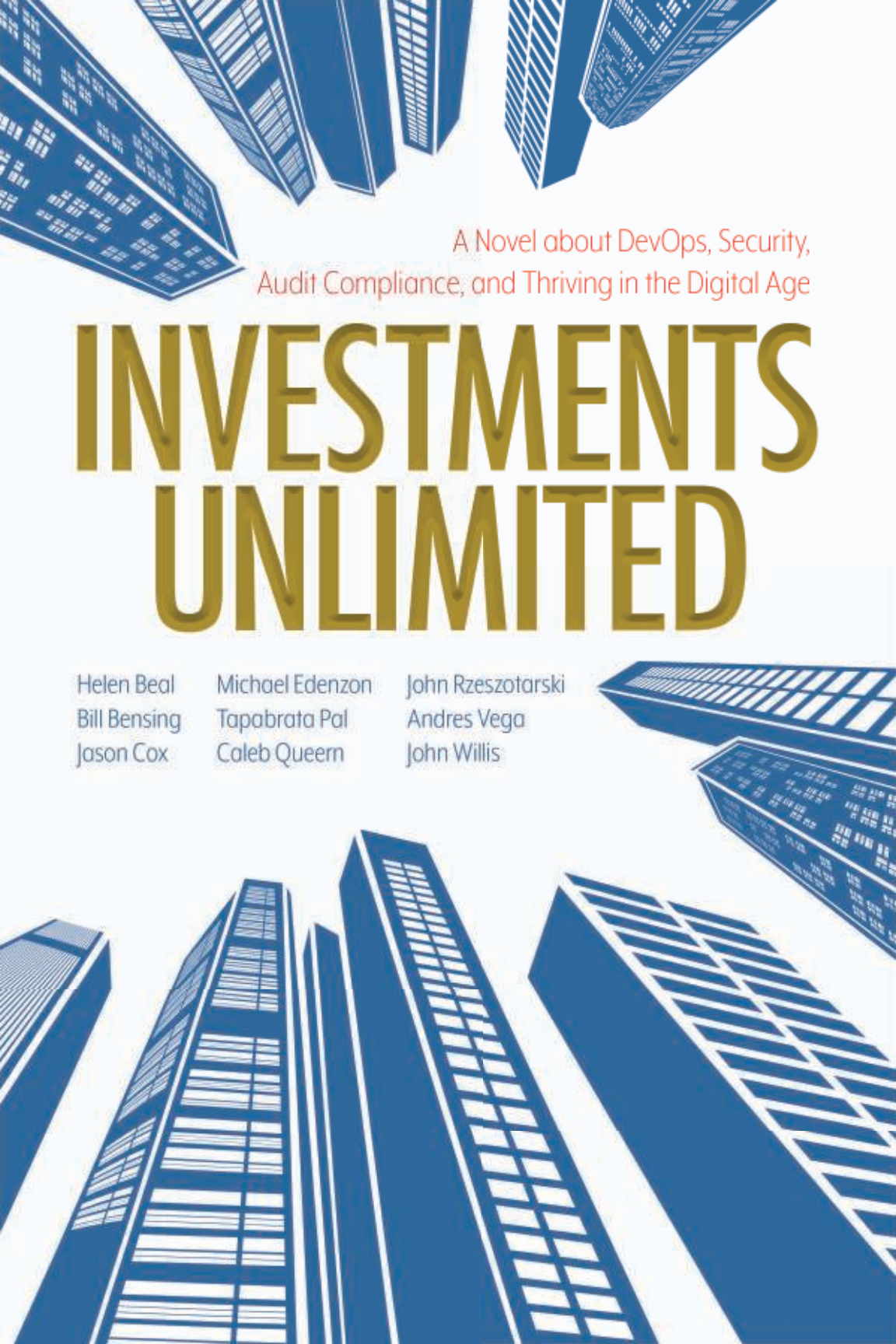
A Novel about DevOps, Security, Audit Compliance, and Thriving in the Digital Age

INVESTMENTS UNLIMITED

Helen Beal
Bill Bensing
Jason Cox
Michael Edenzon
Tapabrata Pal
Caleb Queern
John Rzeszotarski
Andres Vega
John Willis



A Novel about DevOps, Security, Audit Compliance, and Thriving in the Digital Age

INVESTMENTS UNLIMITED

Helen Beal
Bill Bensing
Jason Cox
Michael Edenzon
Tapabrata Pal
Caleb Queern
John Rzeszotarski
Andres Vega
John Willis

