S²AFe- DevSecOps.

A Scaled Agile approach to Security in DevOps organizations.

Introduction: Why?

Security in IT is fundamentally hard.

What drives this difficulty is the challenge of what is known as ***threat asymmetry***.

This is where a defender must protect against *all* attack vectors consistently, however an attacker only needs to find a single vulnerability to launch a successful attack.

Security at scale is ***very hard.***

Our intent with this paper is not to talk about the latest buzzword in technology or the latest tooling in the DevSecOps pipelines, rather to address the fundamental principles of a working DevSecOps security strategy – at scale.

Note: This paper builds upon the solid DevOps scalable foundation of SAFe. A basis understanding of SAFe and the principles of agile development would be highly beneficial in getting the most out of what is presented here. The waterfall methodology will not be covered.

Consistency across DevOps organisations are required to be successful in designing reliable and secure services.

In large organizations with many value streams and hundreds of teams this is a non-trivial issue to resolve consistently.

In SAFe there are some provisions for Security and quality as part of "NFR's" (Non-Functional Requirements) however these topics are not well explored in the model and therefore tend to be forgotten when defining the security organization and planning of activities.

Similarly, "built in quality" is mentioned at the team level… but who represents quality at the team level? The truth is that this is undefined and leaves important quality topics (such as security and regulatory compliance) on the table.

As a result, at our company (Swisscom Switzerland) we have developed an approach that allows a consistent and scalable method to addressing Security in the first line. This model has been developed for Security, Data Governance and addressing quality/Non-Functional requirements in general. It has been benchmarked as part of ETIS Telco Security

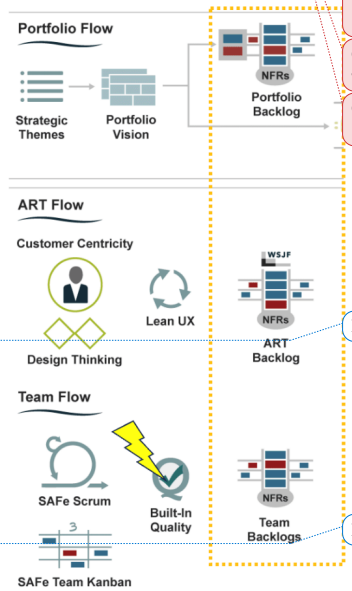


Figure 1: Potfolio SAFe 6.0 Diagram of NFR's and Quality -

Commented [C(1]: I probably would also add some text about the different security approaches if you work agile-based or in a waterfall model: How security is done in the old model (waterfall) and why it no longer works in the agile world. Static checkpoints, manual approvals, etc. And in agile with continuous releases and automation, security must also be developed in this direction.

Commented [C(2R1]: Or if not, then add to the note as well, that this is a prerequisite.

Commented [C(3R1]: Maybe a link to the old DevSecOps Booklet

Deleted: Similarly

Deleted: approach

Benchmark 2023 and was clarified as being best in class. Hence, we feel there is value in sharing the model and our experiences.

Roles and Responsibilities (following standard SAFe Methodology)

In SAFe NFR's (Non-Functional Requirements) are handled by the Solution and System Architects. To quote the official SAFe methodology website:

Deleted: Methadology

Nonfunctional Requirements

Nonfunctional Requirements (NFRs) are system qualities that guide the design of the solution and often serve as constraints across the relevant backlogs.

As opposed to functional requirements, which specify how a system responds to specific inputs, nonfunctional requirements are used to specify various system qualities and attributes, such as:

- **Performance**: How fast a system should respond to requests
- **Scalability**: How well a system can handle an increase in users or workload
- **Security**: How well a system protects against unauthorized access and data breaches
- **Usability**: How easy a system is to use
- **Maintainability**: How easy it is to update and modify the system

NFRs are persistent qualities and constraints typically revisited as part of the definition of done (DoD) for each Iteration, PI, or release. NFRs influence Teams, ART, Solution Train, and Portfolio backlogs.

Figure 2 Excerpt from: https://scaledagileframework.com/nonfunctional-requirements/

In our experience, given the specialist knowledge required to manage security, Architects are ill-equipped to take this responsibility without additional Security training.

