Critical Success Factors for Integrating Security into a DevOps Environment

Completed paper

Jacques de Kock
University of Cape Town
DKCJAC009@myuct.ac.za

Jacques Ophoff
Abertay University &
University of Cape Town
j.ophoff@abertay.ac.uk

ABSTRACT

Integrating security into a DevOps environment, also known as DevSecOps, can allow organisations to deliver more secure applications and services faster to market. While many publications address the theoretical benefits and challenges of security integration, there is a lack of practical insight to guide organisations towards a successful integration. As a result, many organisations fail to achieve DevSecOps due to the historical differences that hinder collaboration between teams. This study investigates the critical success factors for DevSecOps integration using a case study approach. Semi-structured interviews were held with eight senior staff members directly involved in establishing DevSecOps integration within a large organisation. Thematic analysis of data across three categories (people, processes, and technology) identified eight major themes: executive support, security champions, security training, way-of-working, governance framework, secure pipeline, automation, and technology. Based on these findings a framework is proposed to inform and guide organisations on DevSecOps integration.

Keywords

DevOps, DevSecOps, security, critical success factors.

INTRODUCTION

In software development an agile approach allows for continuous adaptation of a product by breaking up one overall rollout into a delivery line of smaller projects, each with its own opportunity for growth and change. Larman (2004) describes this dynamic process as a calculated, time boxed, iterative approach whereby software is delivered incrementally, instead of all at once. The limitation of agile is that it was only designed (although it worked very well) for development teams and cannot easily be rolled over into other teams (Turk et al., 2014).

Historically, this did not present a problem due to the separation between development and other teams. However, organisations recognize the need for an approach unifying development and operations (Smeds et al., 2015). This gave rise to the development of 'DevOps' (DO) which is an enhanced framework that allowed agile to be deployed across both development and operations in recognition of their intricate relationship (Jabbari et al., 2016). The DO approach promoted better communication and collaboration between development and operations. Therefore, DO is seen as a gateway for applying agile more broadly.

A third component to the above approach, which is increasingly important, is security. Traditionally, security was only involved very late in the delivery cycle and its importance is only now being fully realized (Storms, 2015). Integrating security into DO is called 'DevSecOps' (DSO), symbolizing the importance of integrating these three components into one unified approach (Farroha & Farroha, 2014). Mohan and Othmane (2016) defined DSO as the incorporation of security practices into DO processes through the promotion of collaboration between the security teams, the operation teams, and the development teams.

Empirical research shows that although organisations successfully adopted DO and leveraged the benefits, most struggle to integrate security successfully. The reason is that by default DO leaves security until the end and fails to include it throughout any of the DO processes (Storms, 2015). MacDonald and Head (2016) reported that as little as 20% of security architects worked with their DO teams with the goal of actively and methodically integrating security into their DO initiatives. Barriers between DO and Security are systemic and the sharing of responsibilities between application teams and security professionals is not always clear (de Vries, 2014).

While theoretical studies of DSO are increasing (e.g., Sánchez-Gordón and Colomo-Palacios, 2020; Rajapakse et al., 2022; Akbar et al., 2022) few have focussed on people and the implementation of this approach (Myrbakken & Colomo-Palacios, 2017; Rajapakse et al., 2022). While both DO and security have limitations which complicate the successful and seamless implementation of DSO present theory fails to adequately explain strategies and methods for successfully integrating DSO in a business environment. This is particularly worrying considering the rise of security threats. The purpose of this research was to answer the research question: *What are the critical success factors to successfully integrate security into a DevOps environment?* In this regard critical success factors (CSFs) are defined as “key areas where things must go right in order to successfully achieve objectives and goals” (Bullen & Rockart, 1981, p. 9). The identification of CSFs is frequently seen in information systems research and can provide a way of “focusing attention on a limited number of areas” (Borman and Janssen, 2013, p. 397).

BACKGROUND

Security verification is usually one of the last steps in DO before product release (Storms, 2015). While traditionally resistant to earlier integration, DO processes can be favourable to the significant improvement of security defences (Wilde et al., 2016). The automated continuous deployment pipeline and iterative checks provide a strong foundation to integrate security properly and frequently. By leveraging from the automated deployment foundation of the DO processes, security vulnerabilities can be significantly reduced by automating security, integrating, scanning, and testing vulnerabilities (Wilde et al., 2016).

However, Storms (2015), reports that organisations remain unsuccessful integrating security successfully into DO. Research conducted by Myrbakken and Colomo-Palacios (2017) revealed similar results which found that organisations struggle to successfully integrate the three components, which negatively affects the speed of delivery in their DSO implementations. The traditional security processes that focus on documentation, manual processes, and out-dated toolsets, are no longer fit for continued deployment environments created by DO (de Vries, 2014). Organisations have difficulty changing cultural behaviours, adopting to process requirement changes, and selecting the right technology to execute security tasks (DeGrandis, 2011; MacDonald & Head, 2017; Vehent, 2017). Wootton & Erkunt (2018) identified three key categories in need of refinement to address the security integration problem: people, processes, and technology. Applying the correct DSO practices in each of these categories could assist organisations to achieve a successful DSO environment.

People

One of the most difficult challenges DSO faces is to change the traditional way security teams integrate with the rest of the organisation. Security and DO teams struggle to collaborate which negatively impacts secure product delivery (DeGrandis, 2011). Developers are functionality-driven to move as fast as possible to deliver new software with new features. Security is driven to move towards the best protected environment (Carter, 2017). A security culture cannot be achieved between these teams unless the people are convinced of its importance in their daily work (Williams, 2009).

