

The DevOps Transformation Toolbox



RBI's Journey to Modern SW Engineering

Who is RBI?

- Leading corporate and investment bank in Austria
- Acting as RBI Group in 13 CEE markets represented as universal banks
- Around 46.000 employees servicing 19mn customers

Raiffeisen Bank International AG (RBI)



RBI regards Austria, where it is a leading corporate and investment bank, as well as Central and Eastern Europe (CEE) as its home market. 13 markets in the region are covered by subsidiary banks, the Group also comprises numerous other financial services providers, for instance in the field of leasing, asset management, factoring and M&A.

Overview

- 19 million customers
- 46,185 employees
- 1,771 business outlets
- since 1987 successful in CEE
- since 2005 listed at the Vienna Stock Exchange (ATX)

Business focus

RBI not only offers Austrian and international companies a broad range of products in corporate and investment banking, but also a comprehensive coverage in CEE. Through an extensive branch network, local companies of all sizes as well as private customers are supplied with high-quality financial products. RBI maintains representative offices and service branches in selected Asian and Western European locations to support its business activities.

Why do we see the need for transformation?

Less but more customer focused

Happier Employees

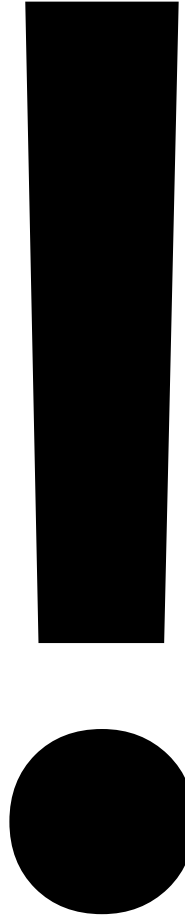


Faster Time To Market

Secure customer journeys

How do we get there?





1.

Have a clear picture of the
target operating model



1. Clear Picture



1. Clear Picture

Key objectives	Key Results
Deliver software in an automated and managed way	<ul style="list-style-type: none">• Software is delivered in an automated, secured and managed way to allow fast and flexible deployments without sacrificing quality. This includes a high level of automation and the use of a <u>proper CI/CD pipeline</u>• Reduce manual IT Operations and have applications run by <u>end2end DevOps teams</u> with high automation. Reduced manpower effort for running IT Products

2.

Objectives and Key Results (OKRs)



2. Objectives and Key Results (OKRs)

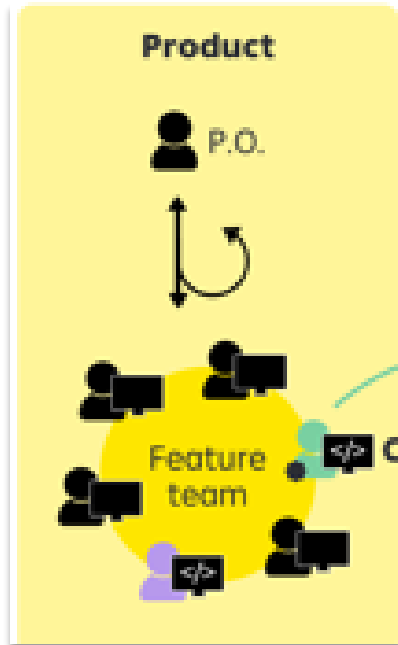
Key objectives	Key Results
Deliver software in an automated and managed way	<ul style="list-style-type: none">• Software is delivered in an automated, secured and managed way to allow fast and flexible deployments without sacrificing quality. This includes a high level of automation and the use of a <u>proper CI/CD pipeline</u>• Reduce manual IT Operations and have applications run by <u>end2end DevOps teams</u> with high automation. Reduced manpower effort for running IT Products

3.

Agile Engineering Maturity Model (AEMM)



