

# How Dev(Sec)Ops are you?

DevOps Conference Berlin  
30.05.2018

Dr. Martin Luckow  
Senior Solution Manager



BASEL ■ BERN ■ BRUGG ■ DÜSSELDORF ■ FRANKFURT A.M. ■ FREIBURG I.BR. ■ GENÈVE  
HAMBURG ■ KOPENHAGEN ■ LAUSANNE ■ MÜNCHEN ■ STUTTGART ■ WIEN ■ ZÜRICH

**trivadis**  
makes IT easier. ■ ■ ■

# ■ Trivadis with over 650 Specialists and IT Experts



- 14 Trivadis branches and more than 650 employees, growing
- Near- and offshore partners with a dedicated, scalable Trivadis team
- 200 Service Level Agreements
- Over 4'000 training participants per year
- Research and development budget: CHF 5.0 million per year
- Financially self-supporting and sustainably profitable
- Experience from more than 1'900 projects...

# ■ Trivadis Portfolio (Weblinks)

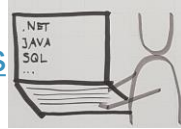
## Analytics, BI & Big Data

- [Advanced Analytics & Reporting](#)
- [Analytical Data Management](#)
- [Big Data & Data Science](#)



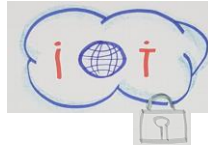
## Application Development

- [Application Modernization](#)
- [Individual Software Solutions](#)
- [Mobile](#)



## Cloud

- [Cloud Solutions](#)
- [IoT](#)



## Digitalization & Integration

- [Digital Integration](#)
- [Digital Transformation](#)
- [Digital Performance Management](#)



## Infrastructure

- [Infrastructure Architecture](#)
- [Infrastructure Engineering](#)



## Innovation

- [Trivadis Innovation Partnership Program](#)
- [Trivadis TechEvent](#)
- [Drone Deer](#)
- [IoT Windpark](#)
- [Smart Building and Co-Working Space](#)
- [Innovation Space: The digital Eco-System](#)



## Services for all Areas

- [Requirements Engineering](#)
- [Project Management](#)
- [Managed Services](#)
- [Products](#)
- [Training](#)



# ■ A path to DevSecOps



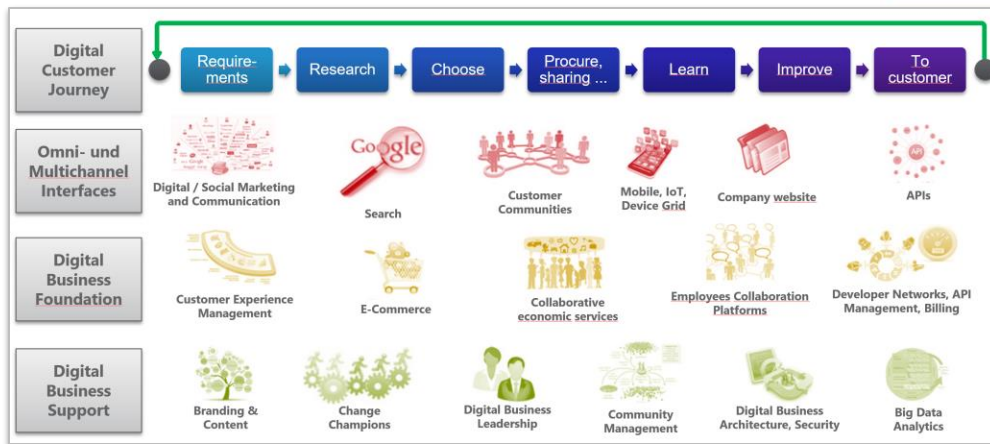
- Looks like waterfall, but it is not 😊
- When you reach 8, the game starts again. However, you have more experience
- Is an iterative self-improvement process. It can be measured and steered.



# A path to DevSecOps



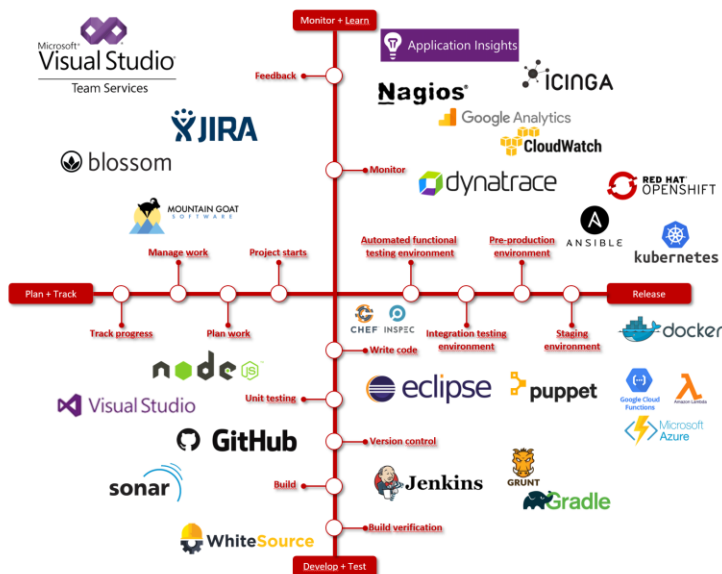
- Looks like waterfall, but it is not ☺
- When you reach 8, the game starts again. However, you have more experience
- Is an iterative self-improvement process.



# A path to DevSecOps



- Toolchain development is late in this process
- Has to be redone in 8



# ■ A path to DevSecOps



- Looks like waterfall, but it is not 😊
- When you reach 8, the game starts again. However, you have more experience
- Is an iterative self-improvement process. It can be measured and steered.

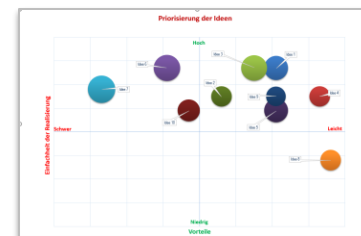
Application for a benchmark





The diagram illustrates the SecOps lifecycle, which integrates DevOps and SecOps processes. It is divided into two main sections: Dev (Development) and Ops (Operations). The DevOps cycle includes Create, Plan, Deploy, and Verify, with a central Monitoring & Analytics component. The SecOps cycle includes Prevent, Detect, Respond, and Predict, with a central Monitoring & Analytics component. A central 'SecOps' component connects the two cycles, and a 'DevOps' component connects the Dev and Ops cycles.