A possible approach is to introduce security champions (Beris et al., 2015). Security champions are employees who form part of a product team and assist with the decisions about when to engage with the security team. The duties of the security champion include to ensure security does not block active development, make important decisions quickly and accurately, assist with quality assurance (QA) and testing, write appropriate test cases, stay informed on modern security attacks/defences, and introduce a body of knowledge to the organisation (Becker et al., 2017). The idea is not to make the developers and operation teams' full security specialists, but to raise awareness of an attacker's perspective so that teams can deliver more secure solutions (Carter, 2017).

Processes

Organisations must move beyond an investment in personnel to implement strong processes. By failing to include security teams in fast deployment iterations, organisations allow software vulnerabilities to increase (MacDonald & Head, 2017). DSO attempts to automate core security tasks by integrating security controls and processes into the DO workflow. By standardizing security integration and strengthening security infrastructure with proper procedures, organisations can help ensure the lasting success of adopted measures.

During the planning phase, security teams should perform threat modelling tasks which must be followed by rapid risk assessments (Shackleford, 2016). Risks should be classified and assigned a priority based on their potential impact to the environment and likelihood (Myagmar et al., 2005; Shackleford, 2016). Automated security checks are required to inspect automated code builds by analysing static source code (also known as linting) of infrastructure and application source code. This could assist the organisations in overcoming security integration difficulties. Static source code analysis is a way of finding bugs and quality-related code disputes that later evolve into security problems (Kirchmayr et al., 2016). Containerized environments should be hardened and scanned with the correct security tools to ensure no impact on speed of delivery during the build process (Myrbakken & Colomo-Palacios, 2017).

Security can profit from automation by incorporating security controls and processes that allow for fast and continuous testing. Such tools may include, but are not limited to, automated code review, vulnerability scanning, privilege management, configuration and patch management, automated testing, automation of compliance testing, event monitoring and logging (Rahman & Williams, 2016; MacDonald & Head, 2017; Shackleford, 2016). The execution of automated security acceptance tests combined with functional security tests ensure systems and applications perform as expected. Automated scanning of infrastructure ensures baseline hardening and compliance requirements are met. Application security scanners and automated security acceptance checks ensure security requirements are met (Cox & Kneidinger, 2017).

To ensure continued production stability, security checks need to be executed before, during and after code deployment. This ensures the delivery of correctly built and well tested software, while integrating industry standard security requirements (MacDonald & Head, 2017). Many organisations find difficulty in selecting the correct toolsets and/or struggle to configure toolsets adequately to match their code delivery iterations. The result is that production software is not protected against latest known vulnerabilities (Wootton & Erkunt, 2018; Yasar, 2017; Zychowicz, 2015). With the correct toolsets and configuration, automated provisioning and deployments can be utilised to expedite delivery of software and ensure security consistency in the software release process. Properties can be audited, and secure configurations verified across all systems and services (Myrbakken & Colomo-Palacios, 2017; MacDonald & Head, 2016). Post-deployment testing collects application-level security metrics to help identify malicious patterns. Strong continuous monitoring and alerting platforms ensure that changes to the production environment do not introduce new security risks (Zychowicz, 2015).

Technology

Technology is the third key category for successful implementation of a DSO environment. Technology enables the people to execute the processes of the DSO methodology. Selecting the best technology for the matching process, implemented by the correct team is of major importance (Wootton & Erkunt, 2018). Literature shows that many organisations fail to match the technological capabilities with the DO functions to enable seamless security verification. Product code created by the development team must be checked constantly against repositories containing the latest vulnerabilities, threats and compliance requirements (MacDonald & Head, 2017; Wilde et al., 2016). Additionally, the technology could scan the code for known vulnerabilities, threats or compliance inconsistencies during live coding practises and provide live updates of possible disputes (Wootton & Erkunt, 2018).

The ability to successfully combine automation and orchestration with the use of the correct technology is a key success factor in the DSO world (de Vries, 2014; Rahman & Williams, 2016). Together, orchestration and automation make use of metadata to provide a baseline of secure infrastructure, repeatable processes, and default product code. This increases the auditing of the environments (Yasar, 2017; Zychowicz, 2015). A strong configuration management toolset uses predefined security and configuration templates to apply over these environments and report on any deviations (Wootton & Erkunt, 2018).

Knowing what is happening in the environment post-product deployment phase is as important as the deployment design and implementation phases (Carter, 2017). Technologies should be selected to monitor security in a fast DSO environment and log events rapidly. The data should then be analysed in real-time to provide the security professionals with the ability to gain knowledge of environmental behaviours and produce automated security auditing results (Rahman & Williams, 2016; MacDonald & Head, 2017; Shackleford, 2016).

Applying the correct DSO practices in each of these categories can assist organisations in achieving successful DSO. The people category contains recommendations for organisational culture change, security awareness and training. The processes category includes specific recommendations for each individual step of the DSO planning stages. Using the correct technologies enables people to execute the processes of the DSO methodology. Selecting the correct toolsets for automation, for example, is instrumental to enabling security to keep up with the DO pipeline.

RESEARCH METHODOLOGY

Using an interpretivist approach, we conducted a case study to understand the decisions, attitudes, and impressions of decision-makers that drive the organisation’s DSO approach. This approach was chosen due to the immaturity of DSO and the lack of published literature to assist organisations with implementation. We aimed to uncover factors important to the organisation which explain how it addressed concerns and approached hurdles in the DSO implementation.

