that many details of the story conflicted with reality and in some ways each other of what such an attack could look like.[3] Later in 2015 a well-researched article in *Sueddeutsche Zeitung* revealed even more of the details about the attack were wrong in comparison to how the pipeline was operated at the time; as an example the camera systems that the adversaries supposedly pivoted through during the attack were not installed until after the attack in response to the explosion.[4]

The BTC pipeline case-study has been captured for years as a real event even though it was debunked by high-profile security experts nearly immediately upon publication. It has been referenced in numerous conferences, academic journals, and even U.S. Congressional testimony. It is otherwise a tantalizing story and to many serves as an example of why we must do more in cybersecurity. The reality of course is that cyber-attacks on gas pipelines are possible. The scenario described is possible in principle. Security is important to safety. However, relying on hyped-up case-studies to make a point is only self-serving. The dedication of resources and studying of attacks to develop defenses and best practices are best suited for real attacks where the study can yield meaningful results. Hyped-up threats only yield results in the wrong direction. Even when there are real threats, we often see the impacts overblown.

Bowman Avenue Dam Infiltration

In 2013, Iranian hackers broke into the Bowman Avenue Dam near Rye Brook, New York. They gained access to a human–machine interface (HMI) and read water levels off the dam.[5] The U.S. Intelligence Community identified the infiltration and informed the U.S. Department of Homeland Security, which kicked off a series of events to identify and respond to the infiltration. The press would later become aware of this infiltration in 2015 and publish on it. Major news outlets covered the story noting that it was a small dam, but significant damage could have been done if only the hackers had the intent. Some argued they had the intent but simply could not manage the attack due to their technical incompetence. Headlines ran with pictures of

[2] https://www.flyingpenguin.com/?p=20958

[3] https://ics.sans.org/media/Media-report-of-the-BTC-pipeline-Cyber-Attack.pdf

[4] https://ics.sans.org/blog/2015/06/19/closing-the-case-on-the-reported-2008-russian-cyber-attack-on-the-btc-pipeline

[5] https://www.cnn.com/2015/12/21/politics/iranian-hackers-new-york-dam/index.html

nuclear meltdowns and major dams breaking with government and industry officials noting this was an example of things to come. Later, the U.S. Department of Justice would indict the hackers and note that they were intending something more nefarious, but the automated controls were down for maintenance that would have allowed them to do their actions. The case was real, there was a lot of hype around it, but many of the technical nuances of the details were not correct.

I closely examined the case and all the details around it including interviews with those involved in New York as well as the U.S. Department of Homeland Security. There were key omissions that were never captured fully. I wrote about the most significant of these in a whitepaper with Michael Assante and Tim Conway, where we covered that the automated controls for the dam were never actually installed by the time the intrusion occurred due to delays in the project.[6] The Iranian hackers had not performed some amazing feat but instead had found an HMI that was remotely connected to the Internet via a cell card with poor authentication. After accessing the system, the Iranians were unable to do anything with it because of the lack of controls, whether they would have intended such, and left. The dam itself was small and handled runover during heavy rains. Damaging it would have resulted in making some people's shoes wet, not the imminent doom and images of nuclear meltdowns that the media captured. Even though the infiltration was real, the details around the case and the potential impact were captured incorrectly. It is important to capture such details properly especially for when attacks are real as such cases can have profound impact on public perception as well as defense lessons learned.

The 2015 and 2016 Cyber-Attack on the Ukraine Electric Grid

In December 2015,a team of adversaries broke into three Ukrainian distribution level power companies. The attack started with malware, known as BlackEnergy3, inside the enterprise information technology (IT) networks but that was just a foothold for the adversary. The real work of the attack was nearly 6 months the adversaries spent inside the operations technology (OT) networks learning the systems and operations of the power companies. The attack was effectively just the adversaries learning how to operate the equipment inappropriately. It was not a focus on exploits and vulnerabilities or malware but instead was the abuse of legitimate functionality and features in the environment for malicious purposes.[7]

The attack ultimately led to around 6 h of outages across 225,000 customers. While this does not seem like a lot, it was the first ever cyber-attack to lead to a power outage. More impactfully, the system operators did not have access to their automation networks and supervisory control and data acquisition (SCADA) systems

[6] https://ics.sans.org/media/SANSICS_DUC4_Analysis_of_Attacks_on_US_Infrastructure_V1.1.pdf

[7] https://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/

for nearly a year due to the attack. The recovery was a significant effort. Michael Assante, Tim Conway, and I led an investigation on the attack and in our report to the industry we highlighted that what happened in Ukraine was not unique to Ukraine; the attack could be replicated elsewhere. Moreover, we warned that there were elements of the attack that indicated there could be a follow-on attack.[8]

In December 2016, that follow-on attack manifested in the form of malware known as CRASHOVERRIDE, that led to a loss of power in Kiev, Ukraine, at a transmission level substation. The outage was only about an hour long but the amount of electric power lost was three times the total amount lost in all of the Ukraine 2015 attack due to the differences in distribution versus transmission level substations.[9] My team at Dragos, Inc. wrote the report on this malware drawing specific attention to the fact that the adversary effectively learned from their attack in 2015 and moved the work that took dozens of people into scalable software, CRASHOVERRIDE, that would only take the adversary positioning and activating it.[10] This effort represented a scalable repeatable attack on electric power.

What More Looks Like

There are plenty of high-profile attacks that target OT and ICS environments. It is hard to ignore such attacks especially as we are seeing more that are dangerous such as the TRISIS malware. TRISIS was leveraged in Saudi Arabia to shut down a petrochemical plant while its true design was likely to kill people via disabling the safety system and a follow-on attack against the petrochemical process.[11] The attack failed to kill anyone but managed to shut down the plant costing the site millions. The adversary, code named XENOTIME, remains active as of the time of this publication, targeting other industrial sites around the world.[12]

There are even more incidents that are below such thresholds and thus remain out of the media and remain private to the companies who work those cases. It is easy to see then why people fear. However, I will return to the initial point of this foreword, which is to also highlight that our infrastructure is reliable, our engineering practices sound, and our community is amazing. Far more is done for security than will ever make headlines; the community is quick to notice attacks but slow to see the day-to-day work of defenders around the world. The adversaries are becoming more aggressive, but they must contend with physics and when organizations prepare correctly, they must also contend with defenders who protect our systems.

