Universiteit Leiden

ICT in Business and the Public Sector

Considerations for the Adoption of a Secure Software Development Framework

Name: Alex Wee
Student Number: S3020908

Date: 21/10/2022
1st supervisor: Asst. Prof. Olga Gadyatskaya
2nd supervisor: Prof. Joost Visser

MASTER'S THESIS

Leiden Institute of Advanced Computer Science
(LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

Background
In response to increasing cyber threats, organizations running IT projects are faced with the question of what measures they can take to strengthen their security posture. One focus area that organizations need to decide on, specific to software security, is which security standard, secure software development framework or activities they need to carry out in order to develop secure software.

Objective
This research looks into a few formal secure software development lifecycle (SSDLC) frameworks and the secure development activities that are prescribed within. It then aims to ascertain what the current practices are with respect to the adoption of these frameworks and the factors that organizations consider when deciding upon which framework/practices to adopt.

Methods
Besides looking into existing literature on secure software activities, frameworks and adoption trends, a survey and interviews were also conducted with IT professionals working in software development projects in order to understand the current software security practices and reasons for their adoption.

Results
This research highlights that despite the many formal SSDLC frameworks available, many organizations and their projects are still customizing their own frameworks or only selecting a few of the activities prescribed, most possibly due to competing demands on finite resources.

Conclusion
To better encourage the utilization of formal SSDLC frameworks, there is a need to simplify them such that they are easier to adopt and less resource-intensive to implement. This is in agreement with prior studies that have observed lightweight security interventions to be more well-received.

Acknowledgement

I would like to express my sincere gratitude to the people who made this thesis possible.

First, I would like to thank my first supervisor, Olga Gadyatskaya, for her close guidance and valuable advice during the entire research process. I am very grateful for her relevant and constructive feedback that helped me considerably in putting this thesis together.

Next, I would also like to thank my second supervisor, Joost Visser, for sharing his unique perspectives that have given my research on this longstanding challenge of cybersecurity special relevance in this present day.

Finally, I would like to thank all the survey participants and interviewees for their precious time in sharing their perspectives with me. Without these valuable inputs, I would not have arrived at this research's conclusive findings.

I hope this research provides you with useful ideas you can benefit from in your effort to develop and deploy more secure software in your projects.

Table of Contents

1. Introduction

1.1 Background

Cybersecurity threats have been on a rising trend in the past decade or so. Besides leading to an increasing number of breaches, they have also been of increasing impact, as seen in two high profile cases in recent years - the WannaCry Ransomware attack in 2017 and the attack on SolarWinds in 2020. These threats and incidents are expected to continue with widespread digitalization, increasing numbers of 5G connected IoT devices, the prevalent shift towards remote working and the accompanying utilization of cloud services to enable all of these trends. There is also increasing regulation on privacy and security that organizations need to comply with or risk paying hefty fines.

With all these factors, most organizations have invested in multi-pronged strategies to improve their security posture. These include infrastructure-related measures (such as strengthening their network defences and hardening their systems) and process controls (such as user account validity reviews and vetting of personnel). In addition to that, organizations have also started to adopt secure software development practices in order to mitigate the vulnerabilities in the software they develop and deploy.

Hence, it is apt to re-visit research in software security to ascertain the different security measures organizations can adopt and the initiatives they are carrying out to enhance software security. This research will help collect and disseminate practical knowledge between organizations to strengthen their security posture.

1.2 Problem Description

With cybersecurity threats threatening the operations and information confidentiality of organizations worldwide, one of the key measures of defending against them is for organizations to adopt a secure software development activities or framework to ensure the security of the software they deploy.

Secure activities such as adhering to secure coding guidelines, doing code reviews and running penetration tests, can be applied individually depending on the specific risk assessment of the organization. Alternatively, organizations can also adopt a secure software development framework, that recommends the integration of secure development practices throughout the software development lifecycle (such as design, development, testing and deployment) to more holistically address software vulnerabilities. Two examples of such secure software development frameworks include the Microsoft Security Development Lifecycle (Howard et al, 2006) and the Comprehensive, Lightweight Application Security Process released by the Open Web Application Security Project (OWASP, 2006).

Recently, the United States government has also required that all their suppliers implement secure software development practices throughout the software development life cycle and released its guidance with the NIST Special Publication 800-218 on Secure Software Development Framework (SSDF) (Souppaya et al, 2022).

While deciding on which framework or activities to adopt, some of the questions that organizations might have in mind are:

1. Are secure software development frameworks really effective? Which framework provides the most secure outcomes?
2. What are the considerations when deciding among the existing frameworks to adopt?
3. Which secure software development framework is the most widely used?

Organizations are often faced with competing demands on their finite resources and are therefore looking for the most cost-effective and least resource-intensive activities to adopt that can offer the most secure outcomes. With that in mind, this research project helps organizations identify the key activities from the existing secure software development frameworks to adopt to improve the security of their ICT systems. The findings of this study will directly benefit organizations that are developing or deploying software to enable their business process.

1.3 Research Questions

The focus of this research is a few of the more prominent formal secure software development lifecycle (SSDLC) frameworks. These frameworks are also referred to as secure software engineering (SSE) frameworks in some of the publications. For consistency, the term SSDLC will be used throughout this report as a generic reference to both these terms.

In this research, these SSDLC frameworks and their practices will be looked into and compared, along with the factors that organizations need to consider when deciding upon which framework/practices to adopt. The considerations will also be fine-tuned, based on any relevant and valuable insights obtained from the survey and interviews.

More specifically, this research will try to answer the following research questions:

- RQ1: What are the current practices regarding the adoption of secure software development frameworks?
- RQ2: What are the key considerations for organizations in deciding on the SSDLC frameworks and practices to adopt?
- RQ3: For organisations that use custom frameworks, what are the reasons behind this, and which elements are common in these custom frameworks?
- RQ4: What are the historical trends in the adoption of secure software development frameworks? What are the major changes in the past, and what are the potential future developments?

1.4 Thesis Outline

This section presents the outline of the thesis, giving an overview of the motivation of the research, the methods used to explore it and the eventual findings from the study. After this section in the introductory chapter, chapter 2 covers the literature review leading to the motivation of this study. Chapter 3 then describes the research methodology for the survey and interviews, with chapter 4 covering the results and findings. Next, chapter 5 discusses the findings and the insights that can be derived from them. Finally, chapter 6 concludes this thesis and suggests some potential follow-up initiatives.