Case study research is a way to observe a phenomenon in its natural setting. A case study employs multiple methods of data collection to gather information about the phenomenon of interest. The leading characteristics of case study research are that it is narrowly focused, provide high levels of data, and combine both objective and subjective data to achieve an in-depth understanding (Yin, 2017).

We conducted a descriptive case study, as the emphasis was on describing how the organisation pursued their DSO journey. Descriptive research studies can yield rich data that lead to important recommendations (Corbin & Strauss, 2008). While empirical research has shown that many organisations fail in effectively implementing DSO, the case organisation manged to overcome similar barriers. Other organisations can gain valuable insight from this and apply similar strategies to reach their DSO goals.

The Case Organisation

The case organisation has a large presence in the financial services sector, with millions of customers. The organisation transmits customer credit card information through digital infrastructure strictly governed by the Payment Card Industry Data Security Standard (PCI DSS) and must comply with the PCI DDS standard (Wilson et al., 2018). Deadlines to achieve full PCI DDS compliance across their digital landscape caused the organisation to realise that they had to fast track resolving challenges.

The organisation had tried to overcome challenges with integrating security into DO sooner, but never really prioritised the approach. With the rapid growth of the organisation as an entity the security team expanded exponentially in a very short time. However, as DO and security teams were working independently, management realised that something had to be done to maintain industry momentum while avoiding introducing new risks into their environment.

Data Collection and Analysis

We identified participants from various groups to provide a true representation of the current DSO implementation. The selection of these participants was based on their decision-making roles and in-depth knowledge of DO and security. Participants were chosen from all levels of management to provide a full spectrum of feedback. Due to their seniority, high-level decision makers were seen as the voice for many employees and sub teams below them.

Semi-structured interviews were conducted with eight participants. The participants represented the following stakeholder teams, with participant identifiers indicated in parentheses:

- Enterprise Architecture (1 interview; EA1)
- Infrastructure Architecture (1 interview; IA1)
- Development (2 interviews; D1, D2)
- Security (3 interviews; S1, S2, S3)
- Operations (1 interview; O1)

Interviews were conducted face-to-face using the instrument in Appendix A. Our review of existing literature revealed three key categories when transitioning to a DSO environment: people, processes, and technology. Although interview questions focused on these three categories there was allowance for other topics to emerge from the dialogue between the researcher and participant. Interviews were focussed on the area of expertise of each participant to understand their individual environments and collaboration with other teams. During the interviews several participants pointed to key policies and guiding documentation that were instrumental to the organisation's transition to DSO. These documents were reviewed and incorporated as a secondary source of data. All interviews were recorded and transcribed for analysis. Thematic analysis (Braun & Clarke, 2006) was used to analyse the interview data and identify CSFs within and across the three key categories.

RESEARCH FINDINGS – CRITICAL SUCCESS FACTORS

Background – The Environment prior to DSO

Before embarking on its DSO journey the organisation used a waterfall development approach, which separated the development, operations, and security teams. The appointment of a new CIO, committed to converting the organisation to an agile methodology, evolved into the organisation's DO practise. Although the organisation managed a successful transition to DO, they continued to struggle with integrating security. This was a common predicament highlighted by research (e.g., de Vries, 2014; Storms, 2015; Vehent, 2017).

The DO teams' worldview differed so significantly from the security team that they found it nearly impossible to collaborate. DO was functionality driven while security's default goal was to protect the environment against the unknown. These differences created significant barriers to collaboration (e.g., Carter, 2017; Storms, 2015; Myrbakken & Colomo-Palacios, 2017; de Vries, 2014; Erich et al., 2014). During the interviews, participants were eager to share the challenges their teams faced and provide context regarding the frustrations experienced between the DO and security teams. Many of these challenges have already been identified in the literature review, indicating that none of these frustrations were unique to this organisation.

Most of the highlighted frustrations stemmed from the fact that the security team was grossly understaffed. As a small team with only a few members, they couldn't keep up with the demand of business applications being created. Participant S1 explained that *"…five years ago there was* *one [person] in the governance risk and compliance side of things…"* and S2 added that *"...a couple of years ago… we were probably five people on the security team driving security practices."* This led to frustrations on both the security side (feeling overwhelmed and overcommitted) and on the developer side (feeling held back by these limited resources). As security was such a small team, S3 observed *"…there was not a lot of technical breath in terms of knowledge and skills…"* The development teams were way ahead of the security teams regarding coding standards and terminology. The disparity between technical capabilities and training made it even more difficult for teams to collaborate.

There was also no mechanism in place for development and security teams to collaborate during the early design phases of a project. D2 described a feeling of isolation and technical disconnect from the security team resulting in "*no close collaboration during our sprints or planning sessions at all."* This was also ultimately a result of the security team being understaffed as frankly, as per participant S2, *"...with our resources it was impossible to get there in time for design phase…"*

Another point frequently cited in published literature is that manual security processes will slow down software delivery (Myrbakken & Colomo-Palacios, 2017). S1 elaborated on the frustrations caused by the manual processes used at the time *"...our speed of delivery was extremely slow and a lot of time our department was seen as the bottleneck…"* and the result of that was *"...frustration between departments and it wasn't good for morale…"* Asking participants why they thought this struggle felt so insurmountable, S1 explained that before DSO their team saw"…*with agile and DevOps that it, to a certain extent, created siloes between departments…"* Agile and DO, they continued, are great software development practises "…*but in the infrastructure, compliance and monitoring space agile and DevOps do not fit well."*

These are just some examples of the challenges this organisation's DO and security teams faced before the integration of DSO. The CIO was worried about security vulnerabilities, the increasing demand for secure applications, and wanted to remove the frustration of security involvement very late in the development process (e.g., Heltzel, 2018; Shankar, 2016; Erich et al., 2014). Motivated by the urgency of a hard deadline to become fully PCI DSS compliant the CIO began a transformational push to DSO. The following sections present the CSFs which emerged out of the interviews, explaining the approach this organisation took to overcome their challenges.