- 
- | Dimension            | Maturity Score |
|----------------------|----------------|
| Automation and tools | 3              |
| Business Value       | 1.5            |
| People & Culture     | 1.5            |
| DevOps agility       | 0.5            |
| Security             | 0              |

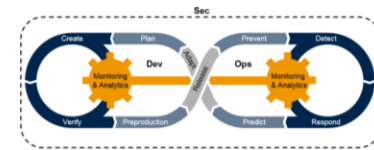
[illegible]

8 28.02.2020 DevSecOps Benchmark



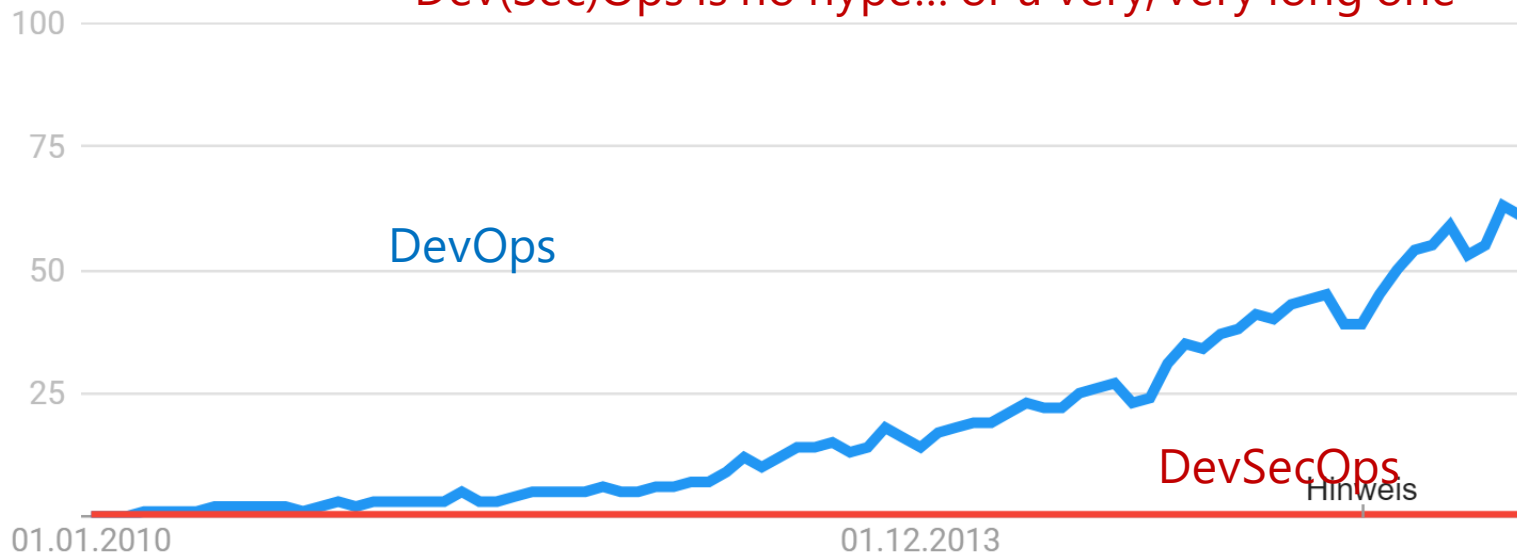
# Some remarks to the „Sec“...

# ■ Why „Sec“?

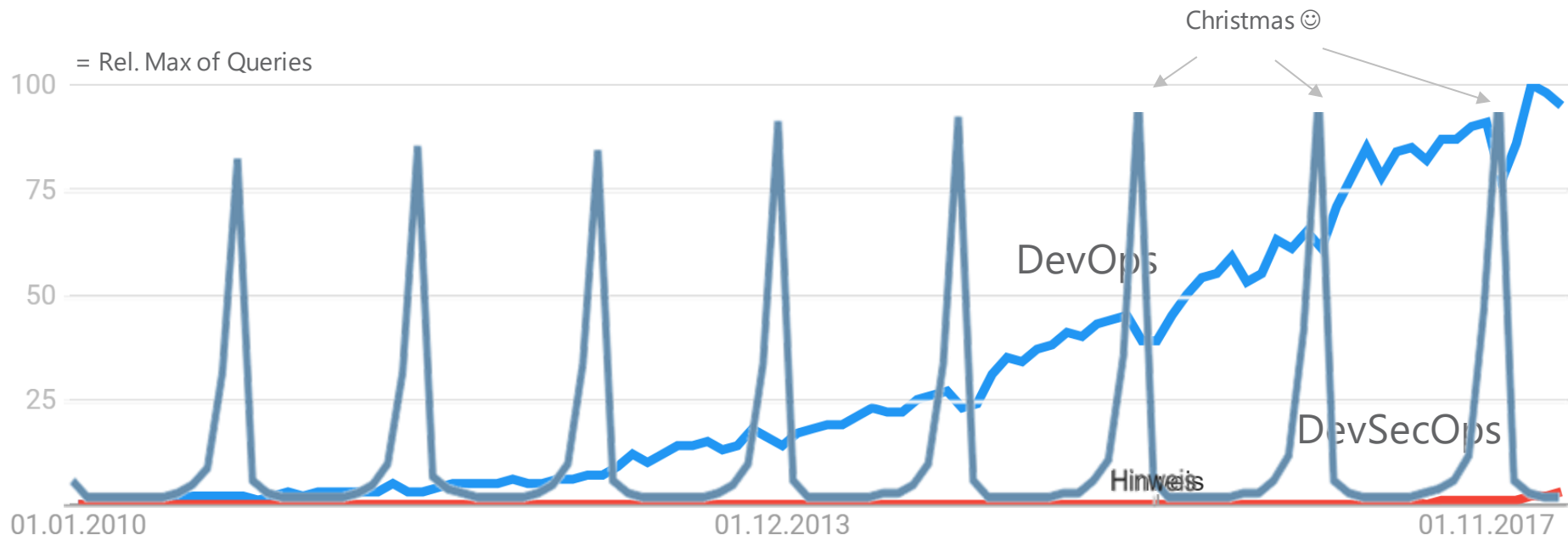
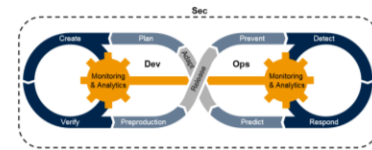