2. Literature Review

The purpose of the literature study is to discover the cyber security frameworks that can be adopted for the secure development of software. Beside the frameworks, the scope of the literature review will also extend to:

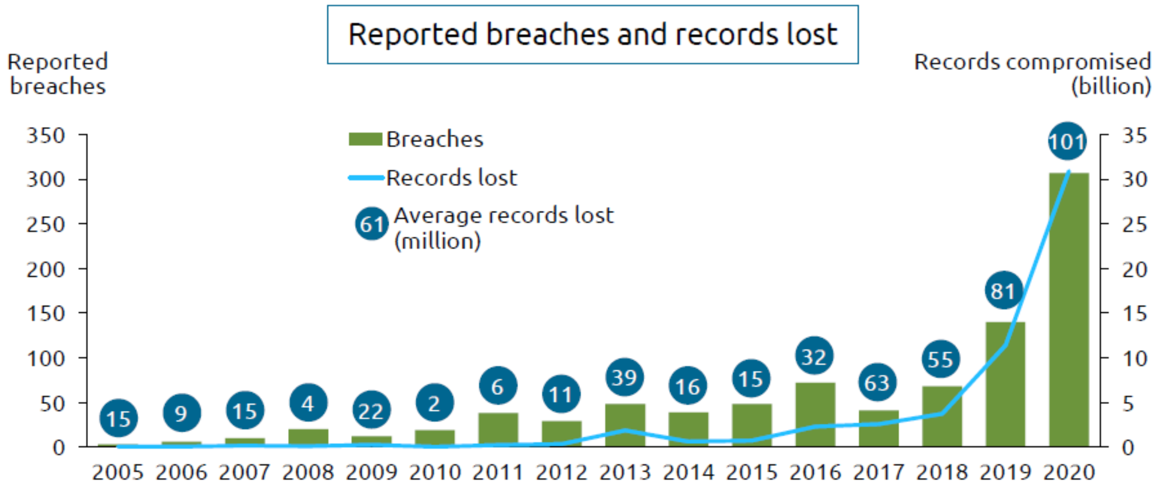
- activities within the frameworks to support the development of software
- activities of other related stages in the software lifecycle besides just the development e.g. training and maintenance

In order to conduct the literature study, tools such as ScienceDirect, ResearchGate, Google Scholar search engine, IEEE white papers, scientific publications and journals were used. The literature of this research is divided into 4 categories:

- Industry standards
- Scientific publications
- Books
- Technology reports

2.1 Recent Cyber-Attacks

The number of cybersecurity breaches has been on a rising trend in recent years. A report on cybersecurity showed that since 2005, there has been and increasing number of breaches and number of data records being compromised (Canalys, 2021). The report also noted that there are no signs of this trend slowing down, especially with the shift to parameter-less IT and the prevalence of digitalization projects. Figure 1 shows the reported cybersecurity breaches and records as presented in the report.

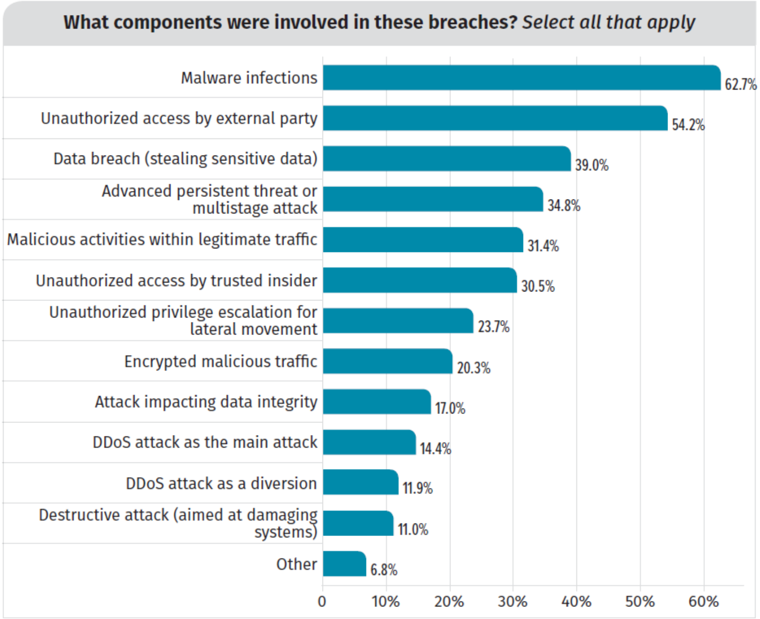


Source: Public sources, January 2021

From "Now and next for the cybersecurity industry – part 1" by Canalys, 2021. Copyright 2021 by Canalys.

Figure 1. Reported Breaches and Records Lost.

There are many varied causes of the cybersecurity breaches with malware infections forming the bulk of them. A SANS incident response survey done in 2019 (Bromiley, 2019) studied the components that are involved in the cybersecurity breaches as presented in Figure 2.



From "SANS 2019 Incident Response (IR) Survey: It's Time for a Change" by M. Bromiley, 2019, SANS. Copyright 2019 by M. Bromiley/SANS.

Figure 2. Components Involved in Cybersecurity Breaches.

Two of the more prominent attacks in recent times are the attack on SolarWinds in 2020 and the WannaCry Ransomware attack in 2017.

Cyber-Attack on SolarWinds

According to an investigation report by a Danish agency (Centre for Cyber Security, 2021), in March 2020, some hackers broke into SolarWinds systems and added malicious code to one of the company's software products. That particular software product was widely used by many of SolarWinds' customers to manage their IT systems and solutions. SolarWinds, like many other IT vendors, provides software updates (to add features or fix bugs) to their customers from time to time. After the hack, SolarWinds, unknowingly continued sending out these software updates to its customers and that included the malicious code, which spread to many of its customers and went undetected for months. This code created a backdoor in their customers' IT systems and infrastructure which the hackers utilized to install even more malware that enabled them to spy on these companies.

WannaCry Ransomware

In May 2017, the WannaCry ransomware spread rapidly and hit hundreds of thousands of computers across more than 100 countries. It swept across Europe and Asia quickly, locking up critical systems in businesses and organizations around the world. WannaCry affects computers running the Windows Operation System (OS). Once infected, it encrypts the contents of the hard drive on a victim's computer and denies access to the user. It then displays a message demanding a ransom in bitcoin payment before the contents will be decrypted. One of the major institutions to be adversely affected is the United Kingdom's National Health Service (NHS). About 60 NHS facilities were affected by the WannaCry ransomware. These facilities could not access patient records and had to delay non-urgent surgeries and cancel patient appointments. Eventually, NHS paid the ransom demanded by the hackers to get their data back. In the investigation after the incident, it was discovered that the infection happened due to employees opening malicious emails in an unpatched system.

These two attacks highlight the need for organizations to take necessary measures to improve their security posture, such as installing the necessary security components to protect their infrastructure (by deploying e.g. firewall, anti-virus VPN, SIEM), providing employees with the necessary cybersecurity education, adopting secure practices in the development of software used and securing the software supply chain, which is an important activity during the release and deployment stage of the software development lifecycle.