3. Agile Engineering Maturity Model (AEMM)



Detailed result		CRAWL	WALK	RUN
CI/CD Automation	Code	100%	100%	100%
CI/CD Automation	Code quality	100%	100%	100%
CI/CD Automation	Automation	100%	100%	0%
CI/CD Automation	Pipeline	100%	100%	100%
CI/CD Automation	Deployment	100%	100%	100%
CI/CD Automation	Environment ownership	100%	100%	100%
CI/CD Automation	Environment setup	100%	100%	100%
DevOps Approach	Operations	100%	100%	100%
DevOps Approach	Monitoring	100%	100%	0%
DevOps Approach	Incident management	100%	100%	100%
DevOps Approach	Release cycle	100%	100%	100%
DevOps Approach	Release impact	100%	100%	100%
DevOps Approach	Quality	100%	100%	50%
DevOps Approach	Development process	100%	100%	100%
DevOps Approach	Team	100%	100%	67%
DevOps Approach	Skills	100%	33%	0%
Test Approach	Test Planning & Control	80%	60%	40%
Test Approach	Test Analysis & Design	100%	100%	0%
Test Approach	Test Implementation & Execution	100%	100%	75%
Test Approach	Test Data	100%	100%	50%
Test Approach	Test Environment	100%	100%	100%
Test Approach	Test Doubles	100%	100%	0%
Test Approach	Test Documentation	100%	100%	0%
Test Approach	Test Skills	100%	100%	100%
Test Automation	Test Design	100%	50%	0%
Test Automation	Test Automation Development	100%	100%	50%
Test Automation	Test Execution	100%	100%	50%
Security	Security Design	0%	0%	0%

4. Agile Engineering Coaches



- Focus on Engineering Practices: CI/CD, Test Automation
- Target Duration: 3-6 months
- Clear Agreement with Team about scope/duration/goals

Boundary Conditions:

- Coaches don't do the work for the team but provide hands-on support
- They enable the team to do the work by themselves

4.

Agile Engineering Coaches

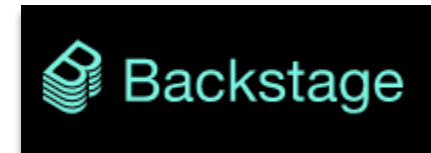
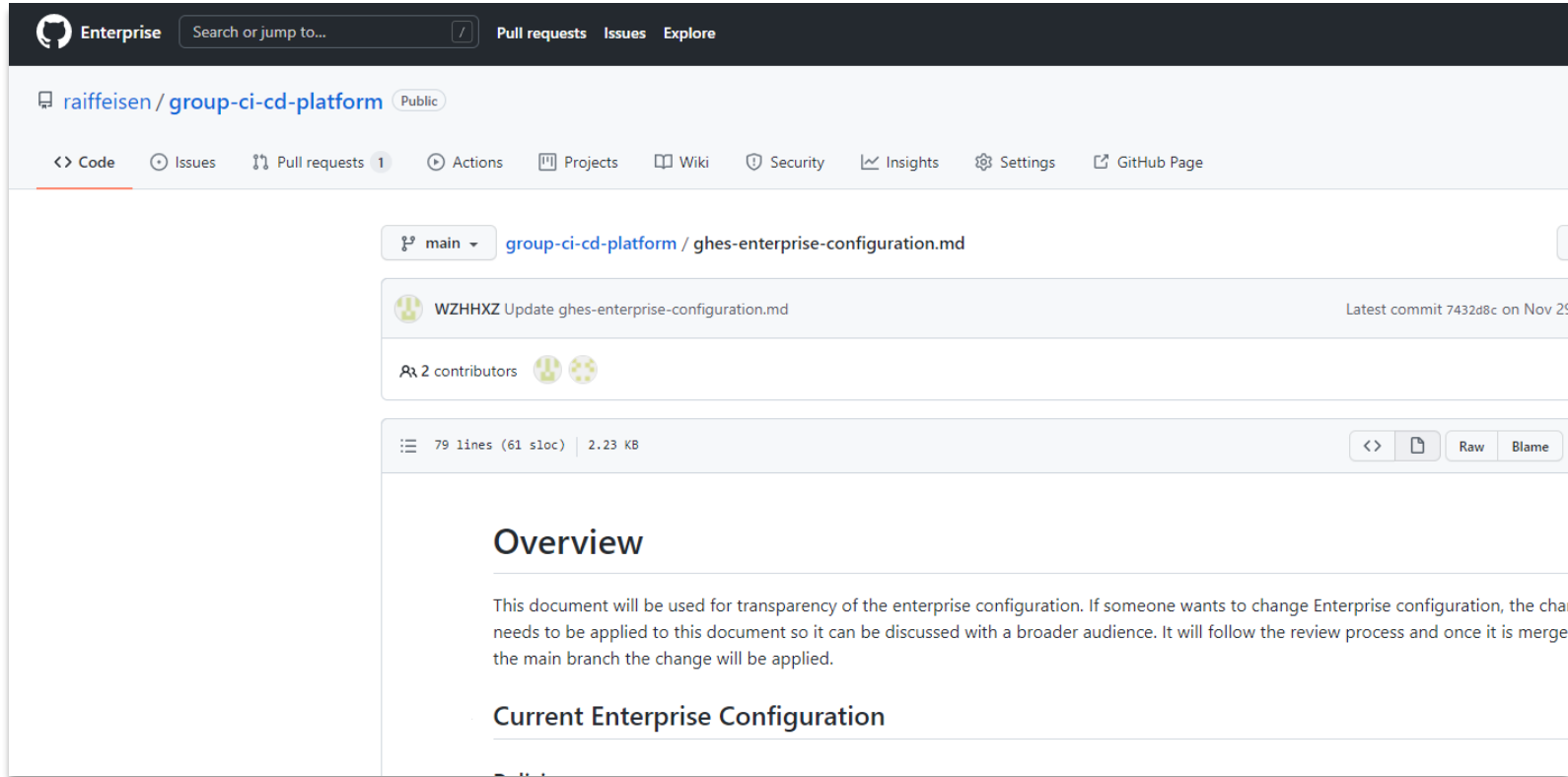