= Rel. Max of Queries

Dev(Sec)Ops is no hype... or a very, very long one



# ■ Why „Sec“?

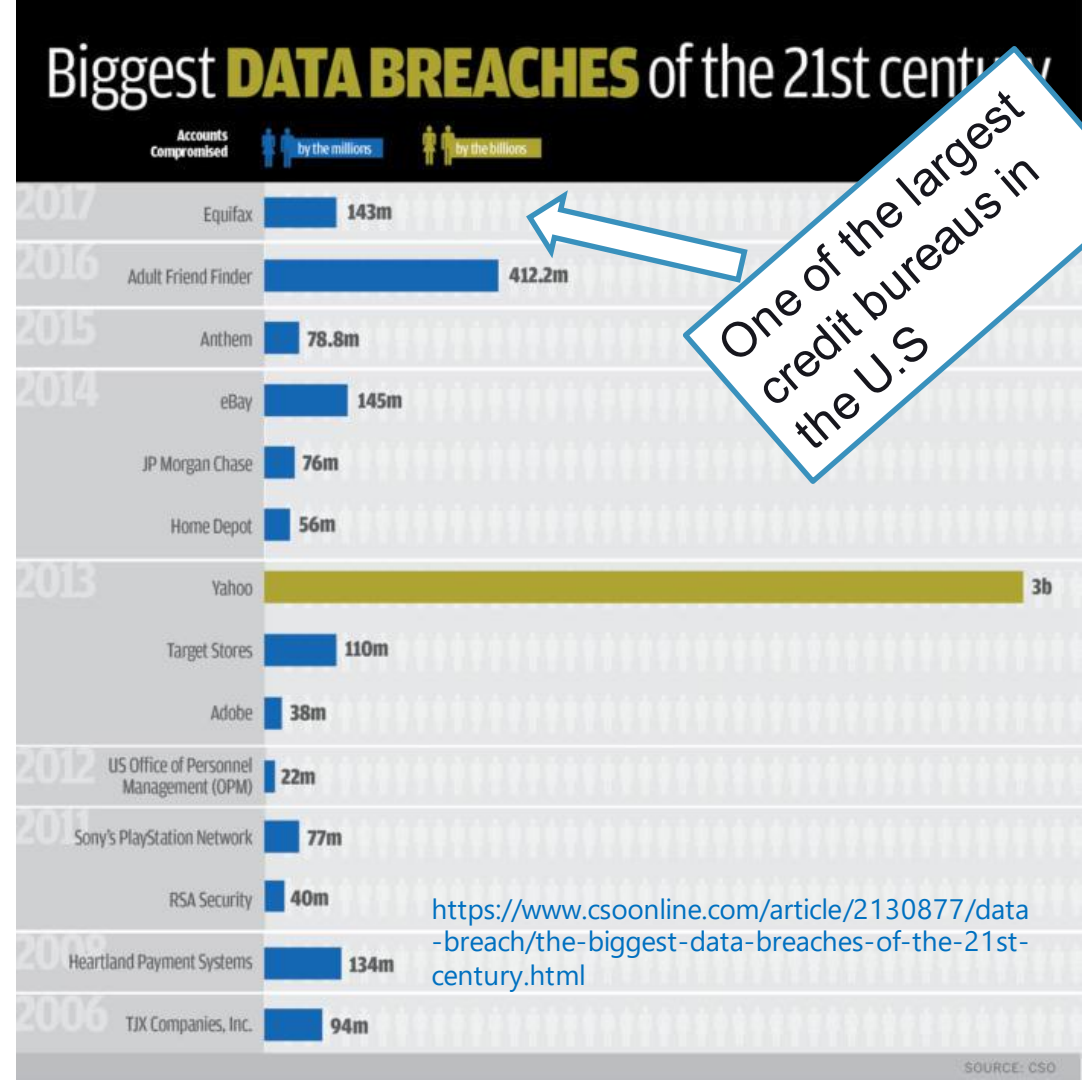


# ■ Why „Sec“?

Nobody wants to be on such a list...

Equifax, July 29 2017

- Personal information (including Social Security Numbers, birth dates, addresses, and in some cases drivers' license numbers) of 143 million consumers
- 209,000 consumers also had their credit card data exposed



# ■ Why „Sec“?

<https://staysafeonline.org/>

And the smaller companies?

- Almost 50 percent of small businesses have experienced a cyber attack
- More than 70 percent of attacks target small businesses
- As much as 60 percent of hacked small and medium-sized businesses go out of business after six months



# ■ Why „Sec“?

<https://staysafeonline.org/>

- How do the costs of a security issue add up across 6 categories?