2.2 Remote Working Security Risks

Besides the already increasing number of cyber-attacks prior to the covid-19 pandemic, the shift towards remote working further increases the attack surface and opportunities for hackers as more people use the internet from their homes for work and communication, when they could be using local area networks (LAN) or intranets within their organizations before (Forrester, 2021). Some of the main security risk with remote working include the following unencrypted file sharing and unsecured home devices, including personal notebooks and WiFi networks. Employees handle sensitive information, from customer data to proprietary product information. When they work on this sensitive information remotely, storing them on home devices or on the cloud, without storage encryption or encryption while in-transit, data theft can happen. Organizations also report increased risks due to them moving more business functions to the cloud, so as to enable a remote workforce.

Tessian, a cybersecurity company, also published a security behaviours report (Tessian, 2021) looking into the behaviour of employees as businesses transition back to the office and the majority adopt a hybrid approach to work. The survey found that about a third (30%) of employees believe that they can get away with riskier security behaviours when working remotely with about 2 in 5 (39%) admitting that their cybersecurity practices at home were less thorough than those practised in the office. The reason for this is that employees admitted feeling less scrutinised by their IT departments.

In tandem with the shift towards more remote working, organizations also relied more on cloud services to support remote work and home offices (Rials, 2021). The quick pivot to cloud enabled operations to continue and organizations to extend their borders and enable remote working but it also introduced new security risks with most services enabled with default configurations that do not take security into account.

2.3 Related Cybersecurity Literature

Secure Development Activities and Interventions

In response to the cybersecurity threats and risks highlighted, there are a myriad of measures that have been developed and proposed. Marin (2005) presented basic measures for securing the network by deploying intrusion detection systems. Ibor et al (2015) proposed a system hardening strategy, where several different security measures are applied to the host, application and operating system, all of which must be overcome before the system can be compromised. Despite these measures, organizations have realized that "securing the perimeter" by means of firewalls and intrusion detection systems is not sufficient to protect against cybersecurity incidents when the software that is developed and deployed has vulnerabilities (Steven, 2006).

To minimize or eliminate software vulnerabilities, organizations need to adopt secure software development activities. Studies have been conducted to highlight and recommend some of these activities. Myagmar et al (2005) investigated how threat modeling can be used to identify the threats to a system and proposed it as a foundation for defining relevant security requirements for the system. Huang et al (2004) tested and described an effective approach to ensuring web application security by employing static code analysis techniques. Shah et el (2015) tested and concluded the effectiveness of other proactive security measures such as carrying out Vulnerability Assessment (VA) and Penetration Testing (PT). There is a whole spectrum of such secure development activities or interventions and a study was conducted to investigate 20 of these security interventions (Such et al, 2016). The study comprised a survey involving industry stakeholders to derive their perception of the characteristics of these interventions such as their cost effectiveness, manpower requirements and duration required for implementation. More recently, Weir et al (2021) conducted a study that identified 12 of the most used secure development activities. Among these 12 are doing penetration testing with internal and external testers and using automated tools along with manual review of codes.

Secure Development Frameworks

A software development life cycle (SDLC) is a methodology for designing, developing, testing and deploying software. There are many SSDLC frameworks that integrate secure development practices throughout the SDLC to more holistically address software vulnerabilities. They are published by companies in the software industry, academia and even government organizations. Souppaya et al (2022), who developed the National Institute of Standards and Technology (NIST) Special Publication 800-218 on Secure Software Development Framework (SSDF), described fundamental practices for these frameworks at a high-level and referenced many of these frameworks for details. De Win et al (2009) also compared the secure software development activities between three of these frameworks. Based on these literature and other internet sources, the more prominent frameworks were ascertained to be:

1. Microsoft Security Development Lifecycle (SDL)
2. OWASP Comprehensive, Lightweight Application Security Process (CLASP)

3. McGraw’s Touchpoints
4. NIST Special Publication 800-160 on Systems Security Engineering
5. Software Assurance Forum for Excellence in Code (SAFECode)

In addition to that, the Supply-chain Levels for Software Artifacts (SLSA), a security framework to prevent tampering and improve integrity of software packages, is also growing in prominence in the light of software supply chain attacks, such as the one on SolarWinds. The commonalities, differences and secure activities between these 6 secure development frameworks will be delved into and compared in the next section.

Besides these 6 prominent frameworks mentioned so far, Núñez et al (2020) also proposed the Viewnext-UEx model, an emerging and preventive approach to develop secure software. This model included the better security activities from the best-known models in secure software development. Other frameworks have been released by nation-wide ICT regulation authority such the Security-by-Design Framework (Cyber Security Agency of Singapore, 2017), which gives its guidance to organizations in designing and building security in every phase of the system development life cycle. It is not possible to explore all the frameworks in detail in this study. However, the survey and interviews will ascertain if any other frameworks are actually adopted in the industry.

Challenges in Adoption

Although there is substantial information available on secure development activities and frameworks, but several studies have shown that organizations are facing challenges in adopting them. Assal et al (2019) observed that many software developers are motivated to develop secure software but are deterred when they have to deal with competing priorities, the lack of resources or insufficient security knowledge. Gasiba et al (2020) in their recent work, also found that while software developers were intent on complying with secure coding guidelines, their knowledge of the guidelines are lacking. This indicates the need for running secure coding training and awareness campaigns.

Several other studies have also observed challenges faced and suggested some form of adaptation of security interventions to better fit the context of the organization or employees involved in the activities. Kirlappos et al (2013), conducted a survey and identified the perceived conflict of security with productive activities as the key driver for employees non-compliance to security-related policies in organizations. They concluded that an effective resolution to this problem requires security measure adaptation and de-centralisation of the decision on how to implement security in specific contexts to employees. Another research found that developers are often over-burdened by other requirements competing for their attention. In addition to these demands on their attention, their specific circumstance and environment also frequently impede their behaviour to code securely (Rauf et al, 2022). The study provided an explanation on why existing interventions are not effective in addressing all security interventions and proposed adaptive interventions that are contextually-aware. At the organization level, the obstacles faced were highlighted in yet another study involving CISO’s of small businesses (Wolf et al, 2021). The observations from this study led to recommendations for tailoring of the processes for their needs and highlighting the resource requirements when adopting any security guidance or standards.

At the framework level, an earlier study by Geer (2010) discovered that formal SSDLC frameworks were not widely adopted by organizations, due to challenges such as cost and complexity.