[8] https://ics.sans.org/media/E-ISAC_SANS_Ukraine_DUC_5.pdf

[9] https://www.wired.com/story/crash-override-malware/

[10] https://dragos.com/wp-content/uploads/CrashOverride-01.pdf

[11] https://dragos.com/wp-content/uploads/TRISIS-01.pdf

[12] https://dragos.com/resource/xenotime/

These defenders thrive when the systems are made to be more defensible. If we are not to fear, but instead are to do more, then it is natural to ask what more looks like. To me, it is in contributions such as this book that we find that answer. This book represents an amazing effort by community members, engineers, and academics who are striving to create more defensible systems through better engineering, system design, and implementation of those systems. These quality and security improvements of long-running technical systems such as ICS will yield a safer and more reliable world even in the face of determined adversaries. It is in this type of collaboration that lessons learned can be documented and made available to current and future practitioners.

Gambrills, MD, USA Robert M. Lee

Foreword on Quality in Systems Engineering

Systems Quality Engineering: Some Essential Steps Forward

As citizens in a connected world, the quality of our daily lives depends on critical infrastructure, such as the Internet and energy and transportation networks, and on industrial production systems, the so-called *Cyber-Physical Systems* (CPSs), which we expect to provide high-quality services economically and safely for their environment. In an increasingly networked world, the quality of the services that these systems provide depends on their security against intended and unintended wrongdoing, both during the operation of the system and during the engineering that designs the system.

Therefore, the topic of this book, *Security and Quality in Cyber-Physical Systems Engineering*, is timely and important, but may sound somewhat puzzling, as security is among the qualities of systems engineering. However, the topic makes sense as security in systems engineering is a fundamental quality, since there is no safety of systems operation without information security in systems engineering. Unfortunately, the quality *security* is not well understood and often not well addressed in systems engineering practice. Therefore, I share in this foreword my view on essential steps and principles to improve *systems quality engineering* both in practice and in university teaching.

I will keep my observations to a minimum by sticking to the very basic ideas of *quality engineering of products and services*. References give considerable detail background for the interested reader.

Security and Safety Are Quality Requirements Security and safety are part of the set of system attributes known as *qualities*. Consequently, the systems engineering methods for quality in general help address *Security* and *Safety*. For example, for Usability, Maintainability, Reliability, and Availability, I define *quality* as the attributes that describe *how well* a system functions (Gilb 2005). In my long consulting experience on quality requirements, I have observed that all qualities are variable in their performance levels. Therefore, we can, and must, *quantify all system quality requirements* as a fundamental aspect of any systems engineering

methods. While this may sound obvious, even systems engineering subjects taught at university, such as *Quality Function Deployment* (QFD), allow unquantified qualities but should be applied more rigorously by *emphasizing the quantification of quality requirements* (Gilb 2005, 2018b).

Quantification of Quality Requirements All quality attributes vary and can be expressed as a quantified requirement (Gilb 2005). A simple method to see that this is widely understood, and somewhere practiced, is to Google a quality name followed by the term *metrics*, e.g., *Usability Metrics*, *Security Metrics*.

Quantification of a quality is not the same as measurement, but quantification is the basis for measurement (quality level testing) and other applications. The essential notion of quality quantification is the definition of a *scale of measure*.

For *Security,* an example *Scale* could be *the share (%) of assessed cyber risks with an event likelihood greater than X% and an impact greater than $Y.*

The *Scale* parameter (Gilb 2005) both defines the quality and enables assigning numeric levels to the quality, for a variety of purposes, such as:

1. *Establishing benchmarks* for that quality, in our own system, competitive systems, and past and future performance levels (Gilb 2005)
2. *Establishing scalar constraint* quality requirements, that is, worst acceptable level of the quality for a purpose (Gilb 2005)
3. *Establishing target levels* of the quality dimensions, such as time, space, and conditions
4. *Making estimates* of the impacts of design ideas on the requirement levels
5. *Measuring actual levels* of the quality in real systems
6. *Bidding, costing, and contracting* using these quality levels

Estimation of Design Quality Impact When considering the design for meeting the required quality levels, we must be numeric, logical, and look at the whole picture (all quality requirements, resource budgets, and constraints). Certainly we cannot afford to discuss or evaluate any single design idea solely in a single quantified dimension, such as security or safety. Therefore, the evaluation of the impact of any given design idea, needs to be quantitative as foundation for combining the impact values. Figure 1 illustrates the *Impact Estimation* principle as a conceptual table for assessing the impact of design ideas that, together, have an impact on the project objectives and resources.

1. An estimate of the *expected degree* that the design will satisfy the *constraint* levels, with regard to all concurrent conditions (project deadlines, budgets, constraints, reaching other quality levels, and more);
2. An estimate of the *expected degree* to which the design will satisfy the *target* levels, with regard to all concurrent conditions (project deadlines, budgets, constraints, reaching other quality levels, and more).
3. The estimates need to document the ranges of experience, evidence for the levels, and sources of the levels (*Competitive Engineering* (CE) (Gilb 2005), *Impact Estimation* (Gilb 2008)), for quality control, for responsibility, and for understanding the quality of the information.