29%  
Reputation and  
brand damage



21%  
Lost  
productivity



19%  
Lost  
Revenue



12%  
Forensics



10%  
Technical  
support



8%  
Compliance  
Regulatory



<https://thenextweb.com/security>

# ■ Why „Sec“? Gartner Strategic Trends 2018

Intelligent

Digital

Mesh



AI Foundation

Intelligent Apps  
and Analytics

Intelligent Things



Digital Twins

Cloud to the Edge

Conversational Platforms

Immersive Experience



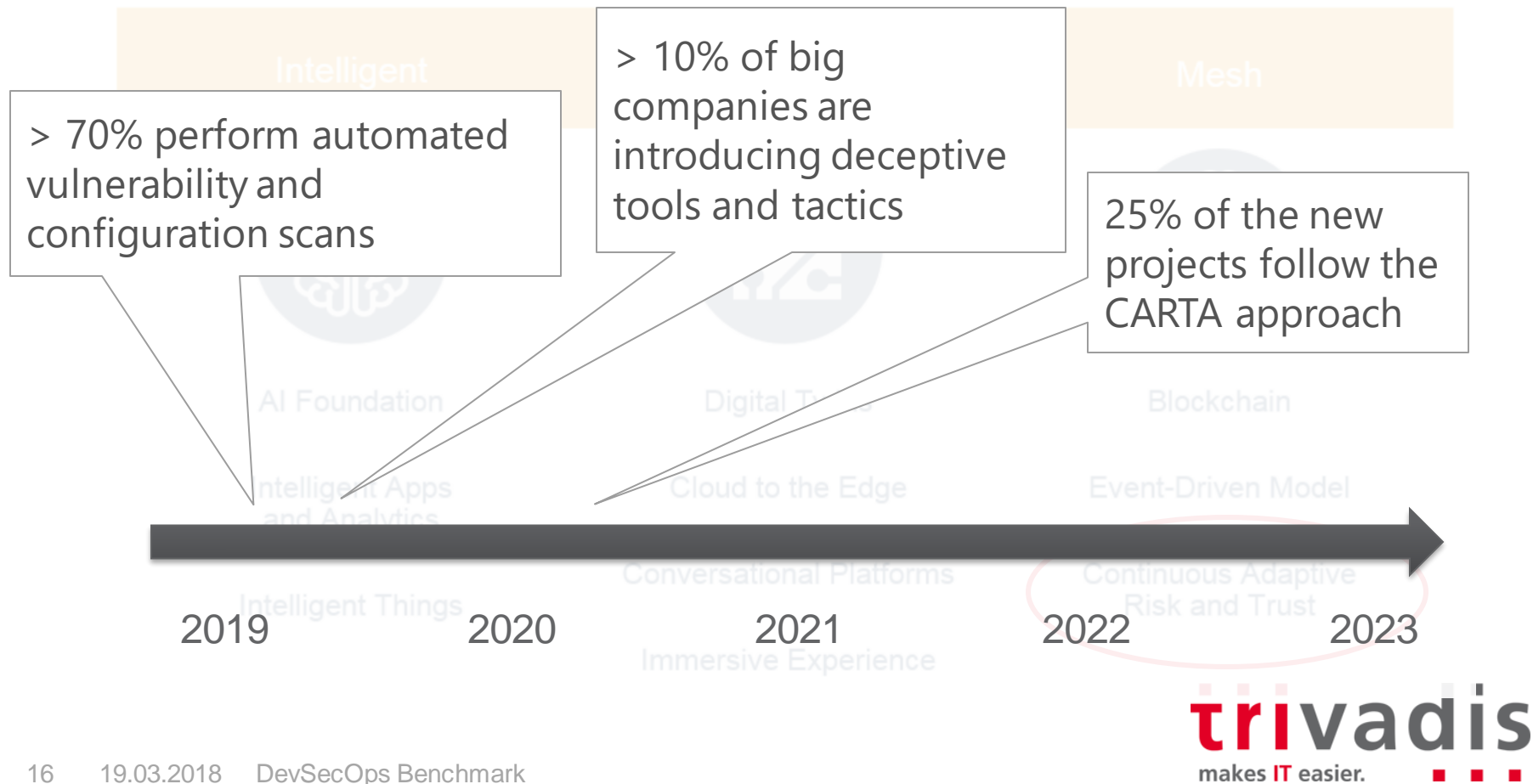
Blockchain

Event-Driven Model

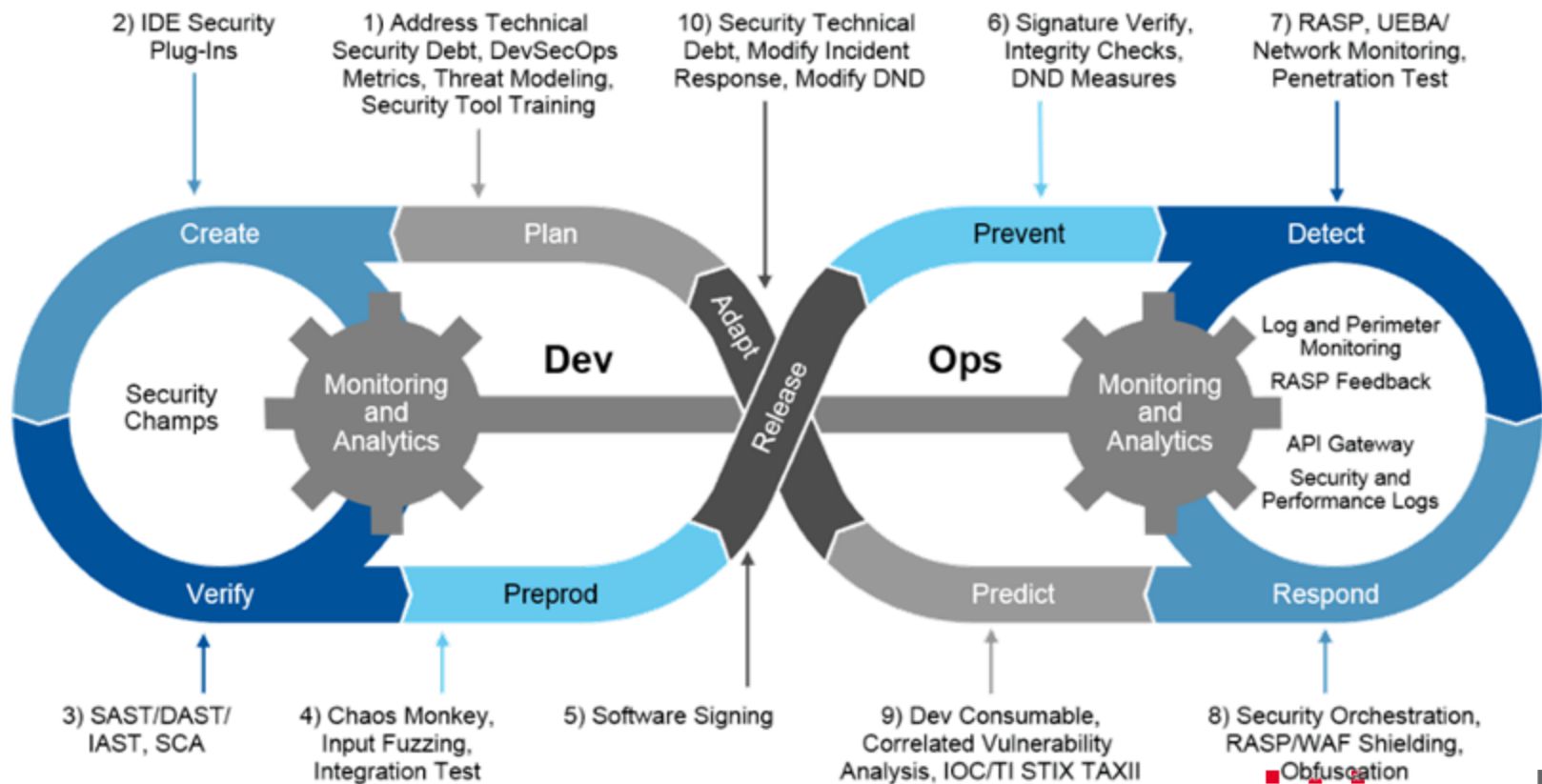
Continuous Adaptive  
Risk and Trust



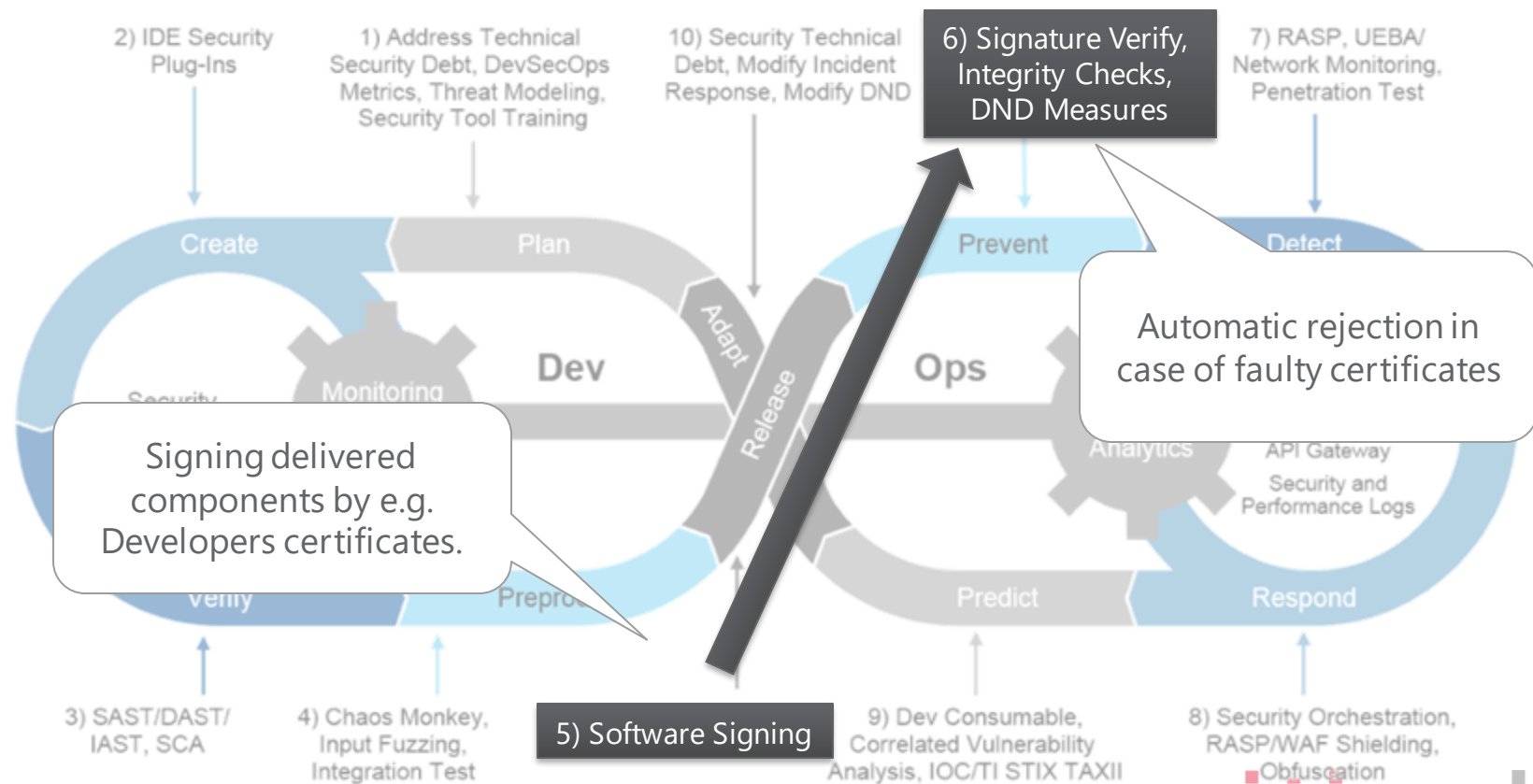
# ■ Why „Sec“? Continuous Adaptive Risk and Trust



# Continuous Adaptive Risk and Trust: build-in security

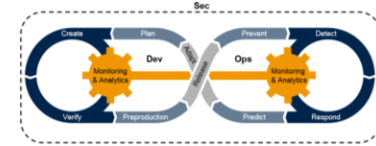


# Continuous Adaptive Risk and Trust: build-in security



# Back to the benchmark ... how to create?

# 1. Define the „why“ of the benchmark



- Understand what DevSecOps is about (What... not how)
- Recognize that it is a transformation process
- Assess where you are in this process right now. Find some easy-to-understand KPIs to describe this
- Create discussions between as many stakeholders as possible
- Develop ideas about what to do next and what not
- Regular progress control





### 3. Define themes and categories



- Look at DevOps from different angles (themes)
- Later define categories for each theme to create even more structure

People and Culture	DevOps agility	Business Value	Automation and tools	Security
Characteristics of organizational culture. The theme includes statements on customer focus, innovation, risk management and change management.	Organizational agility characteristics: maturity of e.g. Scrum, Kanban, use of agile practices, lean management.	Organizational characteristics that affect business value creation.	Statements about the use of automation testing, continuous integration and continuous deployment practices and tools.	Statements about the degree of integrated and lived security. Information security, privacy, technical security, governance and compliance.