Deleted: As you can see there is a small inconsistency

This does and will lead to poor outcomes if not addressed. Let us elaborate.

Architects do not operate at the team level (Only ART/Value Stream/Portfolio level) so the question arises: how can they effectively manage these topics without representation in the team?

This is depicted in the following Full SAFe model… (Sec) is included at the bottom right-hand corner of the model… but shows no way in which System Architects or Solution Architects can steer the outcomes. This inconsistency is the reason we have seen the need to develop this model further.

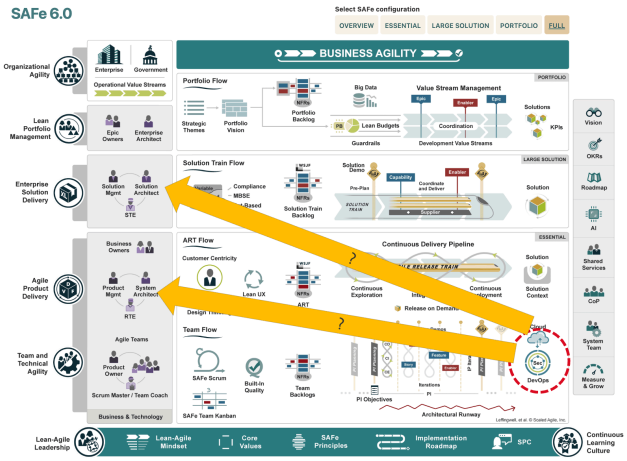


Figure 3 Image Adapted from : https://scaledagileframework.com/#

Quality, Security and Regulatory constraints will likely not be considered since Product Owners, Product Managers and Solution Managers are always focused on the next functional product feature.

While Security is a priority for some, it is often non-trivial to assess the real risks and security posture of the product without expert support inside the DevOps teams. System and Solution Architects can offer limited or no advice on NFR's, since they have no detailed insight into the individual team activities and are often not trained in how to address these topics.

This will result in overlooked consistency on quality topics and constraints. This is due to the solution triangle's primary objective: fulfil *functional* objectives and key results in the eyes of Business Sponsors who measure team success on functional output.

The result can lead to products being delivered with a lack of quality, illegal use of technology (eg, Public Clouds or SaaS Services) or insufficient security measures implemented for the represented use case.

Deleted: ¶
¶

Deleted: a

Commented [BJ4]: While we know its true, I would make this statement less strict. Quality and resilience has been pushed as a business goal frequently (also in PI plannings etc..). The problem is more that people did not know what to do with this. And of course techn. Dept is too big and investment still too small..

Deleted: in the

Defining and Expanding the SAFe Roles.

Swisscom has concluded that an expansion of these roles is necessary. We have defined the following additional roles to supplement and represent Non-Functional Security requirements in our agile framework. We have mapped these roles onto the full SAFe development model for clarity as follows.

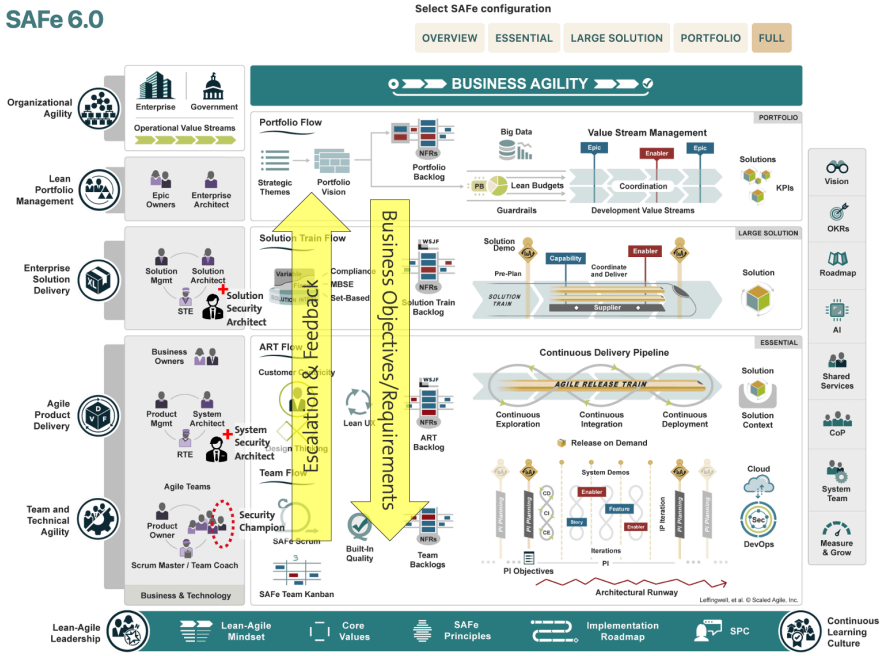
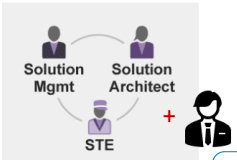