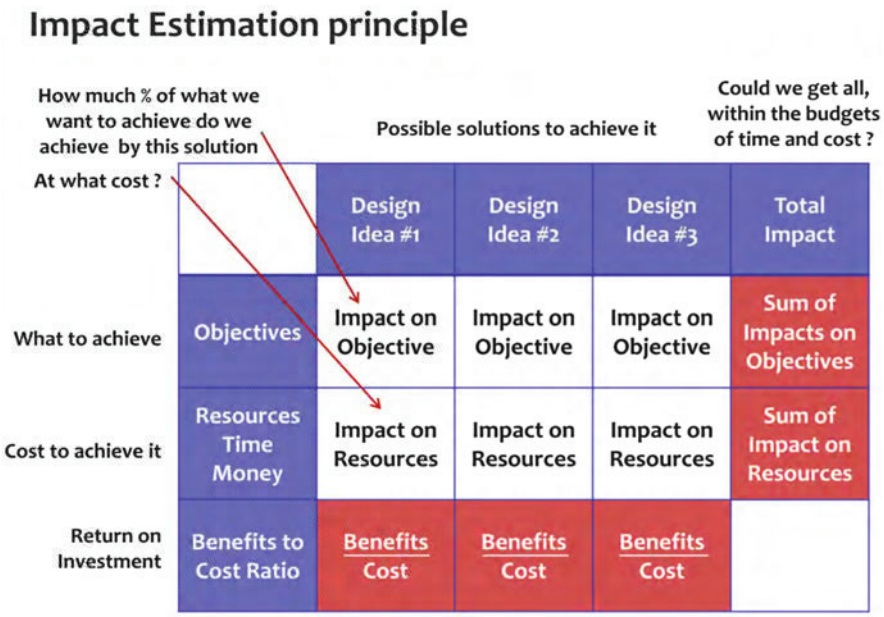

Fig. 1 A conceptual view of the analysis of quality designs

4. The estimates need to be made in a way that provides at least a rough picture of the *aggregate effects of the designs* (see Fig. 1) (a) together with all other complementary or concurrent designs and (b) with regard to the potential of together meeting the final quality level requirements, preferably with an engineering safety factor.

Measurement of Design Quality, and Correction Once a design is estimated and found to be worthy of at least experimental implementation, we need to measure, at least roughly, not necessarily exactly, how well the design performs, in at least one dimension; and possibly in side effects and cost dimensions as well. If the performance deviates negatively from the expected results, then *root cause analysis* should be used immediately at that increment, and redesign made to get back on track. This process is eminently described by Quinnan in *Cleanroom* (Quinnan 1980) and is an inherent part of my *Evo* approach (Gilb 2005). This is *good engineering agile*, not to be confused with current popular software agile methods, which have no consciousness of qualities, engineering, design, architecture, or dynamic design to cost. When a given *quality constraint* level is reached in the system development process, there is an opportunity to trigger contractual minimum payments. When the *quality target* levels are reached then this can be used to target full payment, and to stop engineering or designing those fully delivered qualities.

Collecting Design Information on Engineering Component Candidates It is a long-standing engineering tradition to organize engineering knowledge about potential

design components and processes in engineering handbooks, which include data, tables, etc. regarding their expected attributes. The *World Wide Web* provides convenient means for collecting, accessing, and updating such attribute knowledge. If this knowledge is of good quality, and easily available, then it can be used to make estimates. If it is not good, then the next best thing for those who believe in an idea is to experiment with small-scale incremental implementation and measurement. This might, in fact, provide more useful data on the real system than a general engineering handbook. My father, an engineer and inventor, stressed to me that the engineering tables were not to be blindly trusted, but are likely to be useful in situations without better data. My early books on *Design Engineering* (Gilb1976; Gilb and Weinberg 1977–1978) attempted to show how this might be done for qualities of data structures. One early practical tool-building experiment (later a PhD; see slide 4 in (Gilb 2017) showed how a computer (Apple II, Forth 1979) could automatically pick the best design, if given quantified quality requirements and quantified design component attributes.

Quality Engineering Processes There is a large number of known, and to be invented, engineering processes, which can contribute to the engineering of secure and high-quality products. The important idea regarding a process, as with any system design aspect, are the quality and performance attributes and the resource costs of that process. These quality, performance, and cost attributes could be systematically organized in handbooks, preferably on the Internet for access and updating. Both research and practice could be incorporated. As an example, my *Specification Quality Control* (SQC) process has been studied (Gilb 2005) as a method for measuring the defect quality levels of requirements written in my *Planguage* (Gilb 2005). A key finding by Terzakis from *Intel* (Terzakis 2013) was that *Planguage* together with SQC resulted in a 98% defect reduction in submitted requirements, and in a 233% engineering productivity improvement. Release defects went down to 0.22 defects per 400 (or 600) words, a suitable value for the very high quality levels required for engineering *Intel* chips. Terzakis currently measures the effects of *coaching* on *quality of requirements* (unpublished 2018). My *Planguage* for specification was adopted by over 21,000 *Intel* engineers, demonstrating its viability for embedded hardware logic (Erik Simmons, *Intel*). Therefore, *Competitive Engineering* and *Planguage* (Gilb 2005) should be applied for improving the maturity of engineering *Complex-Cyber-Physical Systems*.

Preventative and Evidence-Based Quality Processes Some of the most interesting engineering processes take a *lean* approach of preventing defects and problems by using common-cause *root-cause analysis* at the grass roots and frequent organization *process and conditions* changes in order to measurably improve organizations' engineering process quality. My favorite generic process is the *Defect Prevention Process* (DPP) developed at IBM (Gilb and Graham 1993; Dion et al. 2018). DPP is a good example of a very generic, organizational engineering improvement, with clear repeatable economic and quality effects that led to lasting results. Unfortunately, the DPP is rarely taught in engineering, or quality engineering, courses, similar to the IBM *Cleanroom* approach (Quinnan 1980). Over the years, I

have often seen new ideas advocated without quantitative evidence. I strongly argue that researchers and teachers should build more on proven methods with strong quantitative evidence. Without collecting and sharing data on quality, performance, and cost attributes as evidence on new and traditional processes, systematic scientific progress is not possible. The similar weakness to improve systems engineering practice systematically is reflected in the high project failure rate, which has remained almost constant for decades. Therefore, preventative and evidence-based quality processes should be applied in engineering software-intensive *Complex-Cyber-Physical Systems*.

Integrated Consideration of Requirements I see the integration of quality requirements, security requirements, and all other requirements as essential. It seems risky to limit the engineering evaluation of security and other quality designs to those areas in an isolated way. We cannot allow specialist "security" and "reliability" engineers to make design decisions with impact on a large system, without balanced due regard for other critical factors in the system. Although it is difficult, we must aim at understanding the side effects between all requirements, including example requirements on performance, costs, and non-quality values, such as image and trust or technical debt. In short, security and quality are important parts of the larger systems picture, and have to be properly incorporated in this picture both in practice and in academic research.