CSF 1 - Executive Support

Realising that PCI DDS compliance could not be reached without resolving the historic challenges between DO and security, management enabled a top-down transformation throughout these two teams. This added level of executive endorsement allowed measures to be pushed through and resources to be allocated which otherwise would never have gotten off the ground.

Determined to remove the challenges of integrating security into DO, top management decided to appoint a new head of the security team department. S3 was particularly supportive of the new appointment: "*With the appointment of our new head of department, he wanted to have a change towards more collaboration…"* The new department head made some changes to the department and allocated more resources. The additional team members injected new energy into the team. There was a general sense of excitement from the security participants about this change in the team. S1 said that the "*…the new blood [management support] had a different view…"* and with the *"…funding of the department and initiatives…"* provided security with a strong foundation of confidence to move ahead with the DSO implementation.

In parallel with reformatting the security team, executives drove multiple security awareness and ownership campaigns. It was important for the teams to understand the security component in one's daily work. The literature review highlighted that such campaigns are important to support security culture across DO and security. Members need to be convinced of the importance of security in their daily work (Williams, 2009). S3 said that executives really lead by example and showed members of all teams the importance of having a security culture and explained that *"This idea of ownership has filtered down from the top and it has been rehashed fairly often with internal campaigns...ownership in the work that you do and pride in the work that you do…that has helped us a lot. We have certainly leveraged off that."* The body language from the participants showed their level of accomplishment for creating a DSO security culture of teamwork and ownership. Multiple participants made it very clear that the success of DSO would not have been achieved without executive support:

> "*If this kind of change is not initiated from the top, it is not something you are going to drive from the bottom necessarily. Yes, you need buy-in from the bottom, but it needs to be driven from exec level. That is exactly what has happened."* [S2]
>
> *"If we didn't have buy-in from right from the top of the organisation, we wouldn't have been successful."* [O1]

Although there was clearly strong support throughout teams for the importance of security and the collaboration with DO, this study discovered that while successful implementation of DSO has been achieved, different senior members had different ideas and opinions as to the future of DSO. While some are comfortable to continue with the current model, others are looking to completely redesign and move towards a platform-based DSO architecture. This misalignment in the future vision of DSO came across as a clear divide along management lines, with more experienced individuals taking a more visionary approach.

CSF 2 - Security Champions

Security champions are employees on a product team whose responsibilities are to engage with the security teams to ensure that security does not actively block current development, assist with quality assurance and are up to date with modern day attack and defence strategies (Becker et al., 2017; Beris et al., 2015). While this model is recommended in current literature, this study discovered that this approach was not actually suitable for this organisation. Instead, they found it much more useful to base the position within the security team. What literature refers to as the security champion, the organisation refers to as the Information Security Officer (ISO). The role of an ISO is to meet with product owners and product developers during new product design. Together they need to understand the business requirements and the security components which should be factored into the design of the new product.

This contradiction with literature is a result of the organisation's belief that their product teams (made up of product owners and DO) should focus on business functionality, driving the organisation forward and enhancing client expectations. The security components were more effective when driven through an external ISO, working *with* the product teams to understand the product's functional components, provide guidance on security standards and document actionable tasks at agreeable intervals. This approach grew out of the new head of the security department's executive vision of *"...putting security back in the hands of the developer…"* (S3).

The ISO's meet with the product owners and DO teams early in the planning phases and have discussions on the deliverables and the possible security touchpoints. The ISO's provide guidance and receive business requirements which are used to create test cases to compare with current security standards. Participant S2 received positive feedback from line managers since assigning ISO's to teams *"...the guys [DO teams] are happy to see us, yes these guys [security teams] can provide value, and it's awesome…"* ISO1 was assigned to a development team and "*…they're bringing him [ISO1] into all these meetings, they love it… and he actually helped on all their projects that are still in design phase, which is awesome…"* S3 added that the ISO's in the cyber team "…*have been consulting with them [DO] and trying to educate them and involving us earlier, [which] will result in less missed deadlines and less cost…"* Configuring security champions as outside resources was a significant advantage in this organisation's transition to DSO. This likely gave product teams a greater sense of control as the resource was framed as a way to help them, lowering the chances of it being seen as threat or imposition on their realm of expertise. It also may have reduced conflicts and power struggles that threaten progress within other organisations trying to integrate these diverse roles into one cohesive unit.