Figure 4: Adapted from https://scaledagileframework.com/

The key objective from this model is that Business Objectives, Requirements and Enterprise risk tolerance reach the teams - and in turn - the teams can provide feedback as to if the objectives are realistically attainable and efficient in implementation.

Solution Security Architect

The Solution Security Architect is an extension to the solution triangle with the focus on Security culture and community and ensuring that NFR's (with regards to security and regulatory primarily) are adequately planned in the value stream.



Solution Security Architect

Figure 5 Image Adapted from https://scaledagileframework.com/#

The Solution Security Architect is a 50-100% Role (dependant upon the size of the Value stream) and is responsibility for visibility of Security and NFR's in the Solution Leadership Triangle.

The Solution Security Architect is responsible for:

Commented [C(5]: Not 100% sure where to put it, but probably at the end of this chapter - after you explained the different roles and their responsibilities:

I miss the big-picture overview. You have multiple pictures with parts of the SAFe model and the communication path. But I think an highlevel overview with the complete SAFe model would be highly beneficial.

Maybe something in the direction of overlaying all roles on the SAFe diagram and show the communication paths on it. Also the feedback culture, providing feedback back up to the Large Solution / Portfolio Level. You had once drawn a picture explaining this.

Commented [C(6R5]: maybe another one showing the two different SCoPs in place and who is in it.

Commented [t(7R5]: I agree on the Model, however "CoP" is covered on the right of the SAFe Model generally, I do not think we should focus on this as it is already in SAFe

Formatted: English (UK)

- **Solution Security Risk Management**: Specifically, ensuring that security risks are represented and treated according to the risk appetite of the business sponsors at the portfolio level, as well as in the Solution triangle. This is performed at regular intervals. We recommend doing this with the solution triangle at least once per PI (Product Increment) *prior* to planning activities for the upcoming PI to ensure that risks are handled *next to* functional requirements.

  When this does not happen, capacity will be reserved for other functional requirements tasks which may lead to incorrect prioritization with bad outcomes for the organization. We must always remember that "business is risk" by its very nature however gauging and measuring that risk tolerance within the organization is critical for the desired business outcomes.

- **Solution Security Community of Practice;** To ensure consistency on the topic of security it is required to build a strong, consistent security culture surrounding the requirements and approach to handling security issues.

  The Security CoP at the Solution level allows System Security Architects (see below) to understand the conditions and focus topics that they need to address within their ART's (Agile Release Train's). This is setup in a "round table" manner where specific topics are not just driven top-down. Rather, security topics such as new 0-Day vulnerabilities, concerns or incidents are discussed - root cause analysis is performed, and activities are planned to remediate any serious findings.

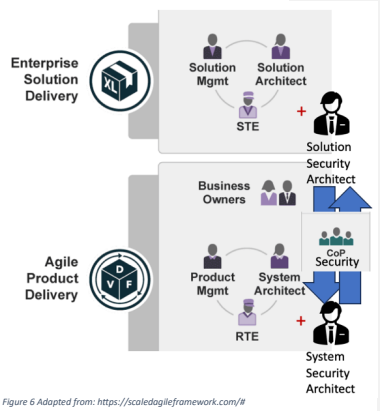


Figure 6 Adapted from: https://scaledagileframework.com/#

  When your organization has a strong Solution Security CoP surrounding security topics you can establish a clear feedback loop from the engineering teams implementing security requirements. This helps you identify inefficiencies, deficiencies or approaches that perhaps do not work as planned in the real world. This in the end results in less wasted effort and better outcomes.