5.

Central Developer Platform



5. Central Developer Platform



6.

Inner Source Initiative



6. InnerSource Initiative

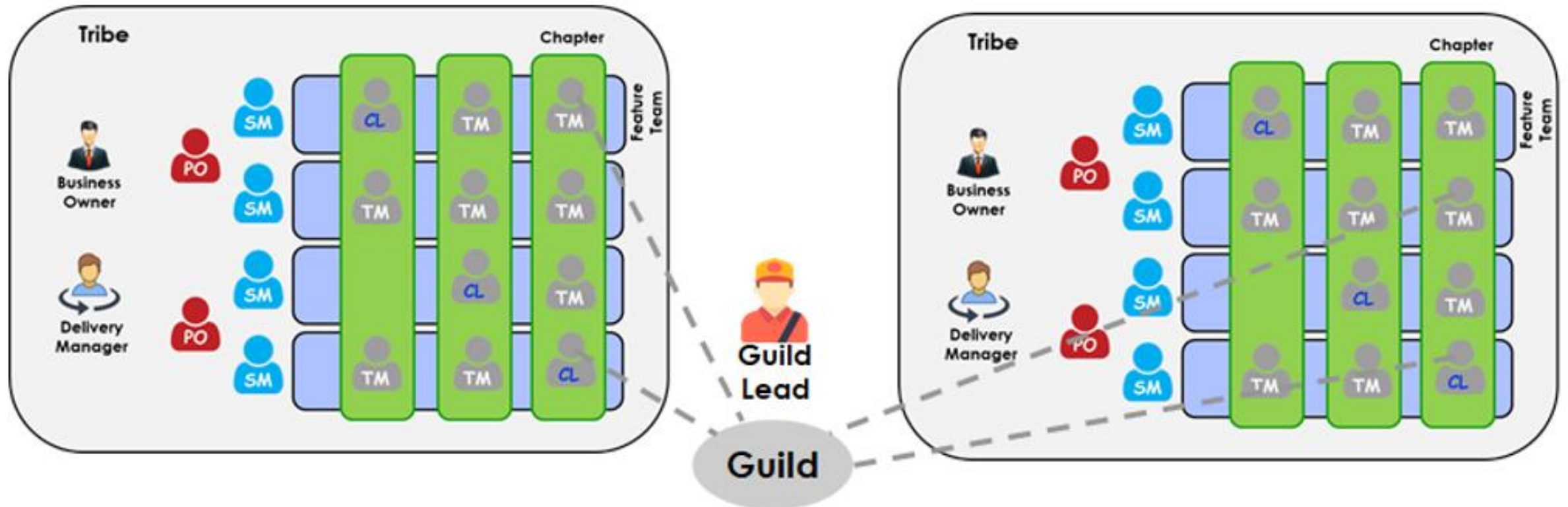


7.