2.4 Comparison of SSDLC Frameworks

The secure activities between the 6 more prominent secure development frameworks will be compared in this section. As highlighted in the previous section, the 6 more prominent SSDLC frameworks are:

1. Microsoft Security Development Lifecycle (SDL)
2. OWASP Comprehensive, Lightweight Application Security Process (CLASP)
3. McGraw's Touchpoints
4. NIST Special Publication 800-160 on Systems Security Engineering
5. Software Assurance Forum for Excellence in Code (SAFECode)
6. Supply-chain Levels for Software Artifacts (SLSA)

To expand upon an earlier study conducted by De Win et al (2009), which compared the secure software development activities between CLASP, SDL and Touchpoints, this report will preserve the same organization of activities among the different frameworks, for example, architecture-level threat modelling (4.3) and software attack surface reduction (5.2). This report will also preserve the same classification of these activities according to the phases of a typical software development life cycle:

1. Education and Awareness
2. Project Inception
3. Analysis and Requirements
4. Architectural Design
5. Detailed Design
6. Implementation
7. Testing
8. Release and Deployment
9. Support

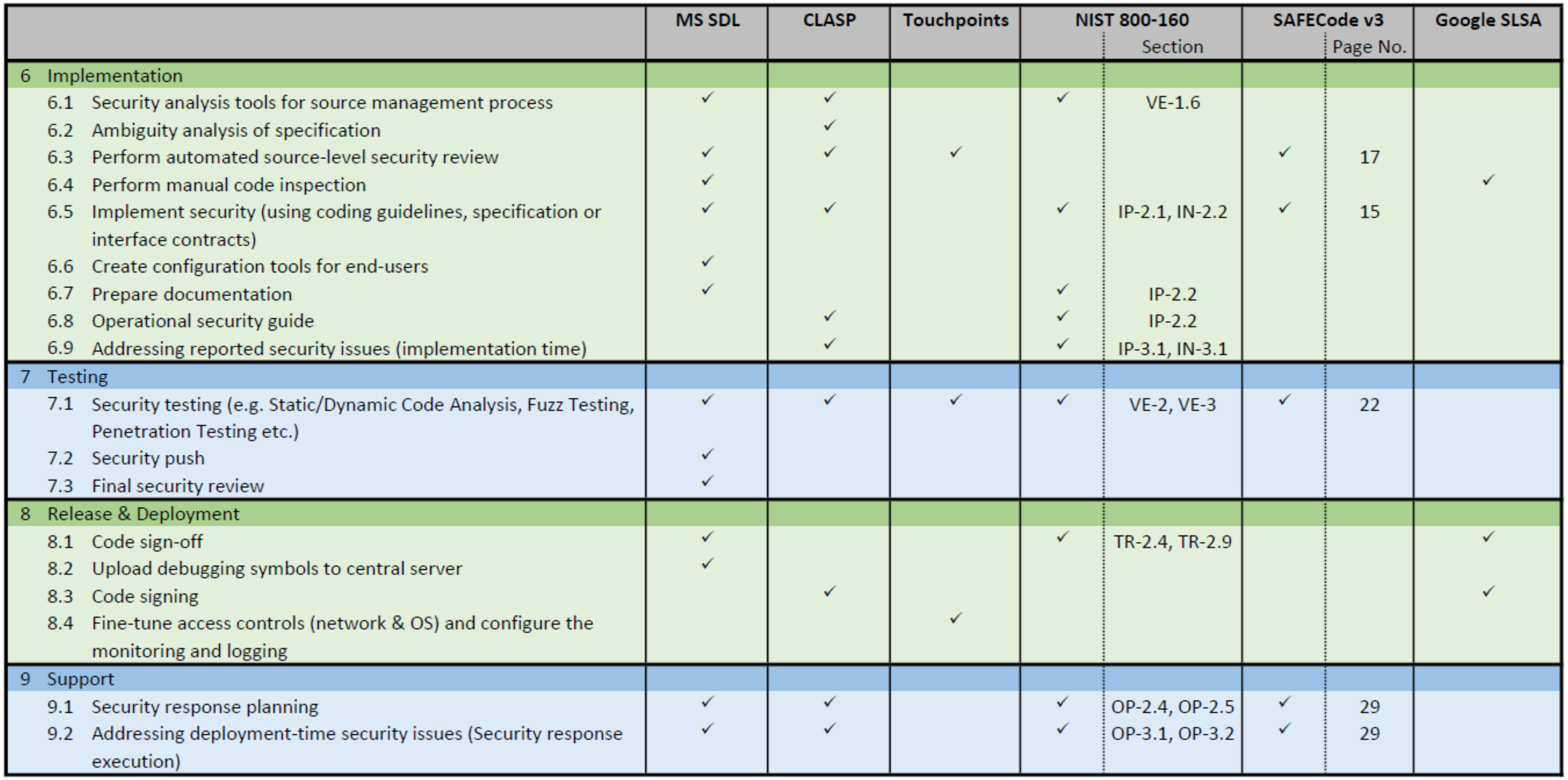
It should be noted that phases 2 to 8 are typical phases of a software development life cycle. Education and Awareness (1) comprises organisation-wide, as well as, project-specific activities. Support (9) as quoted in the earlier comparison between CLASP, SDL and Touchpoints (De Win et al, 2009), comprises activities for security response planning and execution to address deployment time issues, rather than the activities for the maintenance phase.

As the way of choosing the terminology for each activity and organizing them, has already been well rationalized in the earlier study, this report will follow the same naming convention, to facilitate orientation and comparison of other frameworks. Building upon the original study, the same activities in the comparison table and naming convention will be used. Also in the original study the comparison table goes down to three levels – phases (level 1), activities (level 2) and sub-activities (level 3). Once again, an example of an activity is software attack surface reduction (5.2) and three of the sub-activities under this activity include remove unimportant features (5.2.1), determine who needs access from where (5.2.2) and reduce privileges (5.2.3).

In this current report, we will broadly look at the top 2 level (phases and activities). For each framework, we will consider the activity as being covered if one of the sub-activities is described. The phase and activity comparison of the 6 secure development frameworks is illustrated in Table 1. The phases and activities for the first three columns (MS SDL, CLASP and Touchpoints) have been summarized from the earlier research by De Win et al (2009). The corresponding information of the remaining three security frameworks (NIST 800-160, SAFECode & SLSA) have been derived by examining their respective literature in detail.

From the comparison it can be seen that all the frameworks roughly recommend similar activities across the 9 lifecycle phases of software development. Some are comprehensive and recommend activities in all phases while others (e.g. SLSA) focus only on certain aspects of the entire development lifecycle. In the wake of the attack on SolarWinds, the specific focus of SLSA in ensuring the security of the software supply chain becomes critical to adopt. Only code sign-off (prescribed in MS SDL) and code signing (prescribed in Touchpoints) contribute toward that objective. However, both of these two frameworks do not prescribe processes as holistic as that prescribed by SLSA in ensuring the security of the software supply chain.