IT Revolution
Portland, Oregon

25 NW 23rd Pl, Suite 6314
Portland, OR 97210

First Edition
Printed in the United States of America

27 26 25 24 23 22 1 2 3 4 5 6 7 8 9 10

Cover and book design by Devon Smith

Library of Congress Control Number: 2022935846

ISBN: 9781950508532
eBook ISBN: 9781950508549
Web PDF ISBN: 9781950508563

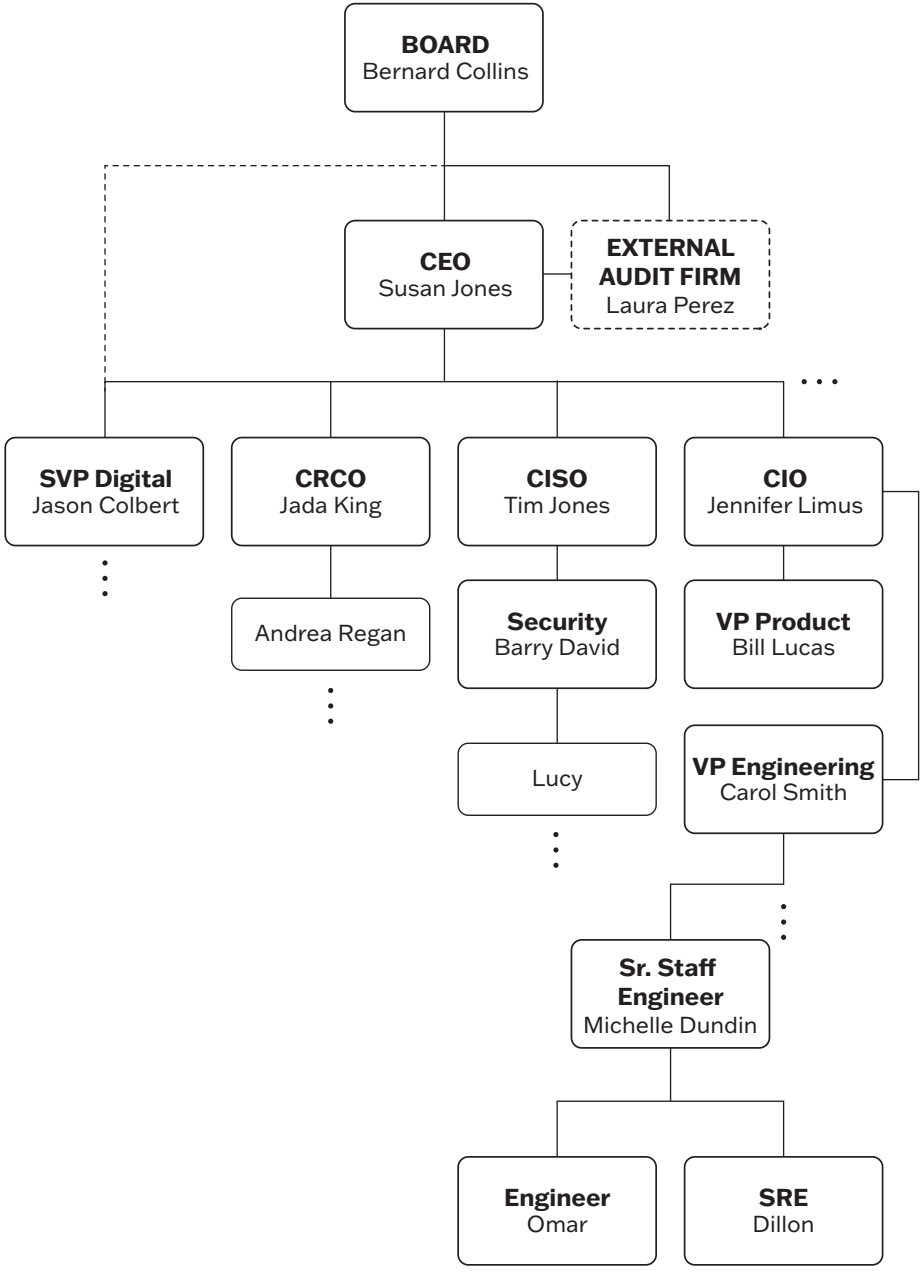
This is a work of fiction. Names, characters, and businesses are the products of the authors' imaginations. Any resemblance to actual persons, living or dead, or actual businesses is purely coincidental. However, certain real long-standing institutions, agencies, and public offices are mentioned. The events in this book are fictional but inspired by many real-life events.

For information about special discounts for bulk purchases or for information on booking authors for an event, please visit our website at www.ITRevolution.com.

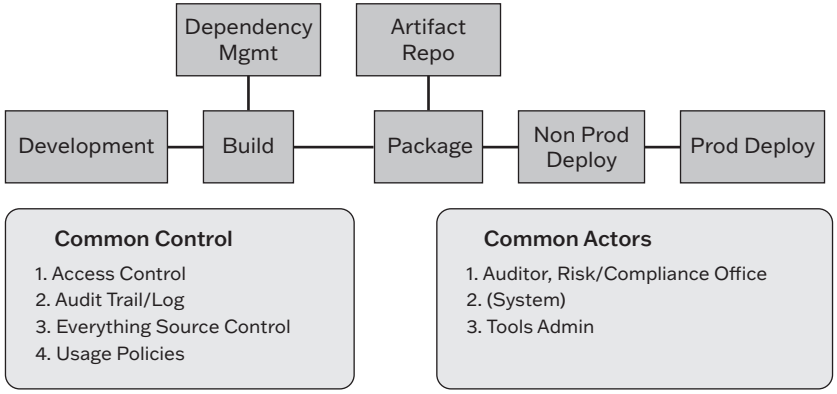
INVESTMENTS UNLIMITED

To all those change agents in every organization
who dare to challenge the status quo,
build bridges instead of walls,
and propel us into the unlimited future.

CONTENTS



Organization Chart



* Source: *DevOps Automated Governance Reference Architecture*

Figure 1

Action Items	Control Stage	Attestation	Source of Truth	Example
Peer Review	Build	Number of Approvers	Source Control Tool	Pass
Controls/Toll-gates	Deployment	Pass/Fail	Policy Engine	Pass
Elevated Access	Deployment	Pass/Fail	Policy Engine	Pass