## ■ 4. Define and assign statements



Each statement

- describes a fact. This is either true or false for an organization
- is assigned to exactly one topic and
- is assigned to exactly one maturity level
- Statements are categorized within a theme to steer the discussion, e.g. "Change management", "Testing", "Monitoring", ...
- Statements of varying degrees of maturity build successively on each other.

Some of us do not care about the result. They do minimal business.

Everyone in our team is willing to share & learn

There are no coding guidelines or they are not always respected.

We proactively participate to improve company policies

Sometimes our stakeholders are surprised by changes in the product.

Our Stakeholders are updated on roadmap progress each month

## 4. Define and assign statements



Business Value

	Level 1 Minimal	Level 2 Basis	Level 3 Intermediate	Level 4 Transformed	Category
	Sometimes our stakeholders are surprised by changes in the product.	We inform our Stakeholders about the changes in the product	Our Stakeholders review the changes in the product and provide us feedback on that	Our Stakeholders actively participate in the definition and outcome of the product	Feedback loops
	Larger changes are not always thought through in their effects.	The Product Owner or our Stakeholders create a business case for each accepted epic	Our Product Owner and Stakeholders jointly create a business case for any new epic	Our Stakeholders discuss business cases with the PO and other stakeholders, to come to clear prioritization	Business value
	Only a few decide how to prioritize the implementation of new features.	We collect input from relevant stakeholders when defining priorities.	We collect feedback from Customers and prospects when defining new features.	We make sure analysis of the market needs and prospected ROI are done, when creating a new feature	Value steering
	Our roadmap is not one. It is sometimes not traceable	We have an agile roadmap	Our Stakeholders are updated on	We plan based on the velocity	Roadmap planning
	Increasing requirements				
	We do not know, or only vaguely, how new functions are accepted by the user.	We measure the usage of new features	We analyze the business cases to verify if they have been realized	The usage result of the new features is verified with all internal stakeholders, customers and Market	Validate value
	Not every one of our stakeholders knows why we realize what and when.	Our Stakeholders understand the backlog we proactively explain it. We measure this on a monthly basis.	Our Stakeholders feel they have an important role in the Backlog prioritization. We measure this on a monthly basis.	Our Stakeholder are satisfied with the features delivered by our team. We measure this on a monthly basis.	Stakeholder Happiness
24	28.02.2020	DevSecOps Benchmark			We have a Stakeholder radar in place, and published.



# 4. Define and assign statements

■ Done 😊

■ The current assessment consists of 264 statements

■ Each statement can lead to (heated) discussions during the assessment. Then you need a mediator 😊

■ At least one should know exactly what is behind the statements - a moderator is beneficial

	Level 1 Minimal	Level 2 Basic	Level 3 Intermediate	Level 4 Transformed	Category
Business Value	Sometimes our stakeholders are surprised by changes in the product.	We inform our Stakeholders about the changes in the product.	Our Stakeholders review the changes in the product and provide us feedback on that.	Our Stakeholders actively participate in the definition and outcome of the product.	Feedback loops
	Larger changes are not always thought through in their effects.	The Product Owner or our Stakeholders jointly create a business case for each accepted epic.	Our Product Owner and Stakeholders jointly create a business case for any new epic.	Our Stakeholders discuss business cases with the PO and other stakeholders, to come to clear prioritization.	Business value
	Only a few decide how to prioritize the implementation of new features.	We collect input from relevant stakeholders when defining priorities.	We collect feedback from Customers and prospects when defining new features.	We make sure analysis of the market needs and projected ROI are done, when creating a new feature.	Value steering
	Our roadmap is not one. It is sometimes not traceable overturned.	We have an agile roadmap.	Our Stakeholders are updated on roadmap progress each month.	We plan based on the velocity report and realize these plans. We follow a process when introducing changes to the roadmap.	Roadmap planning
Automated deployments	We do not know how new functions are implemented.	Own code and infrastructure changes are not only sometimes checked by the team.	All our code and infrastructure changes are reviewed by at least 3 other team members.	We have zero touch continuous deployments.	Automated deployments
	Not every one knows why we change things.	Our pull requests are sometimes merged without verification.	Our pull requests are only merged after review and we enforce this in Slack.	We always roll forward, do not rollback.	
		The deployment is done without our involvement.	We deploy all our applications / infrastructure ourselves (manual or automated).	Our deployments do not impact our service.	
		We build our applications manually on request.	All our code is versioned.		
Testing			All of our applications can be build in minutes and we have them in place where time consuming, high risk features can be tested.	We run automated functional regression, performance and security tests for all applicable areas this is applicable.	Testing
				We have a functional test automation framework in place where time consuming, high risk features can be tested.	
				Unit tests run automatically in every Jenkins build and we fix them when they fail.	
				We have insight into the quality (test coverage, complexity) of the code.	
People & Culture	We sometimes have quarrels in the team and finger pointing.	We measure & share team happiness each sprint.			
	Mistakes are reported to others but we do not learn from them.	We register failures, so we can use this as an opportunity to learn. We no longer display in a sprint after 2 failed deployments (error budget).			
	Some of us do not share their knowledge. Some decisions remain opaque.	We proactively share information with the whole team.			
	Some of us take no responsibility.	We understand each other's concepts, concerns and problems (of different expertise).			
Architecture and Design		We have our own ITD (ITD) to respond when services are down.			
Security	Some of us do not result. They do not result.	We do not have an agile process model like Scrum.			
	We are proclivity that is, we are told to do.	We plan to resolve incidents outside of our team. We only resolve the order for confidentiality.			
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					
Security					

# Next: Perform an assessment



# 1. Get the right people

- Everyone from the (planned) DevOps team
- Every direct stakeholder, especially product managers and IT managers
- If possible the senior management