	MS SDL	CLASP	Touchpoints	NIST 800-160		SAFECode v3		Google SLSA
					Section		Page No.	
1 Education and awareness								
1.1 Baseline education	✓	✓		✓	HR-1, HR-2	✓	33	
1.2 Educate Infosec people on the applications and development environment			✓					
1.3 Advanced education	✓	✓		✓	HR-1, HR-2	✓	33	
1.4 Share security artifacts with team		✓						
2 Project Inception								
2.1 Determine whether the application is covered by methodology	✓					✓	34	
2.2 Security Team (bring together)	✓	✓	✓	✓	HR-3			
2.3 Address security logistics (e.g. bug-tracking tools)	✓			✓	AR-1.5			
2.4 Decide what types of bugs you are going to fix	✓							
2.5 Security Metrics (identity, collect & evaluate)	✓	✓	✓	✓	AR-1.4	✓	35	
2.6 Global security policy		✓						
2.7 Build and execute process improvement program			✓					
3 Analysis & Requirements								
3.1 Resources and trust boundaries		✓						
3.2 User roles and resource capabilities		✓		✓	SN-2.1, SN-2.2			
3.3 Analysis-level threat modeling		✓	✓	✓	SN-2.4			
3.4 Security-relevant requirements		✓	✓	✓	SN-1 to SN-6, SR-1 to SR-3			
3.5 Execute Risk Management Framework			✓	✓	SN-2.3			
3.6 Requirements for the operational environment		✓		✓	SN-3, SN-4			
3.7 Define use scenarios for threat modeling	✓			✓	SN-2.4			
4 Architectural Design								
4.1 Risk assessment for 3rd party products	✓	✓	✓			✓	21	
4.2 Requirements audit		✓		✓	SR-3			
4.3 Architecture-level threat modeling	✓	✓	✓	✓	AR-2, AR-3	✓	10	
4.4 Security design principles		✓		✓	AR-4	✓	9	
5 Detailed Design								
5.1 Assess the privacy impact rating of the project	✓							
5.2 Software attack surface reduction	✓	✓		✓	DE-1.5, DE-2.2			
5.3 Class design annotation		✓						
5.4 Database security configuration		✓						
5.5 Make your product updatable	✓			✓	DE-1.4			
5.6 Determine the security technologies required				✓	DE-1.2			



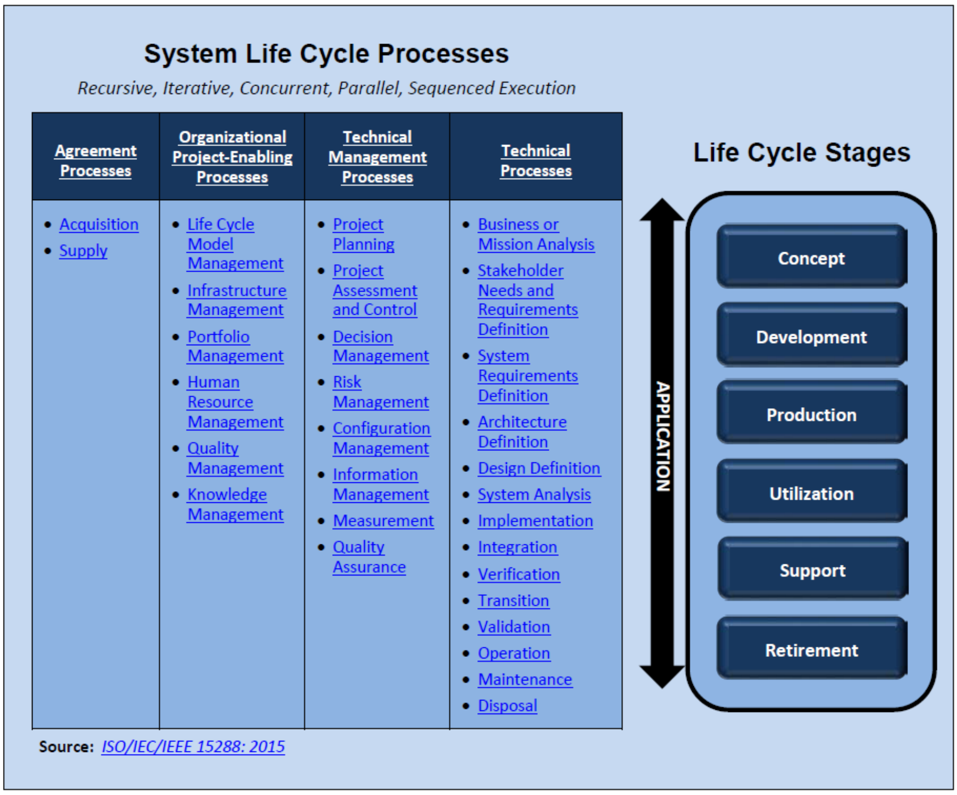
		MS SDL	CLASP	Touchpoints	NIST 800-160		SAFECode v3		Google SLSA
						Section		Page No.	
6	Implementation								
6.1	Security analysis tools for source management process	✓	✓		✓	VE-1.6			
6.2	Ambiguity analysis of specification		✓						
6.3	Perform automated source-level security review	✓	✓	✓			✓	17	
6.4	Perform manual code inspection	✓							✓
6.5	Implement security (using coding guidelines, specification or interface contracts)	✓	✓		✓	IP-2.1, IN-2.2	✓	15	
6.6	Create configuration tools for end-users	✓							
6.7	Prepare documentation	✓			✓	IP-2.2			
6.8	Operational security guide		✓		✓	IP-2.2			
6.9	Addressing reported security issues (implementation time)		✓		✓	IP-3.1, IN-3.1			
7	Testing								
7.1	Security testing (e.g. Static/Dynamic Code Analysis, Fuzz Testing, Penetration Testing etc.)	✓	✓	✓	✓	VE-2, VE-3	✓	22	
7.2	Security push	✓							
7.3	Final security review	✓							
8	Release & Deployment								
8.1	Code sign-off	✓			✓	TR-2.4, TR-2.9			✓
8.2	Upload debugging symbols to central server	✓							
8.3	Code signing		✓						✓
8.4	Fine-tune access controls (network & OS) and configure the monitoring and logging			✓					
9	Support								
9.1	Security response planning	✓	✓		✓	OP-2.4, OP-2.5	✓	29	
9.2	Addressing deployment-time security issues (Security response execution)	✓	✓		✓	OP-3.1, OP-3.2	✓	29	

Note. The data for MS SDL, CLASP and Touchpoints are summarised from On the secure software development process: CLASP, SDL and Touchpoints compared by B. De Win, R. Scandariato, K. Buyens, J. Grégoire, & W. Joosen, 2009. Copyright 2009 by Elsevier B.V.