CSF 3 - Security Training

According to participant S2, before DSO the urgency of reaching PCI DDS compliancy across the digital landscape, prioritised specialised training for the different teams *"...we had to roll out security development training for PCI. To all developers, analysts, marketing, testers, and the whole range of them…"* Understanding all the aspects that needed to be covered to comprehend security requirements for the new compliance was very difficult. The organisation approached an external vendor specialising in customised training material. S2 was very happy with the standard of this training vendor *"...we found this company in Canada...they have extensive training, it's really good"* S3 expressed similar satisfaction from the same training vendor "*they actually do threat modelling and stuff and they talk a lot about leveraging your security team* *and bringing them earlier into your design phase as opposed to right at the end of your deliverable."*

Once purchased, security training material was uploaded onto the organisation's internally available Learning Management System (LMS), managed by the Learning and Development (LnD) department. Training was assigned to the different teams to be accessed at a time of their leisure. Participant D1 noticed the advantages to his team of having access to this specialised training: *"...specific security training to make people aware of what security testers look at…it was a very exciting course…which showed all the cool hacking tricks and I think that created awareness, excitement for the discipline and just to make sure it is part of your daily implementation."* Access to the online specialised security training material helped development teams start thinking out of the box which allowed them to ultimately release better code. Five participants agreed that the quality of the training was exceptional and provided the teams with the skills to enable this organisation to implement DSO successfully. This underscores the importance of organisational investment in not only the right type of training, but in making the training easily accessible and implementing it in a way most likely to be well-received by current staff.

CSF 4 - Way-of-Working

This investigation found that one of the organisation's requirements for the DSO implementation, was for all security teams to have full visibility of projects, deliverables, and tasks within DO at any given time. The executives then introduced what is now known as the Team Cadence Cycle (TCC). Using agile sprint techniques, the TCC allows visibility across DO and security of the projects in planned phase both in-flight and resolved. This custom model provided leverage for the organisation to force early project engagement, create accurate agile stories and synchronise work deliverables. Agile stories are used to break down the work needed to deliver the final product into more manageable chunks. Each story is estimated and sized based on days or story points. The work is then prioritised within teams based on the estimates. Work is also prioritised between teams should any story of one team depend on work to be actioned by a different team (Abrahamsson et al., 2017; Rockefeller et al., 2017). The TCC was created for the DSO initiative to provide project workload insight across DO and security.

This model created a team collaborative foundation by allowing teams to voice their input, negotiate combined efforts and gain forward velocity on more secure products faster to market. S1 said that the TCC helped to "*…get everyone focussed on a common deliverable and not having everybody running in different directions…"* D1 appreciated the visibility across both DO and security *"...more teams are on agile and using the same tools to track stories during planning…visible stories on boards and backlogs that indicate that this is a security related [task]…"* and added that it was positive for the team "*…because you can only manage what you can monitor."*

The literature review found that during the innovate and plan processes, teams should get together, harness new ideas and security should move away from enforcing historical processes. During this time product specific information should obtained, DSO and security champions should use this opportunity to create modelling tasks, discuss risks and prioritise tasks (Myagmar et al., 2005, Shackleford, 2016). The TCC is in direct support of literature and provided extensive value towards the DSO implementation in this organisation.

CSF 5 - Governance Framework

Before DSO, the environment had very relaxed and basic controls in place regarding the access DO members had to systems and controlling what gets released into the production systems. S2 explained how DO would deliberately *"…bypass us in some way…"* to release product into production without security checks *"…because they know coming to us is going to be a problem…"* When security was asked to get involved, vulnerabilities would be identified, and the release of the product was halted until these vulnerabilities were resolved. DO' focus was just to release new functionality, not with security.

To achieve DSO success, security realised the importance of implementing a framework to drive policies and standards across DO and security. Thereby regaining control of security as an entity and providing a secure foundation for all business systems to build from. The literature review shows that DSO attempts to automate core security tasks by integrating security controls and processes into the DO workflow. By standardising security controls and strengthening security infrastructure with proper procedures, organisations can ensure the lasting success of adopted measures (MacDonald & Head, 2017).

Organisations with IT systems cannot just expect those systems to deliver strategic value without certain controls. A mechanism needs to be in place to regulate, monitor and govern the value creation efforts of the IT systems (Calder & Moir, 2009). According to Wiedemann (2018), IT governance can also be seen as a framework of mechanisms for decision-making structures, process alignment and multiple approaches of communication channels. The organisation decided to use the Control Objectives for Information and Related Technologies (COBIT) and the Information Security Forum (ISF) Standard of Good Practice for Information Security, as input principles which later became the organisation's DSO Governance Framework (DGF).

S2 explained how they decided on the standards to use and build a custom framework which suites the organisation: *“we took the root control behind it [COBIT and ISF] … and we created our DevSecOps Governance Framework, and that’s its official title.”* The COBIT framework is an internationally recognised good-practise framework which defines a set of generic processes for IT management through best practise governance and control processes (Huygh et al., 2018). ISF Standard of Good Practice for Information Security is a member-based organisation that provides thorough sets of controls and guidance on current and emerging information security topics. These control statements help organisations to respond to ever increasing threats and risks (McIlwraith, 2016).

S3 explains that the DGF is part of planning discussions and it assisted the security team with building governance into the infrastructure of DO *“…we worked very closely with the integration team…and we determine what governance structures or what governance policies we need to put in place…”* IA1 provided some examples of how the DGF guided the infrastructure team by stipulating *“…the governance and controls around what code gets approved…”* and which *“…authentication and authorization models we need to use…”*

The DGF was a key contributor to the success of the organisation’s DSO implementation. As recommended by MacDonald and Head (2017), the DGF ensured the application of industry standards and strong security controls through strengthened procedures.