Table 1

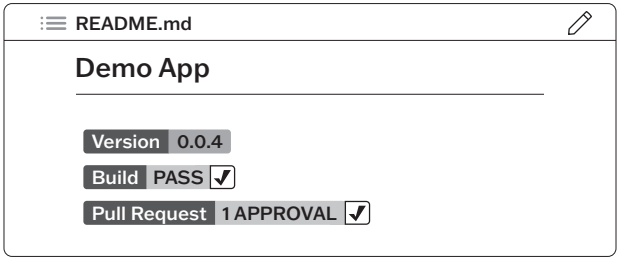

README.md
Demo App
Version 0.0.4
Build PASS
Pull Request 1 APPROVAL

Figure 2

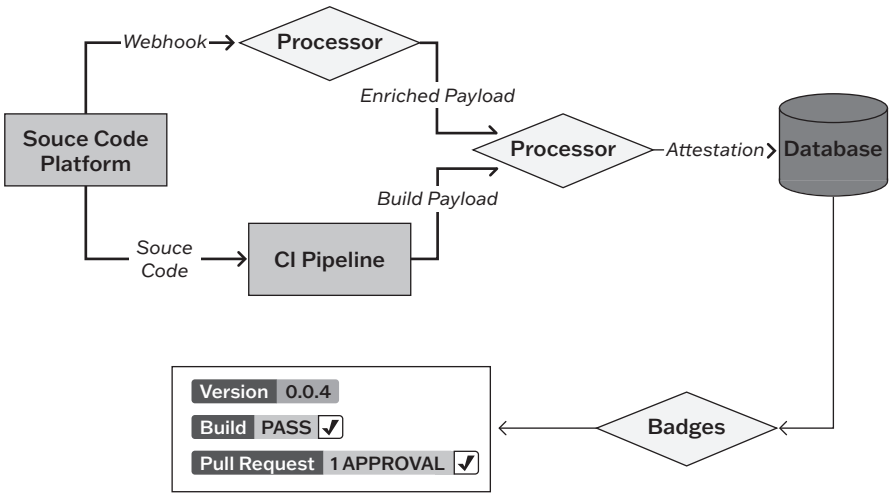

Webhook
Processor
Enriched Payload
Souce Code Platform
Processor
Attestation
Database
Build Payload
Souce Code
CI Pipeline
Version 0.0.4
Build PASS
Pull Request 1 APPROVAL
Badges

Figure 3

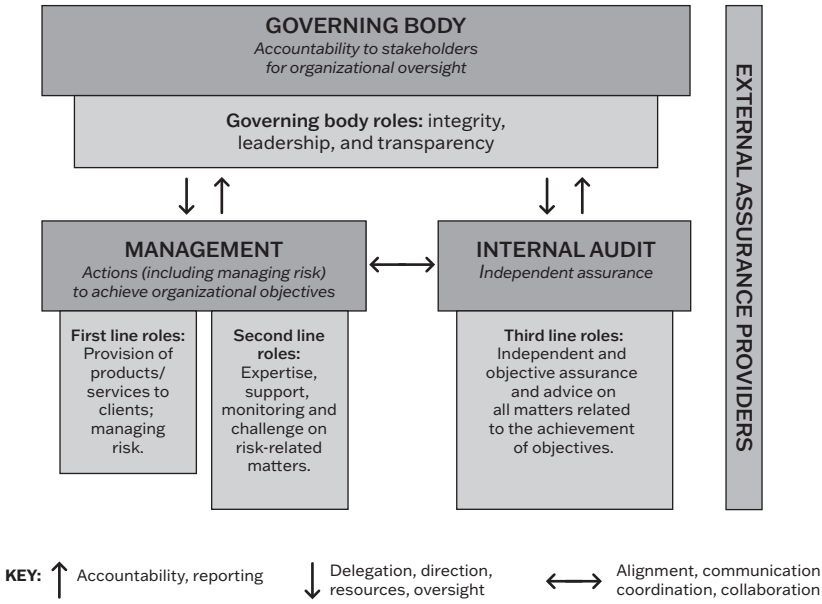

GOVERNING BODY
Accountability to stakeholders for organizational oversight
Governing body roles: integrity, leadership, and transparency
MANAGEMENT
Actions (including managing risk) to achieve organizational objectives
INTERNAL AUDIT
Independent assurance
First line roles: Provision of products/ services to clients; managing risk.
Second line roles: Expertise, support, monitoring and challenge on risk-related matters.
Third line roles: Independent and objective assurance and advice on all matters related to the achievement of objectives.
EXTERNAL ASSURANCE PROVIDERS
KEY:
Accountability, reporting
Delegation, direction, resources, oversight
Alignment, communication coordination, collaboration

Figure 4

Demo App

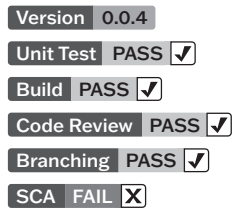


Figure 5

Demo App

Version 0.0.23
Build PASS ✓
Code Review 1 APPROVAL ✓
Branching Pattern TRUNK BASED ✓
Code Signature CHECKSUM VERIFIED ✓
Unit Tests PASS, 82% COVERAGE ✓
Static Code Quality Scan RELIABILITY: A, MAINTAINABILITY: A ✓
SCA PASS ✓
Static Security Test 0 VULNERABILITIES ✓

Figure 6

Demo App

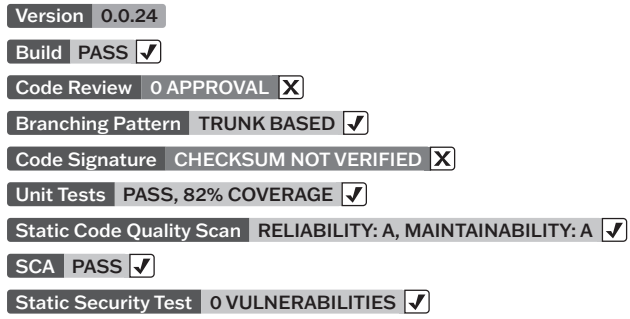


Figure 7

```
2021-05-04 11:01:01   CHECKING OUT HTTPS://GIT.INVESTMENTSUNLIMITEDBANK.COM;KRAKEN/IUI-DEMO/COMMIT/9A6E39
2021-05-04 11:01:02
2021-05-04 11:01:03   VERIFYING POLICY FOR IUI-DEMO-APP:0.0.24
2021-05-04 11:01:04
2021-05-04 11:01:05   IUI-DEMO-APP:0.0.24 FAILED THE FOLLOWING POLICIES ["CODE REVIEW", "CODE SIGNATURE"]
2021-05-04 11:01:06
2021-05-04 11:01:07   DEPLOYMENT FAILED, PLEASE CREATE A SUPPORT TICKET BEFORE OPENING AN INCIDENT
2021-05-04 11:01:08   HTTPS://HTTPS://DEVELOPER.INVESTMENTSUNLIMITEDBANK.COM//HELP
```