  The Solution CoP is voluntary. It is important that all the members feel that their voice is heard and considered with any decision making or strategy to be undertaken. In the end Security is a People problem, if you ignore the people and their needs and competencies – you will have more problems.

- **Solution Security EPIC creation and Prioritization;** Solution Managers are responsible for planning of the resources of the Solution Train. *Through* being an integral part of the "Solution Triangle" the Solution Security Architect raises and represents topics that address security risks, vulnerabilities as well as new security features that need to be prioritized. By planning non-functional security topics

alongside functional requirements, the organizations risk appetite can be determined and managed.
The side benefit of this is that it also creates transparency towards business leaders about what risks they are accepting. In some cases, if not raised, leadership are blind to regulatory or security requirements which can lead to bad organization or even personal legal liability outcomes. No responsible leader wants this.

In the end the Solution Security Architect is the one with focus on non-functional requirements and drives the solution train in the right direction when handling these NFR's as part of the Solution backlog.

Without one, security will always be an afterthought.

System Security Architect

Analog to the Solution Security Architect, System Security Architects are responsible for coordinating security architecture topics across the Agile Release train. This role is an extension to the ART Triangle.

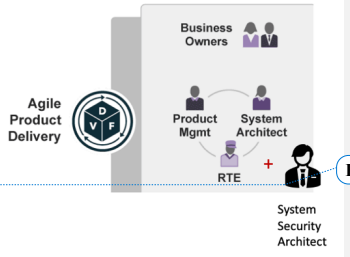


Figure 7 Image Adapted from https://scaledagileframework.com/#

The System Security Architect is a 40-60% Role (dependant upon the size of the ART) and is responsibility for visibility of Security and NFR's in the ART Leadership Triangle.

Formatted: English (UK)

The System Security Architect is responsible for:

- **ART Security Risk Management**; Specifically, ensuring that risks are represented and treated according to the risk appetite of the ART Triangle. Risks are discussed in the ART triangle at regular intervals. We recommend doing this with the ART triangle at least once per PI (Product Increment) *prior* to planning activities for the upcoming PI to ensure that risks are planned next to functional requirements.

  ART's by their nature have a limited capacity for accepting risk usually an order of magnitude lower than at the Solution Level.
  Escalation of risks beyond the ART triangle happens together in the Solution Security CoP to which the System Security Architect is an integral part.

  Like at the Solution level; when Risk Management does not happen, capacity will be reserved for other functional requirements tasks which may lead to incorrect prioritization with bad outcomes for the organization.

- **ART Security Community of Practice;** To ensure consistency on the topic of security it is required to build a strong, consistent security culture surrounding the requirements and approach to handling security issues.

  The Security CoP at the ART level allows System Security Architects to understand the conditions and focus topics that they need to address with their Security Champions

(detailed below). This is setup with a more Top-Down focus however requires feedback from DevOps teams through Security Champions feedback to ensure that measures are implementable and functional at the engineering level. This is where the "rubber hits the road" and products are delivered, so listening to security champions representing the concerns of individual teams is paramount to a successful agile security strategy and feedback loop to determine if the approach is on a successful track.

When your organization has a strong ART System Security CoP surrounding security topics you can establish a clear feedback loop from the individual engineering teams implementing security requirements. This helps you identify inefficiencies, deficiencies or approaches that perhaps do not work as planned (by Architects) in the real world. This in the end results in less wasted effort and better outcomes.