Engineering Communities



7. Engineering Communities

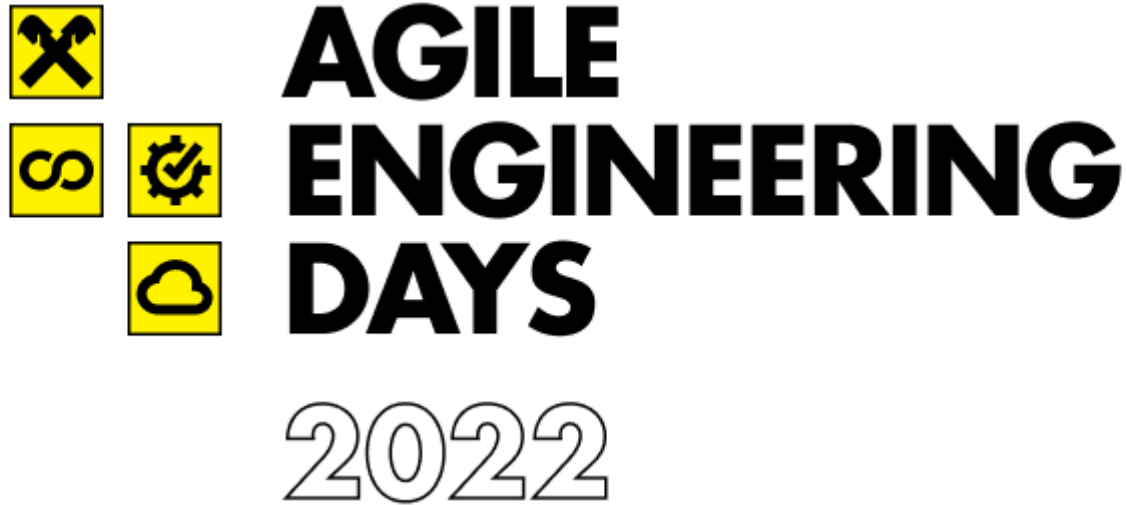


8.

Internal Engineering Conferences



8. Internal Engineering Conferences

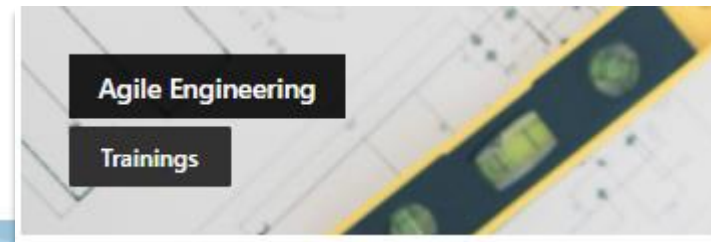


9.

Individual Learning



9. Individual Learning



9. Individual Learning

Learning Journeys

Learning

COURSES

All courses

My courses

Testing_TE

Testing_Foundation_Modules

Testing_Learning_Nuggets

Test Automation

Agile Engineering

DevOps






















Uncategorized

QUIZZES

My quizzes

ADMINISTRATION

Manage categories

<div>ALREADY ENROLLED</div> <div></div> <div>1-TE-Test Engineering</div> <div>Start</div>	<div>ALREADY ENROLLED</div> <div></div> <div>2-TAF-Test Automation Foundation</div> <div>Continue</div>	<div>ALREADY ENROLLED</div> <div></div> <div>3-TAA-Test Automation Adapter</div> <div>Start</div>	<div>ALREADY ENROLLED</div> <div></div> <div>4-TAD-Test Automation Development - API Python</div> <div>Start</div>	<div>ALREADY ENROLLED</div> <div></div> <div>4-TAD-Test Automation Development - Infra - InSpec</div> <div>Start</div>	<div></div> <div>T-DevOps-Agile Engineering Basics</div> <div>Enroll</div>	<div></div> <div>T-DevOps-Merlin</div> <div>Enroll</div>
<div>ALREADY ENROLLED</div> <div></div> <div>T-DevOps-Cortex-Basic</div> <div>Start</div>	<div></div> <div>T-DevOps-Merlin</div> <div>Enroll</div>	<div></div> <div>TAF-001 - Test Design Techniques Basic_Course</div> <div>Enroll</div>	<div></div> <div>TAF-002 - Test Automation Principles / Techniques</div> <div>Enroll</div>	<div></div> <div>TAF-003 - Overview Test Tools / Frameworks / Libraries_Course</div> <div>Enroll</div>	<div></div> <div>TAF-004 - Continuous Security Testing (DevSecOps)_Course</div> <div>Enroll</div>	<div></div> <div>TAF-005 - Tool Support Testing Basic_Course</div> <div>Enroll</div>
<div></div> <div>TAF-005A - The Regression Test Death Spiral_Course</div> <div>Enroll</div>	<div></div> <div>TAF-005A The Regression Test Death Spiral (LNugget)</div> <div>Enroll</div>	<div></div> <div>TE-001 - ISTQB Foundation Level Basic_Course</div> <div>Enroll</div>	<div></div> <div>TE-002 - Agile Testing Basic_Course</div> <div>Enroll</div>	<div></div> <div>TE-004 - Zephyr Basic_Course</div> <div>Enroll</div>	<div></div> <div>TE-005 - Tool Support Testing Basic_Course</div> <div>Enroll</div>	<div></div> <div>TE-006 - Selenium Basic_Course</div> <div>Enroll</div>