Figure 8

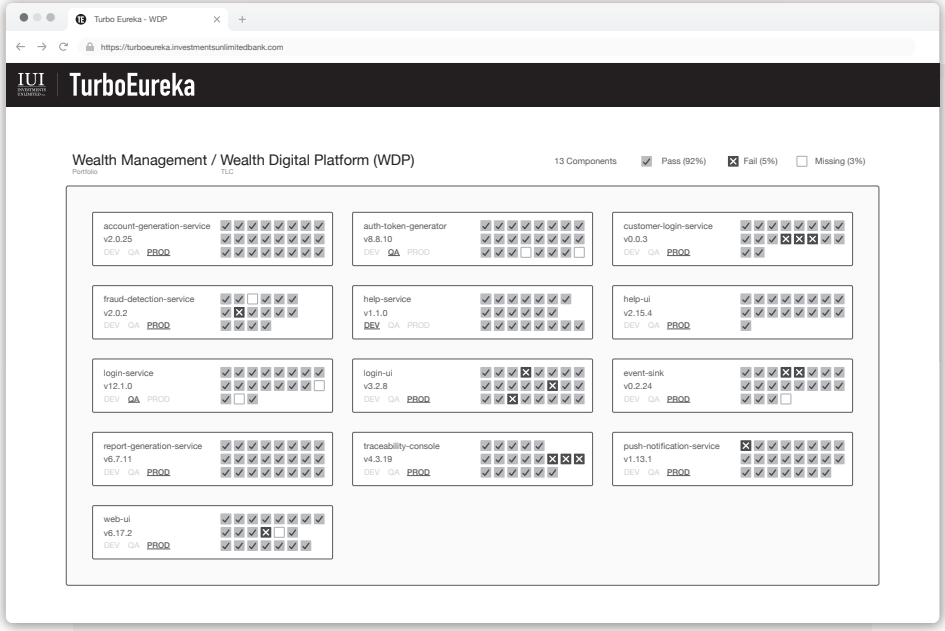

Turbo Eureka - WDP
https://turboeureka.investmentsunlimitedbank.com
IUI INVESTMENTS UNLIMITED
TurboEureka
Wealth Management / Wealth Digital Platform (WDP)
Portfolio
TLC
13 Components
Pass (92%)
Fail (5%)
Missing (3%)
account-generation-service
v2.0.25
DEV QA PROD
auth-token-generator
v8.8.10
DEV QA PROD
customer-login-service
v0.0.3
DEV QA PROD
fraud-detection-service
v2.0.2
DEV QA PROD
help-service
v1.1.0
DEV QA PROD
help-ui
v2.15.4
DEV QA PROD
login-service
v12.1.0
DEV QA PROD
login-ui
v3.2.8
DEV QA PROD
event-sink
v0.2.24
DEV QA PROD
report-generation-service
v6.7.11
DEV QA PROD
traceability-console
v4.3.19
DEV QA PROD
push-notification-service
v1.13.1
DEV QA PROD
web-ui
v6.17.2
DEV QA PROD

Figure 9

Control #	Control	Included in the IUI story?	Might you use this to block a build?
1	Peer Review	Yes	Yes; lack of peer review may block a build.
2	Static Application Security Testing	Yes	Yes; critical and (optionally) high findings may block a build.
3	Software Composition Analysis	Yes	Yes; critical and (optionally) high findings may block a build.
4	Code Quality	Yes	Probably not.
5	Unit Testing	Yes	Probably not.
6	Code Signing	Yes	Yes; lack of code signing may block a build.
7	License Check	No	Probably not.
8	Trusted Dependency Store	No	Yes; use of dependencies originating outside the trusted store may block a build.
9	Container Vulnerability Scan	No	Yes; critical and (optionally) high findings may block a build.
10	Secrets Scanning	No	Yes; the presence of sensitive tokens, keys, passwords, etc. in the source code may block a build.

Table 2

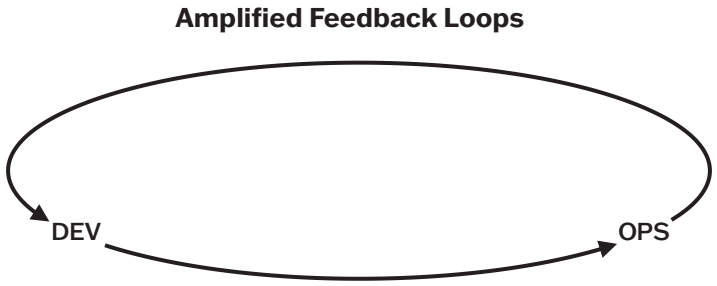


Figure 10

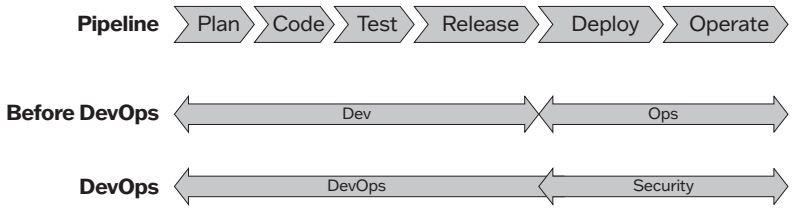

Pipeline
Plan
Code
Test
Release
Deploy
Operate
Before DevOps
Dev
Ops
DevOps
DevOps
Security

Figure 11

APPENDIX 1

MRAs AND MRIAs

MRAs are deficiencies that are important and should be addressed over a reasonable period of time, but where the institution's response need not be immediate. No matter how serious the concern, it is addressed to the institution's board of directors.