Demand for a Stronger Engineering Culture in Software-Intensive Systems Hardware engineering has a mature engineering culture, but is under constant attack from technological change. Systems engineering has also shown engineering culture with good maturity in the face of complexity and change, for example, in space and military application areas. Unfortunately, the newcomer, software, including *logicware*, *dataware*, *peopleware*, *netware*, is, arguably, mostly not engineered in a sufficiently mature *engineering discipline*, but still seems to be all about programming. I have seen this problem for decades.

Hardware experts can engineer a 99.98% available system. In contrast, low-quality software is widely accepted, often with the argument that software can be easily changed. Nobody would buy a machine that seems buggy, but users and lawmakers consider defects in software acceptable if software updates are promised. I see a major reason for this problem in politicians, managers, and researchers not demanding the same quality-of-engineering for IT and software, such as software-intensive systems, as documented in serial aircraft accidents that recently made the news. Therefore, I want to raise awareness for demanding sufficiently high standards for requirements, including security and software quality, for the integrated engineering of software-intensive systems, such as critical infrastructure and industrial systems.

Finally, I summarize my *Basic Principles of Serious Quality Engineering* (Gilb 2018a):

1. All qualities must be treated quantitatively, at all times (Gilb 2005).
2. All other requirements need to be defined rigorously, too.
3. Design options need clear detailed definition, and probably decomposition.

4. All design decisions can, and normally should, be estimated, before selection or prioritization, with regard to their possible and probable impacts on all critical qualities and costs [see Chap. 9 on IET in Gilb (2005)].
5. Incremental quality impacts of designs need to be measured, at that increment (Quinnan 1980).
6. When incremental designs measurably fail to deliver expectations, they need to be immediately replaced or corrected (*Cleanroom*) (Quinnan 1980).
7. Reasonable rigorous quality-control measurements of critical specifications must be carried out with numeric exit levels to determine the economic release level (SQC, Intel; Terzakis 2013).
8. Continuous grassroots analysis of engineering work processes must result in a continuous stream of measurable positive improvement (DPP) (Gilb and Graham 1993; Dion et al. 2018).
9. Technical management (CTO, etc.) need to demand and enforce these principles.
10. Universities need to teach these principles and need to organize the knowledge internationally.

Overall, the topics in this book, *Security and Quality in Cyber-Physical Systems Engineering*, are important and should be complemented with a strong vision on integrated systems engineering that builds on systems quality engineering, according to my *Basic Principles of Serious Quality Engineering* (Gilb 2018a), by quantifying requirements for quality, including security; by collecting and using quantitative evidence on design options to select suitable engineering processes, methods, and tools; and by improving quality based on comparing the quantitative evidence from ongoing projects with the planned constraint and target levels of requirements. My emphasis of quantitative evidence is rooted in empirical scientific principles and has shown to be practical and useful in real-world systems engineering contexts (Terzakis 2013). Therefore, readers of this book can benefit from combining these principles with the lessons on requirements of systems engineering in Part I of this book, on quality improvement approaches in Part II, and on security in engineering in Part III of this book, for improving systems engineering in their academic or practical environment.

Kolbotn, Norway Tom Gilb

References

Dion, K. et al. (2018). https://figshare.com/articles/Raytheon_Electronic_Systems_Experience_in_Software_Process_Improvement/6582863. This gives numeric industrial aspects of the use of *Inspection, and defect prevention processes* over 8 years. They include reducing rework from 43% to under 5%.

Gilb, T. (1976). *Design engineering*. This book was organized to store quantified qualities of data elements and structures. A dataware engineering handbook.

Gilb, T. (2005). Competitive engineering. In *A handbook for systems engineering, requirements engineering, and software engineering using Planguage*. Butterworth: Elsevier. Retrieved from https://www.gilb.com/p/competitive-engineering. This book defines a planning language for quality management, and associated quality processes.

Gilb, T. (2008). *How problems with quality function deployment's (QFD's) house of quality (HoQ) can be addressed by applying some concepts of impact estimation* (IE). http://www.gilb.com/DL119

Gilb, T. (2017). *Ten suggested principles for human factors systems engineering*. http://concepts.gilb.com/dl911. Keynote at WUD (Worldwide Usability Day) Silesia, Katowice Poland, 9 December 2017; Details about the 'Aspect Engine' on slide 4.

Gilb, T. (2018a). *Hundred practical planning principles*. Booklet. https://www.gilb.com/store/4vRbzX6X

Gilb, T. (2018b, December 01). *How to plan for the unknown* about 45 slides keynote at WUD Conference Kawice Poland. Includes Ohno on fake news lean. Americans and happiness index Poland 42, and ten principles of dealing with unknown. http://concepts.gilb.com/dl935

Gilb, T., & Graham, D. (1993). *Software inspection*. See two chapters on *Defect prevention process*. See the many case studies for the industrial attributes of software inspection, and of non-software industrial inspection of specifications.

Gilb, T., & Weinberg, G. (1977–1978). *Humanized input*. This book was organized to store quantified qualities of data elements and structures. A dataware engineering handbook.

Quinnan. (1980). *Mills and Quinnan slides*. http://concepts.gilb.com/dl896

Terzakis, J. (2013). *The impact of requirements on software quality across three product generations*. 21st IEEE international requirements engineering conference (RE), Rio de Janeiro (pp. 284–289). https://www.thinkmind.org/download.php?articleid=iccgi_2013_3_10_10012terzakis

Preface

Sitting at the Berlin Tegel airport, waiting for a flight to Vienna, the preparation of the book at hand provides us with some concerns. Only weeks ago, another *Boeing 373 Max8* airplane had crashed. Following the communication of major air traffic safety organizations, a candidate reason for the accidents was a misleading combination of software and hardware in the airplane, leading to unintended airplane behavior that the crew had not been able to compensate. As a traveler you ask yourself: how can such dangerously misleading combinations occur in a safety-conscious environment?