CSF 6 - Secure Pipeline

Before DSO, the Operations team considered all the processes and decision-making gates within the pipeline (also known as the CI/CD toolchain). The team then realised that the quality of the pipeline, at that time, was not sufficient to be used for the DSO implementation. As per participant S1 *"initially when we started with the DevOps pipeline, governance and inline checking and security testing wasn't really a thing…"* The new version of the pipeline was built under the DGF and created three release environments. Products or code moving through the secure pipeline will always either be in development phase (DEV), preparation phase (QA) or released to production (PROD). The controls enforced by the DGF were immediately noticeable to the teams consuming the pipeline as highlighted by participant IA1 *"…development environment only has access to deploy to the development environment…QA will only have access to deploy to the QA environment… [the pipeline] is setup with that role-based authentication control included…"*

To successfully apply a segregation of duty model for DSO and to control access of identities to the different pipeline environments, the organisation implemented an Identity and Access Management (IAM) system. Widely recommended in the published literature, an IAM system allowed for the application of a Role Based Access Control (RBAC) model (MacDonald & Head 2017). DO and security collaborated to create security roles and responsibilities and applied the RBAC to the environment, through the IAM system and in line with the controls stipulated by the DGF. The IAM allows the security team to report on all access related activities at any given time of the DSO model, as required by legislation and compliance of the organisation's industry (Yasar, 2017).

CSF 7 - Automation

Multiple participants expressed that automation was another key component to DSO success. They believe DSO implementation would not have been possible at all if it wasn't for the ability of the teams to implement automation. S1 said that *"…automation brings repeatability, consistency and speed of delivery…"* S3 added that automation allowed tasks to be more efficient *"...in how things are being done, because it is done in such a fast pace that you can't have manual steps in the process."* As widely recognized in the literature, shifting security left is a key concept of DSO. The experience of this organisations supports this notion, as automation was an instrumental step in their DSO journey.

In support of this shift left and enabling developers, participants D1, D2 and S2 explained how security used automation capabilities with new technologies to allow the development teams to code against live vulnerability checks. S2 elaborated on the function of the new tool *"...which is a static code analysis tool which picks up code..."* Another automated ability that was built into the pipeline and relates to the release environments is automatic source code checking. S1 said that *"...as the guys deploy to QA or DEV or PROD, they can have their code checked again by a source code analyser so that they can find out if they maybe made issues. Again, it is all automated..."*

Automation provided security with the ability to segregate validation between the pipeline and the IAM. This allowed security to automatically test the RBAC model implemented within applications for any misconfiguration of the security segregation. The security participants expressed the criticality of segregation and that roles within application should only access and action what the RBAC enforces – anything else is a breach. These tests are now all automated whereas previously they were manually executed. This was huge frustration at the time, as per participant D1 *"...any automated process coming to a manual step in the process always becomes a bottleneck..."*

Monitoring applications is just as important as scanning for vulnerabilities during code creations and as code cycles through the release environments. Participants explained how the automation capabilities provided the mechanisms to automate monitoring toolsets and enhance monitoring processes. The automation-enhanced monitoring capabilities provided up to date and accurate vulnerability levels of the entire estate. Additionally, it provided auto response and self-healing functions. S3 explained that the team can now *"…detect a cyber-breach or a cyber-incident."* The information is captured in a single location where they *"…initiate cyber response processes…identify, detect, detain and eradicate that breach."* O1 pointed out that it enabled them to measure the outcomes and tie results back to business value *"…we [are] getting better throughput…having less incidents with these more changes… having less audit findings…"* S1 was excited to add that through enhanced monitoring capabilities, the organisation *"…can today safely say that security is entranced into the DevOps pipeline…"*

CSF 8 – Technology and Processes Fit for Purpose

Technology enables the *people* to execute the *processes* of the DSO methodology (Wootton & Erkunt, 2018). The literature review found that the inability of organisations to select the correct technology to match DO and security processes, contributes to failure of achieving DSO in those companies. Before DSO, this organisation had too many technologies with overlapping functionalities. It was difficult to keep people upskilled to support all the technologies. EA1 believes this is symptomatic of IT culture *"…we are fascinated by technology; technologists tend to be fascinated by technology."* With so many different technologies, participant D1 was often asked by the team *"…which technology components and libraries are we going to use?"* The ultimate negative impact of this problem was summarised by S1 *"How do you do enterprise architecture if you have three products doing the same thing!"* According to the participants, executives and teams soon realised that to support all the visions and processes for DSO, that it was crucial that the correct technologies were selected to enable the people to execute the new processes.

PROPOSED FRAMEWORK

The CSFs discussed in this study show key areas that were identified, prioritised, and resolved to integrate DSO into the organisation's environment. In Figure 1 we propose a framework which classifies these factors into the three key areas of people, processes, and technology.