Security Champion

The Security Champion is responsible for "championing" the Security topics at the team level and to raise awareness in the team of any security topics that arise.
The Security Champion is a voluntary position that an existing DevOps engineer (or other suitable technical team member) assumes. Normally to be effective, 20% of a DevOps Engineer's time is required to be reserved for these activities. Please note that this is however highly dependent on the size of the team. The Security Champion with this capacity can support a development team of 5-9 people. For larger teams (not recommended by SAFe) with 10+ people more capacity would need to be reserved. The Security Champion is part of the ART CoP and raises risks that cannot be resolved at the team level to the System Security Architect for treatment at a higher level.

Figure 8: Adapted from https://scaledagileframework.com/

The Security Champion coordinates with other engineering personnel in the team and the Product Owners of the services at the team level for prioritization of security and compliance topics.

It is important to note that the Security Champion does not take responsibility for the security of the product at the team level. This is the Product Owner's responsibility. The Security Champion however is responsible for *transparency* on security topics toward the Product Owner. The activities of the champion are as follows.

Perform Threat Modelling; Threat modelling is integral to the design of good quality services, resolving security deficiencies in the proposed architecture design and ensuring that these deficiencies are placed accordingly in the backlog of activities prior to beginning team level planning activities.

Resolving security issues in the design and plan phases of the DevOps cycle is orders of magnitude cheaper than trying to fix an already running service with a bad architecture. It is akin to fixing an aircraft when it is on the ground, versus while it is flying.

We highly recommend performing threat modelling (we currently use the STRIDE model by Microsoft, however many models exist in the market which may better fulfil your needs) prior to any decisions to begin building a solution.

Threat modelling should also take place whenever new connectivity or changes to the architecture arise. This can take place as an “update” threat modelling session to consider the impact of the new changes on the system. It is the responsibility of the Security champion to represent this as a topic as an engineering team member whenever an architecture change is proposed. This ensures consistency over time with regards to architectural security decision making.

In our organization we also provide a central competence team which can review threat models. Having trained penetration testers as consultants who can help with the quality of threat modelling can be beneficial in ensuring that you are covering the most important topics. Remember a Security Champion is not primarily a security professional, rather a member of the engineering team and as such may miss some things without further expert guidance.

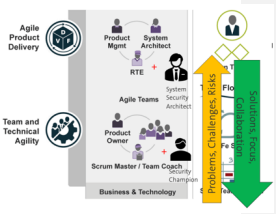
Security Checklists; Most organizations at scale have a designated framework that they follow. Be it ISO 27001, SOGP, COBIT or others the point of these security frameworks is to introduce *consistency* across the approach to security themes.

A checklist for applications with requirements should therefore be performed. This is akin to when you take your car to the mechanic. The mechanic will check that the brakes are in good shape, that the tires do not have uneven or excessive wear and if the oil has been changed to determine if the car is ultimately safe for use.

Security checklists cover the complete list of security requirements to identify any gaps that should be placed on the team backlog for prioritization alongside functional requirements. This is best performed prior to roll-out of a new application to make sure you are not forgetting anything.

We strongly advise creating a checklist for your organization based upon some widely accepted standard. Ours is based upon ISO27001 Annex A controls due to customer requirements, however specific sectors or industries may need to need to consider more specific requirements to conform to regulatory requirements such as GDPR, HIPAA, Telecommunications law as well as the upcoming DORA in the EU which will make more specific requirements for Banking and Insurance sectors.

Security CoP Participation: A security champion is a member of the ART Security Community of Practice. The interaction of the CoP is both top down and bottom up in a collaborative manner.



Security Champions receive instructions about what to do about new vulnerabilities and security initiatives from the System Security Architects. Likewise, System Security Architects receive feedback with regards to the implementation of security requirements.

If requirements are too onerous, this can have a heavy impact on the cost structure of the organization so this continual feedback loop allows us to make better decisions. System Security Architects can then bring this feedback into the Solution CoP round or ART Triangle as necessary.

Leadership Interaction.

A key message here has been consistency.

Consistency in how Risks are reported and escalated across the organization is necessary. This is especially important when escalating risks to the portfolio level or transferring to a different value stream.

It is important we do not get into a situation where we are comparing “apples with oranges” when it comes to risks across value streams.

This consistency with the methodology allows risks to be transferred horizontally between value streams with minimum effort - or escalate risks and share a common understanding of which risks should be prioritized first.