Table 1. Phase and Activity Comparison of the Secure Development Frameworks.

NIST Special Publication 800-160

The security activities and considerations highlighted in the NIST Special Publication 800-160 on Systems Security Engineering (Ross et al, 2016) covers the following life cycle processes as summarised in Figure 3.



System Life Cycle Processes

Recursive, Iterative, Concurrent, Parallel, Sequenced Execution

Agreement Processes	Organizational Project-Enabling Processes	Technical Management Processes	Technical Processes
• Acquisition • Supply	• Life Cycle Model Management • Infrastructure Management • Portfolio Management • Human Resource Management • Quality Management • Knowledge Management	• Project Planning • Project Assessment and Control • Decision Management • Risk Management • Configuration Management • Information Management • Measurement • Quality Assurance	• Business or Mission Analysis • Stakeholder Needs and Requirements Definition • System Requirements Definition • Architecture Definition • Design Definition • System Analysis • Implementation • Integration • Verification • Transition • Validation • Operation • Maintenance • Disposal

Life Cycle Stages

Source: *ISO/IEC/IEEE 15288: 2015*

From "NIST Special Publication 800-160, Volume 1. Systems Security Engineering: Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems" by R. Ross, M. McEvilley, & L. Oren, 2016, National Institute of Standards and Technology. Copyright 2018 by National Institute of Standards and Technology.

Figure 3. System Life Cycle Processes and Life Cycle Stages.

As illustrated, NIST 800-160 covers more than just security activities for the project-specific lifecycle from inception to its deployment for operational use. It also includes the activities for other processes such as:

- Agreement Processes (such as Acquisition and Supply)
- Organizational Project-Enabling Processes (such as Human Resource Management and Portfolio Management)
- Technical Management Processes (such as Risk Management and Project Planning)
- Processes after the software development life cycle (such as Operation, Maintenance and Disposal)

Some examples of these additional activities that were not mentioned in the earlier comparison of the 3 frameworks are:

- Section 3.4.12 (Operation Process) mentions activities such as training the necessary personnel for the secure operation of the system.
- section 3.4.11 (Validation Process) mentions processes that aims to “provide objective evidence that the system, when in use, fulfills its business or mission objectives and stakeholder requirements, achieving its intended use in its intended operational environment.” This could be regular audits that take place after the system is operational.
- section 3.4.13 (Maintenance Process) mentions processes such as preventive maintenance e.g. applying patches, in response to knowledge gained about vulnerabilities.

The NIST 800-160 framework is generic and applies to any form of systems not just software systems. As such, terminology used for the tasks may not be so specific to software development. Some form of interpretation was done and as long as the description of the tasks is similar to that used in software development, the tasks defined in NIST 800-160 were mapped to the equivalent software development activities.

SAFE Code

The Software Assurance Forum for Excellence in Code (SAFECode) first published the “SAFECode Fundamental Practices for Secure Software Development” in 2008 to help others in the industry initiate or improve their own software assurance programs and encourage the industry-wide adoption of fundamental secure development practices. The most current, 3rd edition of “SAFECode Fundamental Practices for Secure Software Development” (SAFECode, 2018) is referenced for this research. It includes updates to the fundamental practices to reflect current best practice.

In a nutshell, the practices recommended by SAFEcode cover the following phases:

- Design Phase. It explains the secure design principles and gives an overview of key secure design activities such as threat modelling, develop an encryption strategy, standardizing identity and access management and establishing log requirements and audit practices.
- Development Phase. It recommends the adoption of coding standards and conventions to prevent introducing vulnerabilities in the code e.g. input validation, error handling and the use of code analysis tools to identify security issues in the code as early as possible. It also highlights the importance of managing the security risk that comes with the use of third-party components & libraries.
- Testing Phase. It highlights the importance of doing the key types of testing to check the developed software for vulnerabilities - Static Analysis Security Testing, Dynamic Analysis Security Testing, Fuzzing, Network Vulnerability Scanning and Penetration Testing.

SLSA

Unauthorized modifications to software packages (also known as supply chain integrity attacks) have been on the rise in recent years. These attacks can happen anywhere along the life cycle during the phases from the development of source code (develop) to its integration with all other code/modules (build) and the subsequent deployment (publish). There is currently no comprehensive framework to mitigate threats across the software supply chain and there is an urgent need in the face of such attacks such as the one on SolarWinds.

Supply chain Levels for Software Artifacts (SLSA) is a community effort organized within the Open Source Security Foundation (OpenSSF), comprising members from organizations such as Google, Intel, Linux Foundation and VMWare. The community has proposed an end-to-end framework for ensuring the integrity of software artifacts throughout the software supply chain (OpenSSF, 2022). The solution consists of four levels, with SLSA 4 representing the desired end state. Achieving the highest level of SLSA may be difficult, but incremental improvements brought about by achieving lower SLSA levels will already contribute towards improving the security of developed applications. The requirements of the four levels are currently defined as follows:

- **SLSA 1** - the build process is scripted and fully automated to generate provenance (metadata about how an artifact was built including the source and dependencies). Although the metadata generated at this level cannot protect against tampering, it supports source code identification that is useful in vulnerability management.
- **SLSA 2** - requires using version control and a trusted build service to generate provenance. The trusted build service provides assurance that the provenance has not been tampered with.
- **SLSA 3** – requires that the source and build platforms meet stated specifications to ensure that information about the source can be inspected and the provenance generated is reliable.
- **SLSA 4** – requires a peer review of all changes and an airtight build process. This provides a high level of confidence that the software has not been tampered with.

2.5 Motivation

Prior work has offered useful guidance on the secure development activities and frameworks that organizations can adopt for their software assurance. Many of the challenges faced in the adoption of these secure activities have also been observed and options for resolutions have been proposed.

The focus of this research will be on the current practice in the adoption of secure development framework. Although the study by Geer (2010) provided an indication that formal SSDLC frameworks were not widely adopted by organizations, however, his findings may be outdated with many recent factors such as increasing digitalization, remote working, the rise in cyber threats and new government regulations. As such, the survey and interviews of this research will measure if more organizations have since started to adopt any formal frameworks in recent years. As part of discovering the current practices in the adoption of SSDLC frameworks or secure activities, the survey and interviews will also ascertain how closely companies follow and enforce the activities that they have adopted.

Based on existing literature, the examination of reasons for adopting or not adopting a formal SSDLC framework has also not been explored in much detail, as well as, the considerations that organizations take should they develop their own custom framework. This study will attempt to ascertain these reasons and consideration. In addition, the secure activities commonly adopted when organizations customize their own secure development framework, will be gathered. This can be compared against the most used secure development activities identified by Weir et al (2021).