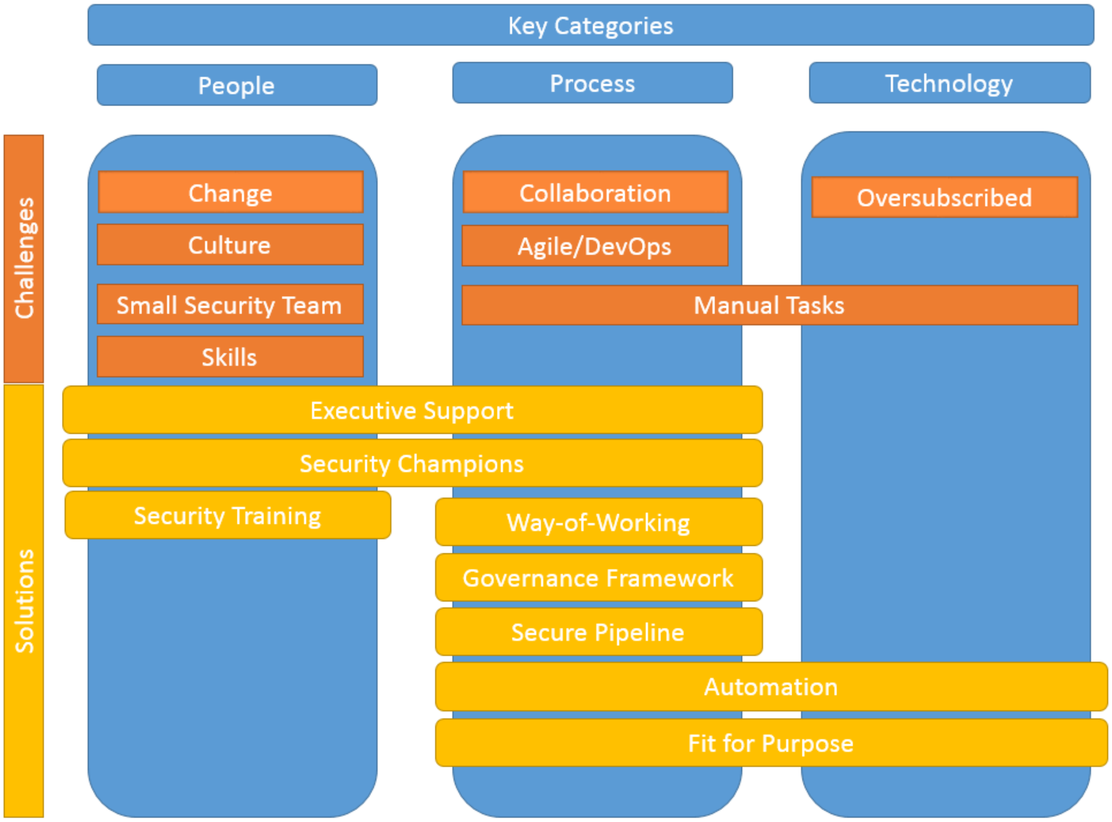

Figure 1. Proposed framework of CSFs for DSO

This framework is specifically targeted to help resolve companies' most common challenges. The framework aims to present a custom, yet repeatable approach the organisation applied to overcome similar challenges. Problem areas were categorised alongside solutions to suggest a new ideology where new processes can be applied to the environment to remediate the challenges. This framework encapsulates the people, processes and technology that ultimately led to the success of DSO for this organisation.

CONCLUSION

The CSFs identified in this research have been classified according to three key categories identified in literature: people, processes, and technology. Within the people category the study found that the success in this organisation started with executive buy-in to a new security culture, driving awareness to every individual. Security visibility increased and security collaborative discussions became easier between upskilled teams. Within the processes category pointed to collaborative team sessions and the creation of a governance model which was the foundation of a secure pipeline built with automated functionalities. Lastly, the technology category shows the importance of selecting the correct toolsets to support the processes, thereby enabling the people.

REFERENCES

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2017). Agile software development methods: Review and analysis. *arXiv preprint arXiv:1709.08439*.

Akbar, M. A., Smolander, K., Mahmood, S., & Alsanad, A. (2022). Toward successful DevSecOps in software development organizations: A decision-making framework. *Information and Software Technology*, 147, 106894.

Becker, I., Parkin, S., & Sasse, M. A. (2017). Finding Security Champions in Blends of Organisational Culture. *Proc. USEC*, *11*.

Beris, O., Beautement, A., & Sasse, M. A. (2015). Employee Rule Breakers, Excuse Makers and Security Champions: Mapping the risk perceptions and emotions that drive security behaviors. In *Proceedings of the 2015 New Security Paradigms Workshop* (pp. 73-84). ACM.

Borman, M., & Janssen, M. (2013). Reconciling two approaches to critical success factors: The case of shared services in the public sector. *International Journal of Information Management*, 33(2), 390–400.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative research in psychology*, *3*(2), 77-101.

Bullen, C. V., & Rockart, J. F. (1981). A primer on critical success factors (Working Papers No. 1220-81. Report (Alfred P. Sloan School of Management. Center for Information Systems Research); no. 69.). Massachusetts Institute of Technology (MIT), Sloan School of Management.

Calder, A., & Moir, S. (2009). Foreword. In *IT Governance: Implementing Frameworks and Standards for the Corporate Governance of IT* (pp. V-Vi). Ely, Cambridgeshire: IT Governance Publishing.

Carter, K. (2017). Francois Raynaud on DevSecOps. *IEEE Software*, *34*(5), 93-96.

Corbin, J., & Strauss, A. (2008). Basics of qualitative research: Techniques and procedures for developing grounded theory.

Crabtree, B. F., & Miller, W. L. (Eds.). (1999). *Doing qualitative research*. Sage publications.

Creswell, J. W., & Tashakkori, A. (2007). Differing perspectives on mixed methods research.

de Vries, S. (2014). Continuous Security Testing In a DevOps World. OWASP AppSec Europe, (Cambridge, UK).

DeGrandis, D. (2011). DevOps: So you say you want a revolution?. *Cutter IT Journal*, *24*(8), 34.

Heltzel, P. (2018). *The 12 biggest issues IT faces today*. https://www.cio.com/article/3245772/it-strategy/the-12-biggest-issues-it-faces-today.html

Huygh, T., De Haes, S., Joshi, A., & Van Grembergen, W. (2018). Answering key global IT management concerns through IT governance and management processes: A COBIT 5 View. In *Proceedings of the 51st Hawaii International Conference on System Sciences*.