Personally after 25 years of experience, I have never seen a “good” risk management methodology. Only what I would consider “good enough”. Typically, we have the options of Quantitative Risk Analysis and Qualitative Risk Analysis.

Our recommendation would be to focus on comparing one risk to one another using a Qualitative *standardized* Risk Management Methodology.

Only if a financial impact is likely to be very large should one take the time to consider assigning a Quantitative risk analysis result. We use the Monte-Carlo method of risk analysis under these circumstances. Otherwise use the methodology as a means of prioritizing activities against each other and measuring risk appetite, not as providing absolute financial risk scenarios. This will save time and aid you in prioritization and planning.

Other leadership topics

Leadership should be regularly informed about compliance and security topics. Inside our organization we have reporting functions built into our own home-grown checklist application. It is important to be able to make it clear what the status is with regards to Security and Compliance topics at regular intervals with Business Sponsors and not just simply within the value stream.

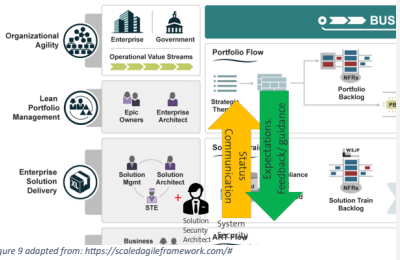


Figure 9 adopted from: https://scaledagileframework.com/#

This approach will change from organization to organization, however, is typically the responsibility of the Solution Security Architect together with the Solution Triangle to represent the status of these topics and take input from the leadership as to progress and risk appetite.

Deleted: apetite

Education

Security Champions are not security professionals. Similarly, Solution Train and ART leadership are often not well versed in their obligations with regards to Security and other Non-Functional requirements.

Solution Security Architects System Security Architects and Security Champions are often not clear where they lie in the SAFe model and what they need to do. After all, where is this discussed in detail in SAFe?

As a result, we have created standardized training programs which address different topics to different audiences:

Deleted: result

Solution/ART Leadership; This is primarily defined to clarify responsibilities and how risk management needs to be considered with regards to NFR's.

Risk Managers; Often areas of an organization will have dedicated risk managers that Security Architects interact with. These Risk Managers need to have a common methodology to be effective across the organization when transferring or escalating risk.

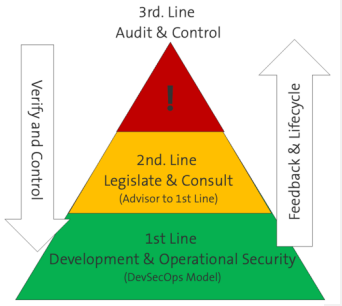
Solution/System Security Architects; This clarifies the details of the responsibilities and how to perform in this role.

Security Champions; This clarifies to DevOps engineers what needs to be considered with regards to security and what the expectations of the role are towards transparency.

DevOps Engineers; This includes all DevOps Engineering personnel with specific education streams for different programming languages such as JAVA, Python or C. This is an integral part in ensuring that Engineers have the information they need to do quality work on a day-to-day basis.

Role of Enterprise Security Organizations

Companies that reach a critical mass staff a central security organization, appoint a CSO,CISO as well as lead Security Officers and Security Architects who oversee the risk posture of the company and propose initiatives at the enterprise level to change.



These anchor, at the enterprise level, security into the fundamental culture of the company and ensure that security related activities are considered at the highest level.

Often these organizations have consulting arms which can provide "security coaching" input to first line DevSecOps organizations as a way to steer the direction of development of security within the value streams of the organization.

We have applied such a model in Swisscom with the "Three lines of Defence" model.

We have focused here specifically on the 1st Line of defence as it represents the broadest and most complex arm of managing security at scale.

The second line is important to steer the overall direction of the company, defines Requirements and identifies compliance topics.

Similarly the Second Line controls that the first line is taking responsibility and managing risk in accordance with the enterprise risk appetite. They make use of the 3rd Line (Audit & Control) to verify that policy is being implemented consistently from top to bottom in the company and to identify any deficiencies in either the policy, or implementation.