Although existing literature has surfaced the trends and many challenges in the adoption of security activities, this study will also elicit from the respondents and interviewees the adoption trend that they have experienced in the past and the potential future trends they anticipate, given the numerous recent developments that affect the security of software and systems.

3. Methodology

3.1 Research Approach

The research approach included a literature study of software security frameworks and activities that have been adopted by development teams. The literature study aimed to provide an understanding of what is available in the industry and gauge its adoption and effectiveness based on past research. After reviewing the related literature on secure development frameworks and interventions, this study then went about its objective of getting inputs from current practitioners in software development projects. This was achieved by two methods. The first was an online survey and the second was a series of interviews with these practitioners.

3.2 Survey Methodology

The survey was first conducted as a qualitative method to measure the adoption rate of secure development frameworks and the various security activities in the development of software. The survey also aims to understand the reason behind the responses. Most of the questions are multiple choice questions that allows for multiple selections with an option for an "Others" input that includes text entry fields. These fields allow the participants to pen down their comments in freeform text. The survey questions from an Android developer survey (Weir, Hermann, et al, 2020) were referenced and adapted from in crafting the questions for this online survey. Questions were included in an attempt to obtain responses to answer the research questions.

The survey published online on the Qualitrics platform. It was opened to responses from 29 March 2022 to 30 April 2022 and required approximately 12 minutes to complete. There were a total of 27 questions split into 4 sections:

- Section 1 – Project Information
- Section 2 – Cybersecurity Questions
- Section 3 – Demographic Information (personal and company)
- Section 4 – Concluding Questions (and email contact)

The complete list of survey questions is included in **Appendix 1**.

The cybersecurity questions from the survey contribute to answering the research questions, as shown in the mapping in Table 2.

Research Questions	Survey Questions on Cybersecurity (Q4 to Q18)
Knowledge & attitude toward cybersecurity	Q4 - Has your company/project ever experienced a cyber-attack related to software that was developed or deployed?
	Q5 - Do you think adopting secure practices or a secure software development framework helps to prevent or reduce the occurrence of cyber-attacks?
	Q7 - What secure software development life cycle or secure software engineering (SSDLC/SSE) frameworks have you heard of?
	Q12 - How necessary do you think adopting one of these frameworks is?
RQ1: What are the current practices regarding the adoption of secure software development frameworks?	Q6 - Who in the company is overall responsible to ensure that developed/deployed software conform to the framework?
	Q8 - What SSDLC/SSE framework is your project currently using?
	Q13 - How strictly does your project enforce the security processes of this framework?
	Q14 - What are the possible reasons for the framework NOT being "almost fully enforced"?
RQ2: What are the key considerations for organizations in deciding on the SSDLC frameworks and practices to adopt?	Q10 - Can you explain the reason(s) for your project adopting the mentioned framework?
	Q11 - Which of the following benefit(s) have you observed after adopting the mentioned framework?
RQ3: For organisations that use custom frameworks, what are the reasons behind this, and which elements are common in these custom frameworks?	Q9 - What SSDLC stages/activities does your project follow?
	Q10 - Can you explain the reason(s) for your project adopting the mentioned framework?
RQ4: What are the historical trends in the adoption of secure software development frameworks? What are the major changes in the past, and what are the potential future developments?	Q15 - When did your company start to adopt a secure software development life cycle or secure software engineering (SSDLC/SSE) framework?
	Q16 - Do you think there is a greater need to adopt an SSDLC/SSE framework as companies increasingly rely on remote working?
	Q17 - Has there been any changes or additions to the framework that has been adopted in the last 2 years?
	Q18 - What are the reasons for the changes?

Table 2. Mapping of Research Questions to Survey Questions

For the survey, participants were chosen opportunistically based on personal connections who were involved in IT projects, undertaking roles such as developers, reviewers, architects, team leaders or project managers. Attempts were also made to reach out to potential participants in focus groups and communities on the LinkedIn and Reddit platforms.

Also they were approached as individuals rather than representatives of organizations to eliminate any reluctance they might encounter about their answers being representative of their organizations' viewpoints or revealing too much about the specific organization's cybersecurity strategy. As such, only demographic information about their projects and organisations were gathered for the purpose of classification. Naturally, their anonymity is also a condition proposed for their frank opinion to the survey and interviews.

At the close of the survey, the results was downloaded, visualized with the charting feature of Microsoft Excel and subsequently analysed. Eventually, the interview questions were designed based on where more detailed elaboration was required or where the verbatim responses from the survey needed further clarification.

3.3 Interview Methodology

After the survey, interviews were done with the aim of obtaining inputs from some current practitioners to get a deeper understanding into the current practice of secure software development, as well as, other related insights that the interviewees can contribute to this topic. A semi-structured interview was employed to ensure some form of uniformity and to prevent the conversation from veering off the intent to provide insights to the research questions. The interview questions were meant to guide the conversation during the interview but eventually the questions that were asked depended on the answers the interviewees provided during the actual interview. Of course, this meant that some questions were skipped or slightly modified to fit the context of the conversation.

All the interviews were conducted online using the Zoom platform and recorded for subsequent transcription. One-on-one interviews were conducted to reduce the possibility of the interviewee's opinion being swayed by other participants.

A brief introduction of the research objective was given before the interview began and the interviewees were reminded to express their own frank opinions. They were also reminded that the interview would be recorded. The first few questions focused on demographic information on the interviewees, their company and the projects they are involved in. Only non-personally identifiable information were discussed, such as, the number of years of experience the interviewee has, his company size and project team size. After that, the interview moved on to the cybersecurity related questions such as the secure development frameworks or activities adopted in their projects and the reasons for their selection. The interviewees were also asked if there have been any recent additions or changes to their practices in the last couple of years. Finally, they were asked to share their perspective of the trend they see in the adoption of security frameworks and practices. The full set of semi-structured interview questions is shown in **Appendix 2**. Also, the interview questions contribute to answering the research questions, as shown in the mapping in Table 3.