... plan to block them for at least one day

	Level 1 Minimal	Level 2 Basis	Level 3 Intermediate	Level 4 Transformed	Category
Business Value	Sometimes our stakeholders are surprised by changes in the product.	We inform our Stakeholders about the changes in the product.	Our Stakeholders review the changes in the product and provide us feedback on that.	Our Stakeholders actively participate in the definition and outcome of the product.	Feedback loops
	Larger changes are not always thought through in their effects.	The Product Owner or our Stakeholders jointly create a business case for any new epic.	Our Product Owner and Stakeholders jointly create a business case for any new epic.	Our Stakeholders discuss business cases with the PO and other stakeholders, to come to clear prioritization.	Business value
	Only a few decide how to prioritize the implementation of new features.	We collect input from relevant stakeholders when defining priorities.	We collect feedback from Customers and prospects when defining new features.	We make sure analysis of the market needs and projected ROI are done, when creating a new feature.	Value steering
	Our roadmap is not one. It is sometimes not traceable overturned.	We have an agile roadmap.	Our Stakeholders are updated on roadmap progress each month.	We plan based on the velocity report and realize these plans. We follow a process when introducing changes to the roadmap.	Roadmap planning
People & Culture	We do not know how new functionality is checked by the team.	Our code and infrastructure changes are not only sometimes checked by the team.	All our code and infrastructure changes are reviewed by at least 1 other team member.	Push button deployment and release of any releasable artifact to any environment.	Automated deployments
	Not every one knows why we do things.	Our pull requests are sometimes merged without verification.	Our pull requests are only merged after review and we enforce this in Slack.	Our integration test environment is automatically updated after merge to master branch.	We always roll forward, do not rollback.
		The deployment is done without our involvement.	We deploy all our applications / infrastructure ourselves (manual or automated).	Pull requests can only be merged after a successful build and we enforce this in Slack.	Our deployments do not impact our service.
		We build our applications manually on request.	All our code is versioned.	We have automatic generation of release notes via the upcoming period.	
Architecture and Tools	We cannot find how questions in the team and finger pointing.	We measure & share team happiness each sprint.	We register failures, so we can use this as an opportunity to learn.	All of our code are currently working has a minimum of 90% good quality unit test coverage.	We run automated functional regression, performance and security tests for all applications where this is applicable.
	Mistakes are researched to others but we do not learn much from them.	We no longer deploy in a sprint after 2 failed deployments (error budget).	We have a functional test automation framework in place where time consuming, high risk features can be tested.	Unit tests run automatically in every Jenkins build and we fix them when they fail.	We run a minimum of 90% good quality unit test coverage.
	Some of us do not share their knowledge. Some decisions remain opaque.	We understand each other's concepts, concerns and problems (of different expertise).			
	Some of us take no responsibility.	We proactively share information with the whole team.			
Security	We have our own ITD (IT) to respond when services are down.	Our team is able to fully develop, maintain and support our products, applications and infrastructure (also in terms of skills & resources).			
		We measure time to resolve incidents and our active incidents are displayed near time and complexity.			
	We are more reactive to change management, that is, we are sometimes unexpectedly asked to do things. Logging sometimes does not take place. Sometimes, changes in us have unmanageable effects.	We understand the change management process and register all relevant infrastructure deployments & customer impacting changes. We make sure architecture & security assessment are part of each change. We register peer reviews of all changes (code & infra).			
	There is no defined architecture or sometimes we do not stick to it.	We understand the architecture policies, and decisions from it are identified.			
DevOps Maturity	Documentation is often done afterwards. It happens that product and documentation are delivered separately.	All our new deliveries & changes are accompanied with the desired documentation.			
	Not every one of us knows the company's policies and they do not have much to do with development and operations.	We follow the company policies for new deliveries & changes.			
	There are no common production standards for development and operation.	Our team is aware of the production standards.			

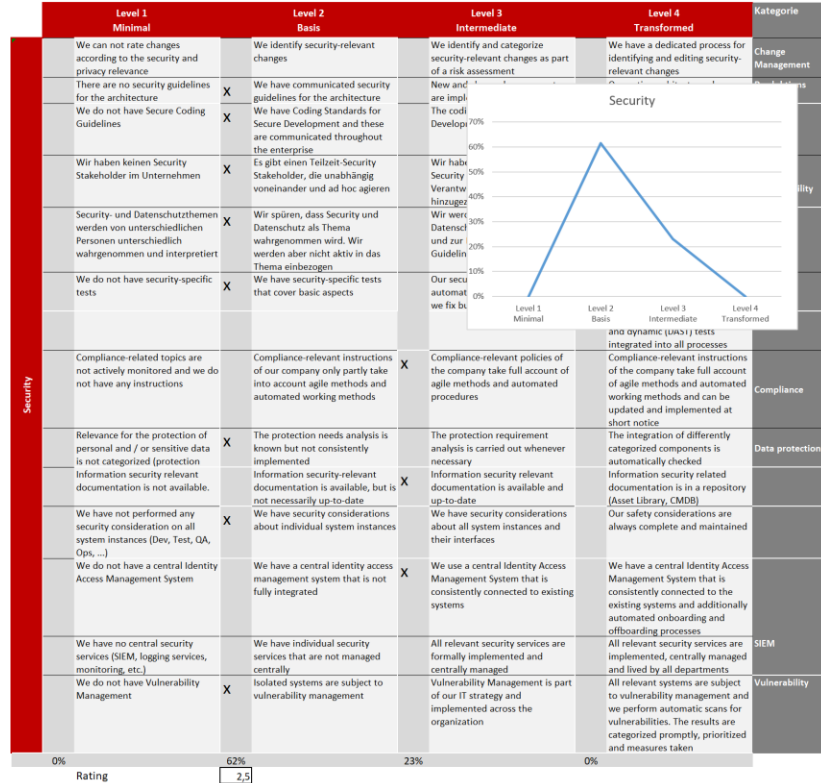


- Get people physically together. No conference calls! Care for the right environment
  - Explain the procedure
  - Ensure good logging of discussions
  - Work through the themes and estimate the maturity level for each theme
  - Then consolidate the results
- For several DevOps teams:
- Collect the results and do the assessment with emissaries from the teams
  - Create a consolidated view

[illegible]

## 2.1 For each theme...

- Work line by line from left to right through the theme
- Discuss each statement and check the correct statement with the highest degree of maturity
- Also look at the statements of the higher levels that you can not check. What has to happen to get there?
- What has to happen so that you do not suddenly lose a set checkmark next time?
- Then estimate the maturity level for the topic based on the checkmarks
- Log everything!
- And take a break after each theme ☺



... expect discussions per statement between 1 – 60 minutes





## 2.2 Consolidate

- You can determine a total degree of maturity by averaging

In subsequent workshops:

- Refine the logged statements and ideas
- Ensure prioritization
- Care for the implementation