Modern airplanes (as most large technical systems ranging from trains and airplane systems to power plants and factories) are *complex cyber-physical systems*. They become software-intensive technical systems from combining physical system hardware, such as jet engines, wings, and flaps, with control software assisting the pilots. Such complex cyber-physical systems are developed in large engineering organizations by executing complex engineering processes. Within these processes, several engineering artifacts developed in parallel describe together the architecture and behavior of the intended technical system. In the engineering organization and processes, several engineering disciplines provide their special skills to the overall success of the engineering project.

Even if each involved engineer follows a discipline's best practices, still inconsistencies, incompatibilities, unclear communication, or even errors may occur, may reduce the engineering quality and, in the worst case, may result in an operational disaster, such as the recent *Boeing 373 Max8* incident. Usually, an incident is not intended, but there are cases where malicious acts are performed by individuals, who are interested in causing engineering projects or the developed technical systems to fail.

Do we have a chance to protect engineering organizations against cyber threats and to ensure engineering project quality? Answers to these questions will be given in the book at hand. Therefore, the book contains three parts that logically build up on each other. The first part discusses the structure and behavior of engineering organizations for complex cyber-physical systems. This part provides insights into processes and engineering activities executed and highlights requirements and bordering conditions for secure and high-quality engineering. The second part addresses quality

improvements with a focus on engineering data generation, exchange, aggregation, and use within an engineering organization and the need of proper data modeling and engineering result validation. Finally, the third part considers security aspects concerning complex cyber-physical systems engineering. Chapters of the last part cover, for example, security assessments of engineering organizations and their engineering data management (including data exchange), security concepts and technologies that may be leveraged to mitigate the manipulation of engineering data, and discussions of design and run-time aspects of *secure complex cyber-physical systems*.

After reaching Vienna with a safe flight in an *Airbus 319* and sitting in the next *City-Airport-Train*, another *complex cyber-physical system*, we are sure that reading this book can reduce the concerns we had in Berlin and can assist engineers and decision makers, researchers, and practitioners in setting up and improving secure and high-quality engineering processes in appropriate engineering organizations.

Magdeburg, Germany Arndt Lüder

Contents

Contributors

Christos Alexakos Industrial Systems Institute, ATHENA Research Center, Patras, Greece

Florian Auer University of Innsbruck, Innsbruck, Austria

Falko Bendik Otto-v.-Guericke University/IAF, Magdeburg, Germany

Stefan Biffl Institute of Information Systems Engineering, Technische Universität Wien, Vienna, Austria

Bernhard Brenner SBA Research, Vienna, Austria

Ambra Calà Siemens AG, Erlangen, Germany

Violeta Damjanovic-Behrendt Salzburg Research, Salzburg, Austria

Matthias Eckhart Christian Doppler Laboratory for Security and Quality Improvement in the Production System Lifecycle (CDL-SQI), Institute of Information Systems Engineering, Technische Universität Wien, Vienna, AustriaSBA Research, Vienna, Austria

Andreas Ekelhart Christian Doppler Laboratory for Security and Quality Improvement in the Production System Lifecycle (CDL-SQI), Institute of Information Systems Engineering, Technische Universität Wien, Vienna, AustriaSBA Research, Vienna, Austria

Michael Felderer University of Innsbruck, Innsbruck, Austria

Apostolos P. Fournaris Industrial Systems Institute, ATHENA Research Center, Patras, Greece

Martin Fruhmann FH JOANNEUM GmbH, Institute of Internet Technologies & Applications, Kapfenberg, Austria

Klaus Gebeshuber FH JOANNEUM GmbH, Institute of Internet Technologies & Applications, Kapfenberg, Austria

Matthias Gusenbauer SBA Research, Vienna, Austria

Kristofer Hell Siemens Mobility GmbH, Braunschweig, Germany

Aljosha Judmayer SBA Research, Vienna, Austria

Athanasios P. Kalogeras Industrial Systems Institute, ATHENA Research Center, Patras, Greece

Lukas Kathrein Christian Doppler Laboratory for Security and Quality Improvement in the Production System Lifecycle (CDL-SQI), Institute of Information Systems Engineering, Technische Universität Wien, Vienna, Austria

Ismail Khalil Johannes Kepler Universität, Linz, Austria

Peter Kieseberg University of Applied Sciences, St. Poelten, Austria

Andreas Komninos Industrial Systems Institute, ATHENA Research Center, Patras, Greece

Christos Koulamas Industrial Systems Institute, ATHENA Research Center, Patras, Greece

Aris S. Lalos Industrial Systems Institute, ATHENA Research Center, Patras, Greece

Arndt Lüder Otto-v.-Guericke University/IAF, Magdeburg, Germany

Johannes Lutz Siemens Mobility GmbH, Braunschweig, Germany

Alexandra Mazak Christian Doppler Laboratory for Model-Integrated Smart Production (CDL-MINT), WIN-SE, JKU Linz, Linz, Austria

Rostami Mehr Volkswagen AG, Wolfsburg, Germany

Kristof Meixner Christian Doppler Laboratory for Security and Quality Improvement in the Production System Lifecycle (CDL-SQI), Institute of Information Systems Engineering, Technische Universität Wien, Vienna, Austria

Mathias Mühlhause Siemens Mobility GmbH, Braunschweig, Germany

Petr Novak Czech Technical University, Prague, Czech Republic

Johanna Pauly Otto-v.-Guericke University/IAF, Magdeburg, Germany

Felix Rinker Christian Doppler Laboratory for Security and Quality Improvement in the Production System Lifecycle (CDL-SQI), Institute of Information Systems Engineering, Technische Universität Wien, Vienna, Austria