11	Baseline questions		min	average	max
12	Lead time to deliver a feature				
13	Monthly number of deployments				
14	Time to repair a production incident				
15	Amount of production incidents after go-live				
16					
17					
18	Overview result		CRAWL	WALK	RUN
19	CI/CD Automation		86%	71%	43%
20	DevOps Approach		100%	67%	37%
21	Test Automation		100%	83%	0%
22	Test Approach		75%	42%	16%
23	Security		0%	0%	0%
24					
25					
26	Detailed result		CRAWL	WALK	RUN
27	CI/CD Automation	Code	100%	100%	100%
28	CI/CD Automation	Code quality	0%	0%	0%
29	CI/CD Automation	Automation	100%	100%	0%
30	CI/CD Automation	Pipeline	100%	50%	0%
31	CI/CD Automation	Deployment	100%	50%	50%
32	CI/CD Automation	Environment ownership	100%	100%	50%
33	CI/CD Automation	Environment setup	100%	100%	100%
34	DevOps Approach	Operations	100%	100%	100%
35	DevOps Approach	Monitoring	100%	100%	0%
36	DevOps Approach	Incident management	100%	100%	100%
37	DevOps Approach	Release cycle	100%	100%	50%
38	DevOps Approach	Release impact	100%	100%	50%
39	DevOps Approach	Quality	100%	0%	0%
40	DevOps Approach	Development process	100%	0%	0%
41	DevOps Approach	Team	100%	67%	33%
42	DevOps Approach	Skills	100%	33%	0%
43	Test Approach	Test Planning & Control	0%	0%	0%
44	Test Approach	Test Analysis & Design	100%	33%	0%
45	Test Approach	Test Implementation & Execution	100%	50%	25%
46	Test Approach	Test Data	100%	50%	0%
47	Test Approach	Test Environment	100%	100%	0%
48	Test Approach	Test Doubles	100%	100%	100%
49	Test Approach	Test Documentation	0%	0%	0%
50	Test Approach	Test Skills	100%	0%	0%
51	Test Automation	Test Design	100%	100%	0%
52	Test Automation	Test Automation Development	100%	50%	0%
53	Test Automation	Test Execution	100%	100%	0%
54	Security	Security Design	0%	0%	0%

min	average	max
CRA	WALK	RUN
100%	93%	64%
100%	100%	56%
100%	83%	33%
94%	73%	38%
0%	0%	0%
CRAWL	WALK	RUN
100%	100%	100%
100%	100%	0%
100%	100%	0%
100%	100%	100%
100%	50%	50%
100%	100%	100%
100%	100%	100%
100%	100%	100%
100%	100%	0%
100%	100%	100%
100%	100%	0%
100%	100%	50%
100%	100%	50%
100%	100%	0%
100%	100%	100%
100%	100%	100%
75%	75%	50%
100%	33%	0%
75%	75%	50%
100%	100%	100%
100%	100%	0%
100%	100%	100%
100%	0%	0%
100%	100%	0%
100%	50%	0%
100%	100%	50%
100%	100%	50%
0%	0%	0%

Comparison to AEMM 2020

**We see
progress on
basically all
dimensions!**

Great job!

10. Engineering KPIs



10. Engineering KPIs

Metric	Explanation
Deployment Frequency	Refers to the frequency of successful software releases to production.
Lead Time for Changes	Captures the time between a code change commit and its deployable state.
Mean Time to Recovery	Measures the time between an interruption due to deployment or system failure and full recovery.
Change Failure Rate	Indicates how often a team's changes or hotfixes lead to failures after the code has been deployed.

Wrap-up



Short wrap-up of our toolbox

1. Clear Target
2. OKRs
3. Agile Engineering Maturity Model
4. Agile Engineering Coaches
5. Central Developer Platform
6. InnerSource
7. Engineering Communities
8. Internal Engineering Conferences
9. Individual Learning
10. Engineering KPIs



Did it work?

**Have you made
progress?**



Achievements

- Awareness about DevOps
- Knowledge about CI/CD and Testing
- Moved dozens of products to pipeline and automated build/test/deploy
- Improved Deployment Frequency and Lead Time considerable
- Big move to the public cloud
- Established lively communities
- Hundreds of people have improved knowledge and skills



The DevOps Toolbox



TIME

Thank You!



Questions for the Community:

1. It is sometimes difficult to motivate people to contribute to communities. Any good experiences that we can learn from?
2. Is there good tooling available around the collection of DORA metrics? We do have something in place, but I am sure there must be quite some demand? Any experiences with tools?
3. Is anybody else doing qualitative capability assessments? Based on any frameworks? We would like to improve here.