MRAs describe practices that:

1. Deviate from sound governance, internal control, or risk-management principles, and have the potential to adversely affect the bank's condition, including its financial performance or risk profile, if not addressed.
2. Result in substantive noncompliance with laws or regulations, enforcement actions, or conditions imposed in writing in connection with the approval of any application or other request by the bank.

The Office of the Comptroller of the Currency (OCC) refers to such practices as deficient practices. Such practices also may be unsafe or unsound generally, any action, or lack of action that is contrary to generally accepted standards of prudent operation and the possible consequences of which, if continued, would be abnormal risk or loss or damage to an institution, its shareholders, or the Deposit Insurance Fund.

MRIAs arise from an examination, inspection, or any other supervisory activity and are matters of significant importance and urgency that the Federal Reserve requires banking organizations to address immediately and include:

1. matters that have the potential to pose significant risk to the safety and soundness of the banking organization,
2. matters that represent significant noncompliance with applicable laws or regulations,
3. repeat criticisms that have escalated in importance due to insufficient attention or inaction by the banking organization,
4. and, in the case of consumer compliance examinations, matters that have the potential to cause significant consumer harm. An MRIA will remain an open issue until resolution and examiners confirm the banking organization's corrective actions.

5. For more, see the following references:
   https://www.federalreserve.gov/supervisionreg/srletters/sr1313a1.pdf
   https://www.federalregister.gov/documents/2017/08/09/2017-16735/proposed-guidance-on-supervisory-expectation-for-boards-of-directors
   https://www.occ.gov/publications-and-resources/publications/comptrollers-handbook/files/bank-supervision-process/pub-ch-bank-supervision-process.pdf (pg 51)

APPENDIX 2

PIPELINE DESIGN WITH CONTROL TOLLGATES

An organization might design their pipelines using a concept of "controls and tollgates." The idea is that certain types of gates in the pipeline can be used to alert or stop the software delivery process. Here are examples of sixteen such controls/tollgates:

- source code version control
- optimum branching strategy
- static analysis
- >80% code coverage
- vulnerability scan
- open-source scan
- artifact version control
- automated provisioning
- immutable servers
- integration testing
- performance testing
- build deploy testing automated for every commit
- automated rollback
- automated change order
- zero downtime release
- feature toggle

Controls in the Build Pipeline

Many events in a build pipeline can be collected and saved in a tamper proof format. Once available, they may:

- inform decisions to block a build
- trigger alerts monitored by a security operations team
- serve as attestation that controls were performed prior to deployment

As more controls from across the software development life cycle are implemented and their events are securely collected in a single place, the likelihood of risky software in production decreases, IT leaders and regulators will have improved visibility that inspires trust, and the organization enjoys safer software that allows it to accomplish its mission.

This table offers a few options readers may consider implementing in their own build pipelines. A list of more than thirty controls you may consider storing can be

found on page 34 of the IT Revolution white paper *DevOps Automated Governance Reference Architecture*.

Control #	Control	Included in the IUI story?	Might you use this to block a build?
1	Peer Review	Yes	Yes; lack of peer review may block a build.
2	Static Application Security Testing	Yes	Yes; critical and (optionally) high findings may block a build.
3	Software Composition Analysis	Yes	Yes; critical and (optionally) high findings may block a build.
4	Code Quality	Yes	Probably not.
5	Unit Testing	Yes	Probably not.
6	Code Signing	Yes	Yes; lack of code signing may block a build.
7	License Check	No	Probably not.
8	Trusted Dependency Store	No	Yes; use of dependencies originating outside the trusted store may block a build.
9	Container Vulnerability Scan	No	Yes; critical and (optionally) high findings may block a build.
10	Secrets Scanning	No	Yes; the presence of sensitive tokens, keys, passwords, etc. in the source code may block a build.

APPENDIX 3

DEVSECOPS MANIFESTO

Through security as code, we have and will learn that there is simply a better way for security practitioners, like us, to operate and contribute value with less friction. We know we must adapt our ways quickly and foster innovation to ensure data security and privacy issues are not left behind because we were too slow to change.

By developing security as code, we will strive to create awesome products and services, provide insights directly to developers, and generally favor iteration over trying to always come up with the best answer before a deployment. We will operate like developers to make security and compliance available to be consumed as services. We will unlock and unblock new paths to help others see their ideas become a reality.

We won't simply rely on scanners and reports to make code better. We will attack products and services like an outsider to help you defend what you've created. We will learn the loopholes, look for weaknesses, and work with you to provide remediation actions instead of long lists of problems for you to solve on your own.

We will not wait for our organizations to fall victim to mistakes and attackers. We will not settle for finding what is already known; instead, we will look for anomalies yet to be detected. We will strive to be a better partner by valuing what you value:

- Leaning in over always saying "No"
- Data and security science over fear, uncertainty, and doubt
- Open contribution and collaboration over security-only requirements
- Consumable security services with APIs over mandated security controls and paperwork
- Business-driven security scores over rubber-stamp security
- Red and Blue Team exploit testing over relying on scans and theoretical vulnerabilities
- 24/7 proactive security monitoring over reacting after being informed of an incident
- Shared threat intelligence over keeping info to ourselves
- Compliance operations over clipboards and checklists

You can read the full DevSecOps manifesto here: https://www.devsecops.org/

APPENDIX 4

SHIFT LEFT

As with most things related to DevOps and DevSecOps, the term "shift left" can be traced all the way back to Toyota Production Systems and the use of the Jidoka and the Andon Cord. The main idea is that when delivering products, it's more cost effective to find defects earlier in the process. This leads to higher-quality output, as well. The first use of "shift left" in software delivery can be traced back to software testing in the software development life cycle (SDLC).