Ronald Rosendahl Otto-v.-Guericke University/IAF, Magdeburg, Germany

Barbara Russo Free University of Bozen-Bolzano, Bolzano, Italy

Marta Sabou Technische Universität Wien, Vienna, Austria

Ina Schaefer Technische Universität Braunschweig, Braunschweig, Germany

Thomas Schäffler Siemens AG, Erlangen, Germany

Alexander Schlie Technische Universität Braunschweig, Braunschweig, Germany

Christoph Seidl Technische Universität Braunschweig, Braunschweig, Germany

Dimitrios Serpanos Industrial Systems Institute, ATHENA Research Center, Patras, Greece

Nicholas Stifter Christian Doppler Laboratory for Security and Quality Improvement in the Production System Lifecycle (CDL-SQI), Institute of Information Systems Engineering, Technische Universität Wien, Vienna, AustriaSBA Research, Vienna, Austria

Christos Tselios Citrix Systems Inc, Patras, Greece

Jan Vollmar Siemens AG, Erlangen, Germany

Laura Waltersdorfer Christian Doppler Laboratory for Security and Quality Improvement in the Production System Lifecycle (CDL-SQI), Institute of Information Systems Engineering, Technische Universität Wien, Vienna, Austria

Edgar Weippl Christian Doppler Laboratory for Security and Quality Improvement in the Production System Lifecycle (CDL-SQI), Institute of Information Systems Engineering, Technische Universität Wien, Vienna, AustriaSBA Research, Vienna, Austria

Ralf Westphal Siemens Mobility GmbH, Braunschweig, Germany

Manuel Wimmer Christian Doppler Laboratory for Model-Integrated Smart Production (CDL-MINT), WIN-SE, JKU Linz, Linz, Austria

Dietmar Winkler Christian Doppler Laboratory for Security and Quality Improvement in the Production System Lifecycle (CDL-SQI), Institute of Information Systems Engineering, Technische Universität Wien, Vienna, Austria

Sabine Wolny Christian Doppler Laboratory for Model-Integrated Smart Production (CDL-MINT), WIN-SE, JKU Linz, Linz, Austria

Ilsun You Soonchunhyang University, Asan-si, South Korea

Chapter 1
Introduction to Security and Quality Improvement in Complex Cyber-Physical Systems Engineering

Stefan Biffl, Matthias Eckhart, Arndt Lüder, and Edgar Weippl

Abstract Providing *Complex Cyber-Physical Systems* (C-CPSs) more efficiently and faster is a goal that requires improvements in engineering process for producing high-quality, advanced engineering artifacts. Furthermore, information security must be a top priority when engineering C-CPSs as the engineering artifacts represent assets of high value.

This chapter overviews the engineering process of C-CPSs, typically long-running technical systems, such as industrial manufacturing systems and continuous processing systems. This chapter also covers major areas of requirements that include: (a) processes with intensive generation of engineering artifacts; (b) challenges regarding dependencies and complexity of engineering artifacts, stemming from variants of a product and the associated production process for a family of products; (c) management of model and consistency rules for dependencies between model parts; (d) the internationalization of the engineering process with partners on different levels of trust; and (e) the security of the engineering processes, such as confidentiality of engineering plans, and the security of the systems to be engineered, such as security aspects in the design phase.

For selected requirement areas, the chapter discusses several approaches for quality improvement from business informatics that addresses important classes of requirements, but introduces new complexity to the engineering process. Therefore, the chapter reviews information security improvement approaches for engineering

S. Biffl (✉)
Institute of Information Systems Engineering, Technische Universität Wien, Vienna, Austria
e-mail: stefan.biffl@tuwien.ac.at

M. Eckhart · E. Weippl
Christian Doppler Laboratory for Security and Quality Improvement in the Production System Lifecycle (CDL-SQI), Institute of Information Systems Engineering, Technische Universität Wien, Vienna, Austria

SBA Research, Vienna, Austria
e-mail: matthias.eckhart@tuwien.ac.at; edgar.weippl@tuwien.ac.at

A. Lüder
Otto-v.-Guericke University/IAF, Magdeburg, Germany
e-mail: arndt.lueder@ovgu.de

S. Biffl et al. (eds.), *Security and Quality in Cyber-Physical Systems Engineering*,
https://doi.org/10.1007/978-3-030-25312-7_1

processes, including the consideration of new security requirements stemming from risks introduced by advanced informatics solutions. Finally, the chapter provides an overview on the book parts and the contributions of the chapters to address advanced engineering process requirements.

Keywords Complex cyber-physical systems · Engineering process · Multidisciplinary engineering · AutomationML · Information security

1.1 Motivation

The engineering of *Complex Cyber-Physical Systems* (C-CPSs), typically long-running and software-intensive technical systems, such as critical infrastructures or industrial production systems and their associated products, is a multidisciplinary, model-driven, and data-driven engineering process that often involves conflicting economic, quality, and security interests, risks, and issues. Quality is a key concern in the engineering process to enable engineers to provide technical systems effectively and efficiently, with a focus on sufficient system quality, particularly on safety and on value to customers. Information security has become increasingly important with growing networking capabilities of technical systems and the rise of the Internet, as the engineering environments and the resulting technical systems are part of a network that allows new kinds of attacks on data and systems. Past cyber-attacks against safety-critical systems, for example, a sewage treatment plant (Slay and Miller 2008) or a steel mill (Lee et al. 2014), demonstrated the devastating consequences that could result from inadequate security measures. Besides potential physical damages, cyber-attacks may also cause silent losses because of intellectual property theft. While practitioners agree that addressing security concerns is crucial for establishing a foundation for system safety and quality, many companies hesitate to introduce sufficient security mechanisms and processes in their environments as well as in their products and engineering processes as they lack methods and information for risk assessment and often cannot relate the benefits to the associated extra cost and reduced usability.

Scope of the Book Providing software-intensive technical systems more efficiently and faster requires improvements of engineering process quality and, often, global cooperation in distributed engineering that requires improvements in security considerations for engineering processes. As a novel research approach, we consider the combination of quality improvement and information security for analyzing and improving engineering processes. Quality improvement contributions tend to make engineering processes faster and more efficient by reducing avoidable rework. On the other hand, even if stakeholders deem security fundamental, they may have difficulties in arguing the considerable extra cost of resources in engineering processes. Therefore, a balance of quality improvement and security would be desirable, a balance that overall reduces the resources required for engineering processes and introduces an adequate level of security, which is necessary for sustainable engineering in a globally distributed environment.