We will not delve further into this subject as it goes beyond the subject of DevSecOps. It is important to note however that value stream organisations do not operate in a bubble, rather are part of a larger enterprise security strategy. Coordination and cooperation as well as a common understanding across the different lines of defence are required to be successful.

Commented [C(8]: I also would highlight a bit more that the 2nd line defines Requirements/Compliance topics, which the 1st line needs to fulfill.

Commented [C(9]: I think it would be very beneficial for the readers to also write about how the 3 lines of defense should/could work together.

Most likely readers of this document are working with SAFe and therefore are from larger corporations, which most likely will have a similar org structure as Swisscom.

One of the big challenges as you know is exactly the cooperation between the different departments.

!?! Profit!?!

There are literally hundreds of different tools for identifying, automating and deploying applications into production in a DevSecOps manner. We will not delve into the details of how "we" have chosen to apply the security tooling landscape, as this is an ever evolving approach that requires constant attention and iteration.

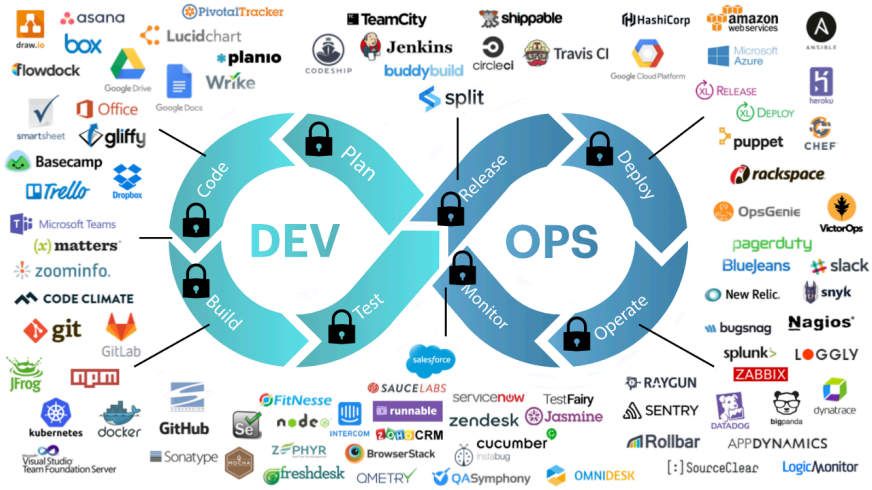


Figure 10 Image adapted from Source: https://www.openxcell.com/devops/

What we would recommend is both simple and complex.

Ensure that at every stage of the DevOps Process Security is considered with the tooling you have chosen. It is essential to understand that no single tool or method resolves all your security challenges. Different tools address different angles, and one might fail or be ineffective for a given threat. Thus, it is crucial to have a multilayered security approach where separate tools and measures complement each other. This principle is called "defence in depth". This will ensure that you are designing and developing secure software, releasing secure software and managing the security lifecycle of software in production.

Credits:

This document would not have been possible without standing on the work of others. As all things DevOps, DevSecOps is an iterative approach that we are consistently refining over time to ensure that our approach is as efficient as possible.

To begin with, thanks to the SAFe framework for providing us a basis for this document. Without it, we would not have a starting point from which to delve into the security aspects of DevSecOps at scale.

Thanks to **Manuel Cianci** of cyllective (cyllective.com) who introduced me to the basis of DevSecOps at the team level. We have come a long way in the years since then, but it was the initial spark that got things moving that was the basis of this document.

Thanks to **Collin Geisser**, Lead Security Architect who provided input and helped us iterate the approach that we have taken here from the early days of DevSecOps at Swisscom.

Thanks to **Jens Birchler**, Lead Security Consultant from Accenture. Your input over the years has been highly valuable in refining the approach. It is greatly appreciated.

Thanks to **Thomas Zeender,** Solution Security Architect who provided both inspiration to publish this document (the cool name S2Fe), and input with fresh eyes to see its value to the wider industry at large.

Deleted: C

Deleted: C

Deleted: h

Deleted: his