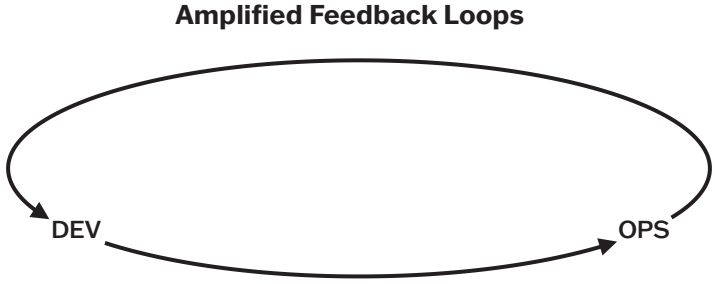
Shift-left testing is important because it helps to prevent the following types of harm due to late testing:

- Testers may be less involved in initial planning, often resulting in insufficient resources being allocated to testing.
- Defects in requirements, architecture, and design remain undiscovered while significant effort is wasted implementing them.
- Debugging (including identifying, localizing, fixing, and regression-testing defects) becomes harder as more software is produced and integrated.
- Encapsulation impedes white-box testing, reducing code coverage during testing.
- There is less time to fix defects found by testing, thereby increasing the likelihood that they will be postponed until later increments or versions of the system. This creates a "bow wave" of technical debt that can sink projects if it grows too large.[1]

The agile movement promoted Test-Driven Development (TDD) as a "shift-left" concept. It was the DevOps movement that really formalized the idea of "shift left" as a common term. Gene Kim et al. further explored this concept of "shift left" in

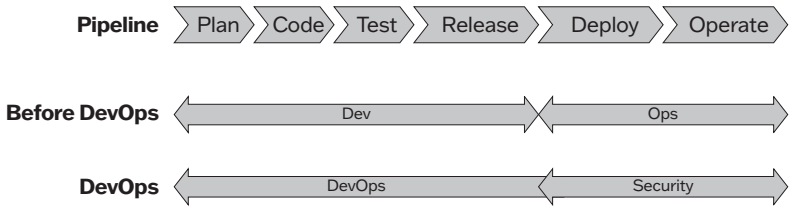
1. Wikipedia, "Shift-Left Testing," modified November 8, 2021. https://en.wikipedia.org/wiki/Shift-left_testing#:~:text=Defects%20in%20requirements%2C%20architecture%2C%20and,software%20is%20produced%20and%20integrated.

the book *The DevOps Handbook*. In it, they describes the Second Way as a process of amplifying feedback loops.



In 2014, Andrew Storms tied the concept of "shift left" to security in an article for DevOps.com called "Moving Security to the Left in a DevOps World."

A concise summary of "shifting left" may be the intentional prioritization of controls, behaviors, and capabilities in the SDLC that prevent defects in software in production rather than those which detect and respond to such defects.



APPENDIX 5

SOFTWARE COMPOSITION ANALYSIS

Software Composition Analysis (SCA) is the process of identifying the components that comprise a given piece of software. The components may be identified at a range of levels, from higher level (such as corresponding to items in "cloud diagrams") to mid level (such as distinct classes, modules, or files) to low level (such as functions or methods comprising a file or class).

The software to be examined may be generally perceived as a single monolithic entity (in which case SCA aims to reveal its constituent components), or it may—as in the case of modern operating systems—already be seen as a collection of components (in which case SCA may identify components at a greater level of granularity or identify the interrelationships among already-known components).

The term SCA may also refer to the analysis of a single component, showing for example its inputs, outputs, and side effects.

In the industry, SCA is often viewed as limited to identification of open source used within a software product. For example, a typical definition is: "SCA is the process of automating visibility into the use of open source software (OSS) for the purpose of risk management, security, and license compliance." However, SCA does not need to be limited to open-source and may include identification of proprietary third-party or in-house components. The common restriction to open source may be based on if software components are invisible without source code. However, software reverse-engineering, including both static examination (e.g., disassembly and decompilation) and dynamic examination (e.g., packet sniffing) of commercial products, provides often-feasible methods to determine the composition of software without the benefit of source code.

The purposes of SCA include security audits (particularly when a product's use of a particular version of a component can be identified and compared to repositories of known security vulnerabilities), license compliance (both OSS and proprietary components), and intellectual property infringement.

The SCA process should produce a valid software bill of materials (SBOM) or a "software tear down." The industry is aligning on the CycloneDX and SPDX standard formats for SBOMs.

US EXECUTIVE ORDER ON IMPROVING THE NATION'S CYBERSECURITY

On May 12, 2021, President Biden signed an executive order to improve the nation's cybersecurity and to protect federal government networks. This executive order was directly related to recent cybersecurity incidents such as SolarWinds, Microsoft Exchange, and the Colonial Pipeline.

"It is the policy of my administration that the prevention, detection, assessment, and remediation of cyber incidents is a top priority and essential to national and economic security. The federal government must lead by example. All federal information systems should meet or exceed the standards and requirements for cybersecurity set forth in and issued pursuant to this order."1

In short, the executive order calls for the:[2]

- Removal of barriers to threat information sharing between government and the private sector.
- Modernizing and implementing stronger cybersecurity standards in the federal government.
- Improving software supply chain security.
- Establishing a Cybersecurity Safety Review Board.
- Creating a standard playbook for responding to cyber incidents.
- Improving detection of cybersecurity incidents on federal government networks.
- Improving investigative and remediation capabilities.

1. https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/
2. https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/

APPENDIX 7

FAQ

This section is meant to provide general guidance on how to think about embarking on an automated governance effort and dispel misconceptions that may inhibit readers from getting started.