Contributions to Scientific Communities *Automation Systems Engineering*. One family of *Software-Intensive Technical Systems* are *Complex Cyber-Physical Systems* (*C-CPSs*). Following the *Encyclopedia of Business Informatics*,[1] a *Cyber-Physical System* (CPS) is defined as a system of communicating components that have both physical and data processing parts. They usually establish a hierarchy of technical system components that are controlled in a closed-loop structure.

This closed loop is established by measuring the state of the technical system using appropriate sensors, deciding on necessary control actions within an information processing system based on measured state and behavior specification, and executing these control actions by actors (VDI 2206 2004; Lunze 2016). Thus, information processing is essential for the behavior quality of the technical system.

Cyber-Physical Systems range from very simple embedded systems, such as drives or rotary encoders, up to very large systems, such as production systems, power plants, energy transmission systems, air planes, and train systems. In this book, such *Complex Cyber-Physical Systems* (*C-CPSs*) are considered, characterized by the nature of such systems and the fact that

- They require significant effort, material, and financial means.
- Their life cycle is measured in decades.
- Their system behavior is complex.
- Their economic and social impact is significant.

Within the engineering processes of a *C-CPS*, information about the overall structure and the mechanical, electrical, etc., construction of the applicable sensors and actors is required in conjunction with the specification of the intended behavior as input. This information must have sufficiently high quality to finally ensure system safety and economic efficiency (Schnieder 1999). In addition, the engineering results of information processing are required within virtual commissioning and commissioning of the *C-CPS* (Strahilov and Hämmerle 2017). Thus, it is obvious that engineering processes of *C-CPS* require information exchange between engineering disciplines, involved engineers of possibly different legal entities, and involved engineering tools.

Contributions in this book consider requirements, risks, and solutions to guarantee the security and quality of *C-CPS*. Involved engineers and project managers will be enabled to identify possible quality and security challenges they have to cope with. In addition, possible measures are described assisting involved staff to handle the identified challenges.

C-CPS Software and System Quality Analysis and Improvement. Typical assumptions for research in quality assurance and improvement for small software-intensive systems that are safe and easy to reset as well as for business software systems that use standard operating systems and hardware and do not rely on specific real-time hardware may not hold for long-running technical systems, such as critical

[1] http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/informationssysteme/Sektorspezifische-Anwendungssysteme/cyber-physische-systeme

infrastructure or industrial production systems. Therefore, researchers in quality assurance and improvement can benefit from better understanding challenges on quality assurance and quality improvement coming from the engineering of Complex Cyber-Physical Systems based on the use cases and requirements presented in Part I. The use cases and methods for software and system quality analysis and improvement discussed in Part II, such as engineering process analysis, model-based systems engineering, or test automation, provide researchers with insights from in-depth examples that can be adapted to a range of similar but different applications. Finally, the security threats and countermeasures discussed in Part III provide nonexperts in information security with insights into issues to consider when designing quality improvements in engineering contexts. Therefore, researchers and practitioners can take away requirements and building blocks for future methods and tools from the discussion of quality improvement approaches to address selected challenges from automation systems engineering.

Information Security. Researchers in the information security area gain a comprehensive understanding of the security challenges involved in engineering Complex Cyber-Physical Systems. Furthermore, the proposed concepts for securing the engineering process will allow them to evaluate other approaches in order to determine whether they can be applied to overcome these challenges. Since this book also covers security aspects that go beyond protecting the engineering process, researchers gain insights into how the security of Complex Cyber-Physical Systems can be enhanced by integrating security into the engineering phase. Finally, the discussed open challenges may motivate scholars to develop and pursue new research directions.

The remainder of the chapter is structured as follows: Sect. 1.2 introduces Engineering Processes for Software-Intensive Systems and requirements for quality and security improvement. Section 1.3 discusses business informatics approaches for quality improvement that address classes of requirements from engineering software-intensive systems, but may introduce unwanted IT security risks and requirements. Section 1.4 analyzes potential security issues that may be encountered when engineering software-intensive systems. Based on this analysis, the section defines security requirements that must be met; otherwise, these issues may result in compromised engineering processes, which eventually also affect the security of the systems to be developed. Section 1.5 provides an overview on the book parts and the contributions of the chapters to address advanced engineering process requirements. Section 1.6 suggests relevant contributions in this book for selected reader groups.

1.2 Engineering Processes for Software-Intensive Systems

This section introduces an engineering process view on the engineering of long-running software-intensive technical systems, such as industrial manufacturing systems and continuous processing systems, and derives major areas of requirements.

1.2.1 Background

Our daily life is characterized by technical systems that make our life easier, more comfortable, and safer compared with the lives of our ancestors. These systems create, distribute, maintain, and dispose goods, energy, and information relevant for our lives. Some of these systems are designed for a very long life span, for example, nuclear power plants or production systems in the process industry running several decades. Some of these systems have a medium long life span, for example, production systems, wind mills, or ships running about one decade. And some of them have only a very short lifespan, like rockets.

All of these technical systems (fulfill at least three of the four characteristics of a *C-CPS*) have in common the need to be designed, established, and controlled. Therefore, appropriate information processing systems are required to establish *C-CPS*. Especially, the control of the behavior of these systems is a critical problem requiring high-quality, safe, and secure software systems. Therefore, such systems are considered *cyber-physical systems* combining physical parts (establishing the mechanical, electrical, electronical, etc., construction) and cyber parts (establishing the information processing for control of behavior) (Zanero 2017; Monostori 2014; Lee 2008).

The engineering of such *cyber-physical systems* comprises usually the phases requirement collection, architecture design, implementation, test, deployment, and operation. In all phases, the duality of hardware and software needs to be considered (Gruhn et al. 2017). Considering requirement collection, architecture design, implementation, and operation of these systems is of dual nature. On the one hand, the proper system behavior has to be achieved regarding the reason why the system is designed. In case of a nuclear power plant or a wind mill, proper system behavior is the correct current and voltage intended over time, in case of a ship or a rocket proper system behavior is proper transportation conditions enabling proper transport services; and in case of a production system proper system behavior is the proper product creation. In all cases, the intended output of the system drives the correct behavior.