Jabbari, R., bin Ali, N., Petersen, K., & Tanveer, B. (2016). What is DevOps?: A Systematic Mapping Study on Definitions and Practices. In Proceedings of the Scientific Workshop Proceedings of XP2016 (p. 12). ACM.

Kirchmayr, W., Moser, M., Nocke, L., Pichler, J., & Tober, R. (2016). Integration of static and dynamic code analysis for understanding legacy source code. In *Software Maintenance and Evolution (ICSME), 2016 IEEE International Conference on* (pp. 543-552). IEEE.

MacDonald, N., & Head, I. (2017). *10 Things to Get Right for Successful DevSecOps.* https://www.gartner.com/doc/3811369/-things-right-successful-devsecops

Mohan, V., & Othmane, L. B. (2016). SecDevOps: is it a marketing buzzword?-mapping research on security in DevOps. In Availability, Reliability and Security (ARES), 2016 11th International Conference on (pp. 542-547). IEEE.

Myagmar, S., Lee, A. J., & Yurcik, W. (2005). Threat modelling as a basis for security requirements. In *Symposium on requirements engineering for information security (SREIS)*(Vol. 2005, pp. 1-8).

Myrbakken, H., & Colomo-Palacios, R. (2017). DevSecOps: A Multivocal Literature Review. In International Conference on Software Process Improvement and Capability Determination (pp. 17-29). Springer, Cham.

Rahman, A. A. U., & Williams, L. (2016). Software security in DevOps: synthesizing practitioners' perceptions and practices. In *Continuous Software Evolution and Delivery (CSED), IEEE/ACM International Workshop on* (pp. 70-76). IEEE.

Rajapakse, R. N., Zahedi, M., Babar, M. A., & Shen, H. (2022). Challenges and solutions when adopting DevSecOps: A systematic review. *Information and Software Technology*, 141, 106700.

Sánchez-Gordón, M., & Colomo-Palacios, R. (2020). Security as Culture: A Systematic Literature Review of DevSecOps. In *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*, 266–269.

Shackleford, D. (2016). A DevSecOps Playbook. SANS Institute InfoSec Reading Room. GIAC.

Vehent, J. (2017). Test Driven Security in Continuous Integration. https://www.usenix.org/conference/enigma2017/conference-program/presentation/vehent

Wiedemann, A. (2018). IT Governance Mechanisms for DevOps Teams-How Incumbent Companies Achieve Competitive Advantages. In *Proceedings of the 51st Hawaii International Conference on System Sciences*.

Wilde, N., Eddy, B., Patel, K., Cooper, N., Gamboa, V., Mishra, B., & Shah, K. (2016). Security for DevOps Deployment Processes: Defences, Risks, Research Directions. *International Journal of Software Engineering & Applications (IJSEA)*, *7*(6).

Williams, P. A. (2009). What does security culture look like for small organizations?.

Yasar, H. (2017). Implementing Secure DevOps assessment for highly regulated environments. In *Proceedings of the 12th International Conference on Availability, Reliability and Security* (p. 70). ACM.

Yin, R. K. (2017). *Case study research and applications: Design and methods*. Sage publications.

Zychowicz, J. L. (2015). The Great Orchestrator: Arthur Judson & American Arts Management. *ARSC Journal*, *46*(2), 331-333.

APPENDIX A. SEMI-STRUCTURED INTERVIEW QUESTIONS

General

1. How long have you been involved with DevOps or Security (depending on the resource being interviewed)?
2. In your opinion, what value does your environment contribute to the organisation's competitive advantage in its industry?
3. At what stage did your team realise that security need to be incorporated sooner into the DevOps iterations?
4. How did the DevSecOps journey impact or change current policies in your team?

People

5. Historically DevOps and security have been working as separate teams. What were your concerns regarding this new engagement and how did you address those concerns?
6. Literature suggests the use of security champions to facilitate the integration between DevOps and security teams. How was the facilitation managed from your team?
7. Was there any form of security awareness training and how was it conveyed?
8. Was the collaboration between DevOps and security a major cultural change for your team?
9. Was the team involvement incentivised in any way or how did you manage buy-in from all the members?

Processes

10. Integrating security into DevOps iterations start at the innovation and planning phases. How did this impact your current processes?
11. Literature refers to 'secure by design' and 'shift security left' whereby security principles are applied during the code build process. However the biggest unknown amongst organisations is the possible negative impact this could have on the fast DevOps build process. How was this concern addressed?
12. Functional and security compliance tests are combined where previously executed separately. Which processes had to be changed to ensure both testing domains are achieved at speed?
13. A DevSecOps milestone is to rapidly release zero-defect code. The release phase is the last verification check before secure code is promoted into a stable production environment. How did this impact your continuous integration/continuous development processes?
14. With security being integrated into each step of the continuous integration/continuous development pipeline, monitoring output and customer satisfaction needs to feedback to the beginning to enhance the next iteration. How did you achieve this circular behaviour to ensure continuously enhancing your environment and increasing customer satisfaction?

Technology

15. Literature shows that many organisations fail to match the technological capabilities with the DevOps functions to enable seamless security verification. How did your team overcome this difficulty?
16. What impact did the DevSecOps journey have on your technology portfolio?
17. Automation and orchestrations is a major component of DevOps. Adding security into these components introduces other difficulties. How did your team overcome these complexities to achieve success?

Value

18. How did the organisation measure the success factors of implementing DevSecOps?
19. What is the future (in your opinion) for DevSecOps in this organisation?