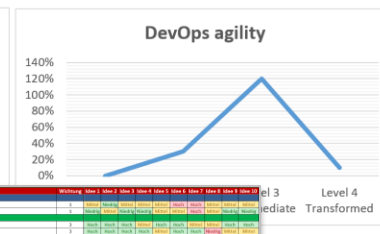
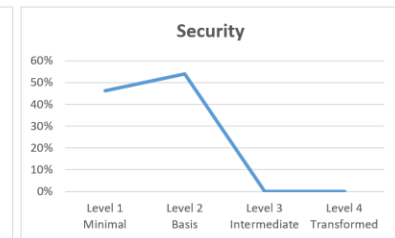
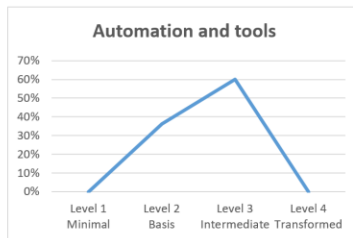
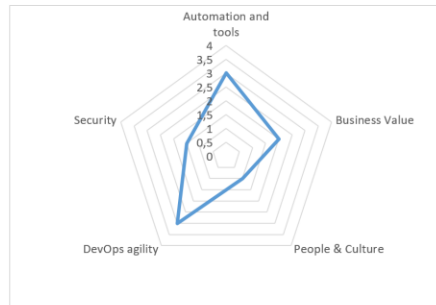
And schedule the repetition of the assessment at regular intervals

## DevOps Toolkit

### DevOps Benchmark

In this table, note the degree of maturity that you have given each of the pillars, and then give the organization an overall rating. The values are preset with the values from the previous pages.

Thema	Maturity Level Rating
Automation and tools	3
Business Value	2
People & Culture	1
DevOps agility	3
Security	1,5
Zusammenfassende Bewertung	2,1



Thema	Stadium	Stadium 1	Stadium 2	Stadium 3	Stadium 4	Stadium 5	Stadium 6	Stadium 7	Stadium 8	Stadium 9	Stadium 10
Automation and tools	3	1	2	3	4	5	6	7	8	9	10
Business Value	2	1	2	3	4	5	6	7	8	9	10
People & Culture	1	1	2	3	4	5	6	7	8	9	10
DevOps agility	3	1	2	3	4	5	6	7	8	9	10
Security	1,5	1	2	3	4	5	6	7	8	9	10
Zusammenfassende Bewertung	2,1	1	2	3	4	5	6	7	8	9	10



# Outcome

- A common understanding of what DevOps means to the organization
- An estimate of where you are in the transformation process. Simple KPIs that even a manager understands 😊
- Discussions between stakeholders and thus the basis for transparency
- Ideas as input for the improvement process

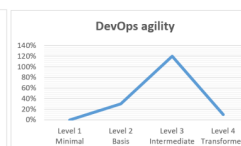
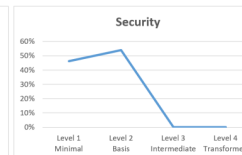
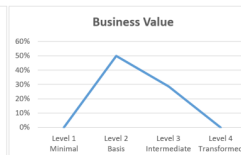
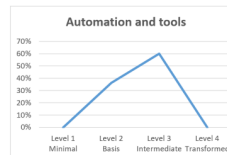
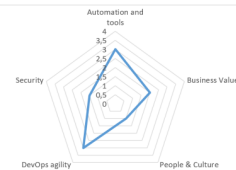
... and a regular progress check

## DevOps Toolkit

### DevOps Benchmark

In this table, note the degree of maturity that you have given each of the pillars, and then give the organization an overall rating. The values are preset with the values from the previous pages.

Thema	Maturity Level Rating
Automation and tools	3
Business Value	2
People & Culture	1
DevOps agility	3
Security	1,5
Zusammenfassende Bewertung	2,1



We will gladly send you a detailed report of your results with suggestions.

Name	<input type="text"/>
Prename	<input type="text"/>
Company	<input type="text"/>
E-Mail	<input type="text"/>
Phone	<input type="text"/>
Role	<input type="text"/>



# ■ DevSecOps Online Benchmark (experimental)



- Gives an impression of how the benchmark works.
- Has (currently) the full scope of 5 themes, 4 levels and 264 statements/facts.
- Gives first ideas, where to stand and what to do
- Can NOT replace a moderated assessment
- Updated regularly

<https://assessments.trivadis.com/devops>

Theme	Level	Statement/Fact	Score
Architecture	Basic	There is a defined architecture and it is not subject to change.	1
	Intermediate	We have a defined architecture and it is not subject to change.	2
	Advanced	We have a defined architecture and it is not subject to change.	3
	Expert	We have a defined architecture and it is not subject to change.	4
Change Management	Basic	We have a defined change management process and it is not subject to change.	1
	Intermediate	We have a defined change management process and it is not subject to change.	2
	Advanced	We have a defined change management process and it is not subject to change.	3
	Expert	We have a defined change management process and it is not subject to change.	4
Compliance	Basic	We have a defined compliance process and it is not subject to change.	1
	Intermediate	We have a defined compliance process and it is not subject to change.	2
	Advanced	We have a defined compliance process and it is not subject to change.	3
	Expert	We have a defined compliance process and it is not subject to change.	4
Development	Basic	We have a defined development process and it is not subject to change.	1
	Intermediate	We have a defined development process and it is not subject to change.	2
	Advanced	We have a defined development process and it is not subject to change.	3
	Expert	We have a defined development process and it is not subject to change.	4
Incident Response	Basic	We have a defined incident response process and it is not subject to change.	1
	Intermediate	We have a defined incident response process and it is not subject to change.	2
	Advanced	We have a defined incident response process and it is not subject to change.	3
	Expert	We have a defined incident response process and it is not subject to change.	4

**trivadis**  
makes IT easier.

■ <http://m.trivadis.com/DevSecOps>

**Trivadis DevOps**

Trivadis makes IT

Bringen Sie ... zusammen.

**DOAG**

Visit us @

-   
Kontinuierliche Verbesserung
-   
Kontinuierliche Auslieferung
-   
Kontinuierliche Integration
-   
Kontinuierliches Testen
-   
Kontinuierliches Monitoring

# Thank you