One special type of long-living *C-CPS* are production systems. These systems possess a duality of products to be produced by executing production processes and the production system executing the production processes. Both system kinds need to be engineered and depend on each other. Here, the so-called PPR concept, concerning product design, production process design, and production resource design, provides the background (Biffl et al. 2017a). Their life cycle and their engineering are detailed in Biffl et al. (2017b). Figure 1.1 illustrates the engineering process steps and selected domain expert roles for engineering a *C-CPS*.

Within such a *C-CPS*, the product definition as the first step of the engineering process provides both the behavior specification as well as technical, economic, environmental, legal, etc., bordering conditions. As an example, a rolling mill shall be considered. Starting point of the rolling mill engineering is the product definition of the rolled steel. Thereby, the production process of the intended steel coils is defined and steel properties, such as steel type and mass, are given. This specification

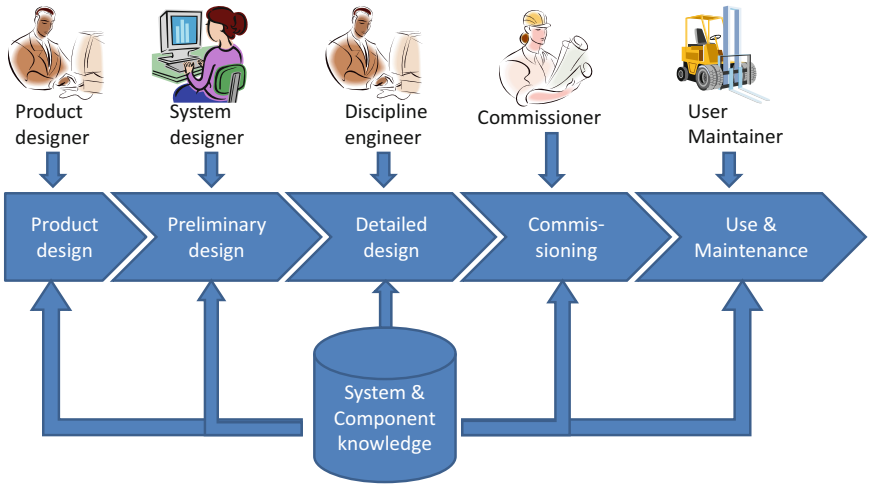

Fig. 1.1 Domain expert roles in the engineering of *C-CPSs*, typically long-running industrial production systems

is from the information point of view similar to the definition of the intended current flow in a power plant or the transport behavior of a rocket.

Usually, the product definition is provided by a product designer as first stakeholder of the engineering process. As indicated by Foehr et al. (2013), the product definition is accompanied by a description of the intended product quality, which is a requirement for the production process execution and, thereby, for the technical and behavioral properties of the production system. For a rolling mill, the product definition is given by the intended material thickness and material properties resulting in a set of necessary milling steps with milling pressure and cooling properties.

The product definition is succeeded by the preliminary design of the production system. Within this second phase, the production system designer as second involved stakeholder develops the overall system structure by assigning manufacturing resources to the different required production process steps of the product. In a steel mill, this overall system structure defines for example the number of mating roles, the length of the transport system (and thereby the number of transport roles), and the number of cooling units.

Exploiting the developed overall system, design groups of discipline-specific engineers covering mechanical, electrical, automation, etc. engineering act as the next stakeholders in the third production system engineering (PSE) process phase. They detail the production system design in the different engineering disciplines in a parallel and round-trip-driven way leading to a system specification in such detail that it can be physically established. For a rolling mill for example, the mechanical engineering defines type and location of drives and roles of the different required types, the electrical engineering defines their wiring related to electrical power

transmission and control signal exchange, and the automation system engineering implements the necessary control system parts required to drive the roles dependent on the intended steel quality.

The next phase in the production system engineering process contains its physical realization based on the developed engineering artifacts by several involved installation and ramp-up specialists as last set of stakeholders. As indicated by Lüder et al. (2017a, b), the involved stakeholders exchange engineering information along the complete life cycle of the production system. Therefore, they require standardized information exchange technologies as indicated by Lüder et al. (2017c). The intended increasing digitalization of a wide range of technical systems following approaches like *Industry 4.0* or *Industrial Internet of Things* (IIoT) establishes additional intentions to the engineering. The different components of technical systems shall be accompanied by their digital representation realized as the *management shell* of the *Industry 4.0 component* or the *digital shadow* in the IoT world. Using this digital representation, necessary activities like system maintenance and system optimization can be simplified and improved, leading to safer, more environmental-protective, and more economic systems. Thus, also stakeholders like users, maintainers, etc., come into play for the high-quality and secure engineering process.

Another main development direction in the engineering of production systems is the increasing use of system knowledge. This covers for example the reuse of existing artifacts, the system-wide standardization of components, the application of system family architectures, etc. This trend is mainly addressed by engineering organization improvement approaches as for example in VDI Guideline 3695 (2009). Comparing the engineering and use of other types of *C-CPS* with the engineering and use of production systems, it is easy to see that they usually involve similar life cycle phases and sets of stakeholders (Lindemann 2007).

1.2.2 Research Questions

The intended quality and security improvement of engineering processes of *Complex Cyber-Physical Systems* (*C-CPS*) initially requires an improvement of the understanding of these processes to enable the identification of possible impacts of quality and security improvement approaches that have been developed in information sciences. Thus, the initial research question considered in this book is related to a detailed knowledge about structure, behavior, and further characteristics of engineering processes for *C-CPS*.

RQ1a: What Are Typical Characteristics of Engineering Processes for Long-Running Software-Intensive Technical Systems? As indicated earlier, an engineering process of *C-CPS* establishes a network of engineering activities connected by information exchange. The different involved engineering disciplines step-by-step enrich the overall model system of the intended technical system. Each discipline works with its own models, while all related models need to be kept consistent, that is, there