SpringerBriefs in Computer Science

SpringerBriefs present concise summaries of cutting-edge research and practical applications across a wide spectrum of fields. Featuring compact volumes of 50 to 125 pages, the series covers a range of content from professional to academic.

Typical topics might include:

- A timely report of state-of-the art analytical techniques
- A bridge between new research results, as published in journal articles, and a contextual literature review
- A snapshot of a hot or emerging topic
- An in-depth case study or clinical example
- A presentation of core concepts that students must understand in order to make independent contributions

Briefs allow authors to present their ideas and readers to absorb them with minimal time investment. Briefs will be published as part of Springer's eBook collection, with millions of users worldwide. In addition, Briefs will be available for individual print and electronic purchase. Briefs are characterized by fast, global electronic dissemination, standard publishing contracts, easy-to-use manuscript preparation and formatting guidelines, and expedited production schedules. We aim for publication 8–12 weeks after acceptance. Both solicited and unsolicited manuscripts are considered for publication in this series.

**Indexing: This series is indexed in Scopus, Ei-Compendex, and zbMATH **

Wanja Zaeske • Umut Durak

DevOps for Airborne Software

Exploring Modern Approaches

Springer

Wanja Zaeske
Institute of Flight Systems
German Aerospace Center
Braunschweig, Germany

Umut Durak
Institute of Flight Systems
German Aerospace Center
Braunschweig, Germany

ISSN 2191-5768 ISSN 2191-5776 (electronic)
SpringerBriefs in Computer Science
ISBN 978-3-030-97578-4 ISBN 978-3-030-97579-1 (eBook)
https://doi.org/10.1007/978-3-030-97579-1

This book is dedicated to the open-source communities. Not only did their collaborative strive for progress in software engineering allow us to compile our vision into a demonstrator—their very existence is what sparked most of our ideas in the first place! The elegance of their countless innovations deserves acknowledgement.

This book both attributes said communities for their achievements and promotes their innovation in the aviation community.

Preface

It was the 2019 SciTech Forum; we were discussing at the Software Technical Committee of the American Institute of Aeronautics and Astronautics (AIAA) with esteemed colleagues from the major industry players and influential academic and research organizations how aerospace software engineering is lacking in keeping up with the game-changing Agile practices. Legacy processes were providing low risk, on the one hand, and proven design assurance practices, on the other hand, hindering the potential for streamlined and Agile software development. How could modern approaches—such as DevOps—be adapted so that they can provide agility while supporting the regulations, such as DO-178C? These discussions led to a well-attended and well-received panel session about DevOps at the 2021 SciTech Forum.

It was then hard to find any publications about DevOps for airborne systems. There was also no real tool, technology, or solution provider for this domain. That motivated us to start research on DevOps for airborne software to explore modern Agile practices using demonstrators, to identify the unique challenges of this highly regulated domain, and to develop approaches to tackle them. Early results were published in 2020 at the Dependable DevOps Workshop within the International Conference on Computer Safety, Reliability and Security (SAFECOMP) and in 2021 at the AIAA SciTech Forum. We shared them at the abovementioned panel session. This SpringerBrief now presents the round story that reports on our exploration in using modern approaches to implement DevOps practices for avionic software. It tries to render all the steps of the DevOps cycle by promoting one or another tool or technology that may suit well for airborne systems. The highlights include Rust, the modern systems programming language, Behavior-Driven Development (BDD) using Rust, DevOps automation with Nix and Hydra, and virtualization with the embedded hypervisor XtratuM Next Generation.

We invite the reader to this first experience report about using DevOps for airborne software. We further encourage the reader to continue the development of this promising direction for advancing the current state of the art in aerospace software engineering.

Braunschweig, Germany
January 2022

Wanja Zaeske
Umut Durak

Contents

1 Introduction 1
- 1.1 Issues in Airborne Development 1
- 1.2 From Agile to DevOps 2
- 1.3 Constraints in Avionic Software Engineering 4
- 1.4 Structure 5
- References 5

2 Background 7
- 2.1 Certification in Avionics: DO-178 7
- 2.2 Version Control with Git 9
- 2.3 Rust, a Modern Systems Programming Language 11
- 2.4 Test-Driven Development 13
- 2.5 Automation in DevOps 14
  - 2.5.1 Continuous Integration 14
  - 2.5.2 Continuous Delivery 16
  - 2.5.3 Continuous Deployment 17
- 2.6 Behavior-Driven Development 17
- 2.7 Embedded Virtualization 19
  - 2.7.1 Virtualization in Avionics 19
  - 2.7.2 XtratuM Next Generation 20
- 2.8 Nix and Hydra 21
- 2.9 RTLola 23
- References 25

3 Approach 27
- 3.1 Development 28
  - 3.1.1 Avoiding Errors 28
  - 3.1.2 Requirements from Plan to Verification 29
  - 3.1.3 Unifying Build System and Package Manager 30

3.2 Operation 31
3.2.1 Operating Product and Toolchain 31
3.2.2 Monitoring the Product 32
3.2.3 Closing the Feedback Loop 33
3.3 Summary 33
References 33

4 Demonstrator and Evaluation 35
4.1 TAWS and openTAWS 35
4.2 Enhancing Hypervisor Partitions with Rust 37
4.3 Streamlining the Requirements Engineering with BDD 41
4.4 Continuous Integration 44
4.4.1 GitHub Actions 44
4.4.2 Nix and Hydra 46
4.5 Monitoring with RTLola 47
References 49

5 Outlook and Conclusion 51
5.1 Outlook 51
5.1.1 Modify Setup for Full XNG Compatibility 51
5.1.2 Allow for Code Coverage Analysis 52
5.1.3 RTLola and Rust for Resilience 52
5.1.4 Online Monitoring for Software Planning 52
5.1.5 Operating Development Grade Products in Real Aircraft 53
5.1.6 Shortcomings of Nix 53
5.1.7 Fulfilling More DO-178 Objectives 53
5.2 Conclusion 54
References 55

Acronyms

ABI	Application Binary Interface
API	Application Programming Interface
BDD	Behavior-Driven Development
CD	Continuous Delivery/Deployment
CI	Continuous Integration
CPS	Cyber-Physical System
FPGA	Field-Programmable Gate Array
HTTP	Hypertext Transfer Protocol
IMA	Integrated Modular Avionics
IDE	Integrated Development Environment
IMA	Integrated Modular Avionics
IO	Input/Output
IT	Information Technologies
MC/DC	Modified Condition/Decision Coverage
MCU	Microcontroller Unit
MOPS	Minimum Operational Performance Standards
OS	Operating System
PRNG	Pseudorandom Number Generator
SKE	Separation Kernel Emulator
SSH	Secure Shell
TAWS	Terrain Awareness and Warning System
TDD	Test-Driven Development
UB	Undefined Behavior
UI	User Interface
URL	Uniform Resource Locator
VCS	Version Control System
VM	Virtual Machine
XCF	XNG Configuration File
XNG	XtratuM Next Generation