1. **The story focuses mostly on the technical approach to modernizing governance. Is there more to it than technology?** Yes. You may recall the classic triad of "People, Process, and Technology," a useful, if simple, way to think about the elements of such a transformation. For the purposes of quick storytelling, we focused on the tech; neglect the people and process aspects at your own risk.
2. **Do I need to be mostly a cloud native organization to pull all this off?** No, but we do see a correlation between maturity in cloud adoption and becoming a high-performing IT organization. Automated governance, like most modern development practices, benefits from the speed and agility offered by cloud computing capabilities.
3. **Does my organization need to have mostly in-house, non third-party developers (like the large tech companies) to pull all this off?** No. Anecdotally, however, there is evidence that such transformations are less challenging when there are less entities to account for, such as various outsourced software development partners who may not embrace the mission as energetically or possess as much agency as in-house developers. There are even examples of hesitance or even authority among third-party development teams to prioritize security fixes with the same enthusiasm as new features. This is likely an understandable, if unfortunate, consequence of contracts signed with third-party development teams being focused on new features being delivered by an agreed upon date; security and compliance efforts like automated governance may thus take a back seat.
4. **If my application portfolio is predominantly made up of certain programming languages or frameworks, will that make any of this easier**? The specific languages or frameworks in use in an organization are less important than the number of languages or frameworks that the organization decides

to officially support. Being intentional about such support and limiting the number of languages or frameworks reduces entropy in the organization and can generally make it easier to get work done together, like automated governance.

5. **Does my organization need to already be a moderate- to high-performing DevOps organization before considering automated governance?** No. There are few truly high-performing DevOps organizations. There are, however, parts of many organizations that are high performing when it comes to DevOps. While maturity in certain aspects of DevOps such as expressing "everything as code" and holistically applying software development best practices across development and operations, DevOps is not strictly a precursor to automated governance. In fact, automated governance can catalyze and accelerate progress along the continuum of the DevOps journey.
6. **Do I need to have already consolidated/homogenized the disparate build pipelines across my organization to pull all this off?** W. Edwards Deming professed that "uncontrolled variation is the enemy of quality". In that spirit, there are clear benefits to standardizing on the infrastructure and workflow that development of software must adhere to. Driving consolidation of the number of build pipelines and their configuration in an organization makes managing software delivery easier because when new practices such as automated governance are to be implemented, there are less environments that may drift and require attention. If your organization has not already taken active steps to do so, consider a formal effort to minimize the "build pipeline sprawl" so automated governance and other optimizations are effective across your software portfolio.
7. **Is a crisis needed for an organization to consider pursuing automated governance?** While crises like receiving an MRIA can focus an organization's attention and will rally action, we do not recommend waiting for a crisis. Instead, we encourage being proactive. You can start small with a few applications in an automated governance effort to demonstrate the value to leadership and earn permission for a broader effort—one that should reduce the likelihood of crises.

ACKNOWLEDGMENTS

When we first assembled to pull together a guidance document about governance, we struggled to get a compelling outline in place. We had several great ideas we wanted to convey, but no matter how we structured it, it was going to be very academic and profoundly dry.

Then we had an idea . . . why not turn it into a short story? That's exactly what we did. Susan, Tim, Bill, Jada, Michelle, Jason, and the rest of the cast sprung to life in a brief narrative to convey what we were trying to capture. The technical guidance paper suddenly became approachable. We were happy with the result, and it seemed the DevOps Enterprise Forum community agreed.

Four months later we got a call . . . "Gene and the staff at IT Revolution have discussed your paper. We want to turn it into a book!" Leah Brown told us. We were stunned and delighted. Leah went on to explain that IT Rev would do some editing and expansion of the narrative so that it would be a short novel. We all agreed that was a great idea. The more we thought about it, the more excited and enthusiastic we all became.

Finally, John Willis said he had another idea. He asked us if we were all willing to invest some additional time and turn this good idea into a great idea and into a full-length novel. We agreed and invited Helen Beal to join our squad of authors.

This book would not have happened without the incredible encouragement of the larger DevOps community. We are indebted to the inspiration and support of the Scenius[1] and the community of leaders from DevOps Enterprise Summit and DevOps Enterprise Forum. Gene and Margueritte Kim are at the top of that list as both the organizers and inspirational leaders of the DevOps movement.

1 Scenius: Breakthroughs typically emerge from a scene: an exceptionally productive community of practice that develops novel epistemic norms. Brian Eno, who first coined this, wrote, "major innovation may indeed take a genius—but the genius is created in part by a scenius." https://itrevolution.com/love-letter-to-conferences/#why-i-think-virtual-forum-worked-so-well

The core concepts presented throughout this book had been brewing through the years in the community, as well as in the form of several DevOps Enterprise Forum guidance papers produced at the annual gathering of community leaders and experts, including many of the authors of this book.[2] Without these papers and their collaborators' vision and research, this book would not exist. We want to mention those foundational papers and their collaborators for their invaluable contribution to the DevOps community:

- *An Unlikely Union: DevOps and Audit* (2015) by James DeLucia, Paul Duvall, Chairman, Mustafa Kapadia, Gene Kim, Dave Mangot, Tapabrata "Topo" Pal, James Wickett, Julie Yoo.
- *Dear Auditor* (2018) by Ben Grinnell, James Wickett, Jennifer Brady, the late Rob Stroud (may he rest in peace), Sam Guckenheimer, Scott Nasello, Tapabrata "Topo" Pal.
- *DevOps Automated Governance* (2019) by Michael Nygard, Tapabrata "Topo" Pal, Stephen Magill, Sam Guckenheimer, John Willis, John Rzeszotarski, Dwayne Holmes, Courtney Kissler, Dan Beauregard, Collette Tauscher.

How do you write a book with nine authors? Cat herding has been an important part of arriving at our destination. We would like to thank Leah Brown for her insightful and supportive sessions with the panel of authors, as well her shepherding of the collaborative editing process. This book would not have happened without her.

Subject matter experts were key to ensure our message stayed relevant and accurate. While this is a work of fiction, our intent was to convey the learning in prose that represented plausible real-world situations. We would like to thank Chris Palumbo for the regulatory insight and Branden Williams and Jen Suiters for their powerful lessons on MRIAs and what regulators would expect.

We would like to thank our peer reviewers, Gene Kim, Courtney Kissler, Emily Fox, Jeff Gallimore, Jennifer Hansen, Cameron Haight, and Maya Senen for their brilliant insight and critical and candid feedback that helped nudge the story from good toward great.

We would also like to thank Brian Scheck, Keith Silvestri, and Mike Onders, whose vision and dedication laid the foundation for many of the stories, learnings, and outcomes in this book.

---

2 The DevOps Enterprise Forum is an annual event held by IT Revolution, in which industry leaders and experts come together to discuss the most important challenges facing the community. From this event, a series of guidance papers are produced. You can view the full collection of papers here: https://itrevolution.com/resources.

Even with nine authors, the time investment was significant. We are indebted to our families and loved ones who gave us space to write, encouraged us despite the chatty late-night sessions with the group, and the supportive understanding when our noses were buried in our screens typing away at the narrative.

Bill Bensing would like to thank his mom, dad, family, and Kendra for their enduring support. The Nelsons', Tampows', Tingles', and Willis' of the world make these opportunities possible. They, and many others like them, give Bill the friendship, mentoring, and opportunities to be his best. These are the type of people Bill hopes everyone finds in their careers and lives.

Jason Cox would like to thank his wife Jane and their four children, Jonathan, Julia, Jessica, and Jenna. He would also like to thank his incredible team of SREs, fellow technology and business leaders who build magic every day and prove that we can all do the impossible.

Michael Edenzon would like to thank his family: Kathy, Marc, AJ, Zach, Irwin, and Frankie

Tapabrata Pal would like to thank his wife, Chiru, and their two children, Shaily and Ayush.

Caleb Queern would like to thank his wife Marian and his son Joseph.

John Rzeszotarski would like to thank his family: Marla, Sophia, Sebastian, Sawyer, and Simon.

Andres Vega would like to express his deepest gratitude to Olga, Victoria, and Mateo for giving him purpose and constantly challenging him to make Husband and Dad Unlimited thrive. They are the joy and happiness of his life.

And thank you! We are indebted to all of you, the larger community of business and technology leaders who are willing to listen, learn, experiment, and teach. We believe that the future is truly unlimited. With your help, we can all unlock new potential for our businesses, embrace better ways of working, and elevate our human experience across the planet. Thank you for joining us on this journey. Now, let's go change the world!

ABOUT THE AUTHORS

Helen Beal is a DevOps and Ways of Working coach, Chief Ambassador at DevOps Institute, and ambassador for the Continuous Delivery Foundation. She is the Chair of the Value Stream Management Consortium and provides strategic advisory services to DevOps industry leaders. She is also an analyst at Techstrong Research, hosts the Day-to-Day DevOps webinar series for BrightTalk and the Value Stream Evolution series on TechStrong TV. She currently lives in the UK.

Bill Bensing builds things that build things. He is a skilled leader and architect of software, people, teams, and companies. Bill is an expert at making innovation a wholly inclusive process. His love of DevOps comes from a background in logistics and operations management. Automated Governance is a topic Bill finds very interesting. He believes a lack of good governance is the single biggest issue preventing breakthrough value. Bill will tell you, "Good strategy and good governance are the grease and guide rails for success." He lives in the Tampa Bay, FL, area.

Jason Cox is a champion of DevOps practices, promoting new technologies and better ways of working. He enjoys helping organizations deliver more value, better, faster, safer and happier. He is an inspirational speaker who loves people and delights in amplifying their abilities with technology. Jason frequently speaks at conferences, contributes to open source and writes on technical and leadership topics. He currently leads several SRE teams and resides in Los Angeles with his wife and their children.

Michael Edenzon is a senior IT leader and engineer that modernizes and disrupts the technical landscape for highly regulated organizations. Michael provides technical design, decisioning, and solutioning across complex verticals and leverages continuous learning practices to drive organizational change. He is a fervent advocate

for the developer experience and believes that enablement-focused automation is the key to building compliant software at scale.

Topo Pal is a thought leader, keynote speaker, evangelist in the areas of DevSecOps, Continuous Delivery, Cloud Computing, Open Source Adoption and Digital Transformation. He is a hands-on developer and Open Source contributor. Topo has been leading and contributing to industry initiatives around automated governance in DevOps practices. Topo resides in Richmond, VA, with his wife and two children.

Caleb Queern helps CIOs and CISOs reduce risk across the software development life cycle so they can innovate quickly and win in the market. He lives in Austin, Texas with his wife, Marian, and son, Joseph.

John Rzeszotarski has led organizations with a focus on digital, payments, security, and development. His primary passion is solving complex business and IT problems through technology, fast flow, and building learning organizations. He loves coding new things and driving change in insanely regulated environments. He lives in Pittsburgh, PA, with his family.

Andres Vega helps engineering organizations securely build large-scale, distributed software leveraging novel approaches to reduce the compliance toil associated with the area. He is recognized in the open-source community as a maintainer, contributor, and technical leader focused on the improvement of ecosystem security. Outside of his profession, he is a family guy and an avid outdoors person. You are sure to find him adventuring with his family all over the trails of the San Francisco Bay Area in his best attempt not to get mauled to death by hungry mountain lions.

John Willis is an author and Senior Director of the Global Transformation Office at Red Hat. John is considered one of the founders of the DevOps movement. He lives in Acworth, GA.