Research Questions	Interview Questions on Cybersecurity
Knowledge & attitude toward cybersecurity	Q9. Could you describe in your own words what are secure software development frameworks?
	Q10. Do you think adopting secure practices or a secure software development framework helps to prevent or reduce the occurrence of cyber-attacks?
	Q11. What secure software development life cycle or secure software engineering (SSDLC/SSE) frameworks have you heard of?
RQ1: What are the current practices regarding the adoption of secure software development frameworks?	Q12. What SSDLC/SSE framework is your project currently using? Can you explain the reason for your project adopting the mentioned framework? - The majority of our survey respondents use custom frameworks. What do you think could be the reasons behind this? - Does your company have a specific policy regarding secure software development? (If yes) Can you tell me more about this policy?
	Q14. How strictly does your project enforce the security processes of this framework? - How does your company/project enforce the processes? - If the framework is NOT almost fully enforced, what are the possible reasons?
RQ2: What are the key considerations for organizations in deciding on the SSDLC frameworks and practices to adopt?	Q12. What SSDLC/SSE framework is your project currently using? Can you explain the reason for your project adopting the mentioned framework? - The majority of our survey respondents use custom frameworks. What do you think could be the reasons behind this? - Does your company have a specific policy regarding secure software development? (If yes) Can you tell me more about this policy?
RQ3: For organisations that use custom frameworks, what are the reasons behind this, and which elements are common in these custom frameworks?	Q12. What SSDLC/SSE framework is your project currently using? Can you explain the reason for your project adopting the mentioned framework? - The majority of our survey respondents use custom frameworks. What do you think could be the reasons behind this? - Does your company have a specific policy regarding secure software development? (If yes) Can you tell me more about this policy?
	Q13. What are the major stages/activities that your project follows from these frameworks? - What do you think are the key and important stages/activities that must be done to improve the security posture?
RQ4: What are the historical trends in the adoption of secure software development frameworks? What are the major changes in the past, and what are the potential future developments?	Q15. Has there been any changes or additions to the framework that has been adopted in your project in the last few years? If so, can you describe the major changes and the reasons for the change?
	Q16. In your professional opinion and based on your experience, how would you describe the past trend and future development of adopting secure software development frameworks or practices?

Table 3. Mapping of Research Questions to Interview Questions

The interviewees were recruited from 2 main sources. The first was the participants of the survey, who were asked if they were willing to take part in the interviews. The rest of the interviewees were nominated based on connections and the relevance of their current role in software development projects. After the participants were recruited, an email was sent informing them of the objective of the research and to give them a rough idea of the interview questions. They were also informed that they could withdraw from the process at any point, if they wanted to.

Throughout the whole process, the information gathered are:

- The name and email addresses of the participants in the informed consent email. This will only be kept by the research team and filed for review by the ethics committee
- Audio recording of the interviews that will collect:
    - The demographic information of the interviewees (years of experience and job role)
    - Size of their project
    - Size and type of company

The interview questions were designed to ensure that the answers provided by the participants could not be traced back to them. The participant was also asked for permission to process data in the informed consent email which describes how the data will be handled and stored.

The interviews were recorded and manually transcribed in ATLAS.ti. To ensure that the analysis of the data was reliable, the transcript for one of the interviews (the sample) was coded by an independent coder and the inter-coder agreement was measured using ATLAS.ti based on the Krippendorff alpha coefficient as an indicator. A test coding was first done to ensure the stability of the semantic domains and codes defined, and to ensure that both coders understood the meaning of the codes. After the initial test coding, a final independent coding was done on the sample and the Krippendorff alpha was measured.

4. Results

4.1 Survey Results

As mentioned earlier, the survey was published online on the Qualitrics platform and opened to responses from 29 March 2022 to 30 April 2022. There were a total of 27 questions split into 4 sections:

- Section 1 – Project Information
- Section 2 – Cybersecurity Questions
- Section 3 – Demographic Information (personal and company)
- Section 4 – Concluding Questions (and email contact)

The complete list of survey questions is included in **Appendix 1** and Table 2 in the previous chapter also illustrates how the cybersecurity questions contribute toward answering the research questions.

For the survey, participants were asked questions about security practices in their current or most recent project done within the last 2 years. The project information was asked first as they were important to set the context of the cybersecurity questions. The demographic information (personal and company) was asked at the end, to ensure that their responses to the more important cybersecurity questions were recorded, should they decide to abandon the survey half-way through.

A total of 67 individuals responded to the online survey from which 37 valid respondents were obtained. Respondents were considered valid if the participants answered all the key cybersecurity questions in the survey. They key cybersecurity questions that were necessary for a meaningful interpretation of the results are questions 7, 8, 9 and 10 as listed here:

- Q7. What secure software development life cycle or secure software engineering (SSDLC/SSE) frameworks have you heard of?
- Q8. What SSDLC/SSE framework is your project currently using?
- Q9. What SSDLC stages/activities does your project follow?
- Q10. Can you explain the reason(s) for your project adopting the mentioned framework?

This is to disregard all the responses from other participants who provided incomplete responses that may be due to the following reasons:

- Participants realizing that the questions are too difficult and that he/she is not a suitable participant for the survey
- Participants who were just looking through the questions for reference

To avoid confusion, from this point onwards, we will term these 37 as valid respondents and the options they selected for each of the questions will be termed responses. For the rest of this section, unless explicitly stated otherwise, only the responses from the 37 valid respondents are shown in presenting the results to the survey questions.

Although the order of the sections was different when done by the respondents, the results will be presented in the following order for easier understanding:

- Project Information (Q1 to Q3)
- Demographic Information (personal and company) (Q19 to Q24)
- Cybersecurity Questions (Q4 to Q18)

4.1.1 Project Information

For the survey, participants were asked questions about security practices in their current or most recent project done within the last 2 years. The first few questions (1 to 3) cover the role of the participant in the project and other project-related questions. Figure 4 summarizes the responses to the very first question. The requirement is for participants with roles in IT projects, such as (but not limited to) developers, testers, reviewers, consultants, architects, team leaders or project managers. There were a total of 58 responses for this question, as some of the 37 valid respondents indicated that they assumed more than one role in their projects, giving an average of about 1.6 roles per respondent. The responses show that the survey respondents play roles that make them suitable participants for the intent of this survey.

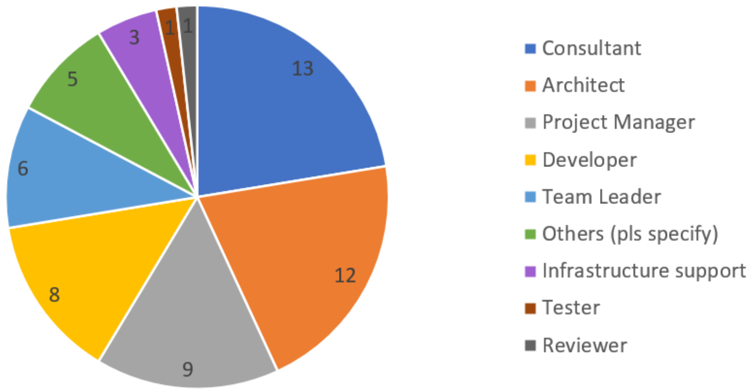


Figure 4. Responses to (Q1) What is your role in the project?

Figure 5 summarizes the responses to the second question. Most of the participants were involved in projects that included the development phase of the life cycle. Quite a number also indicated the acquisition and retirement phases, since they were listed as one of the options for selection. Verbatim responses for the 2 who indicted "Others" were:

- Mainly on DevOps for application software
- Regulatory compliance evaluation

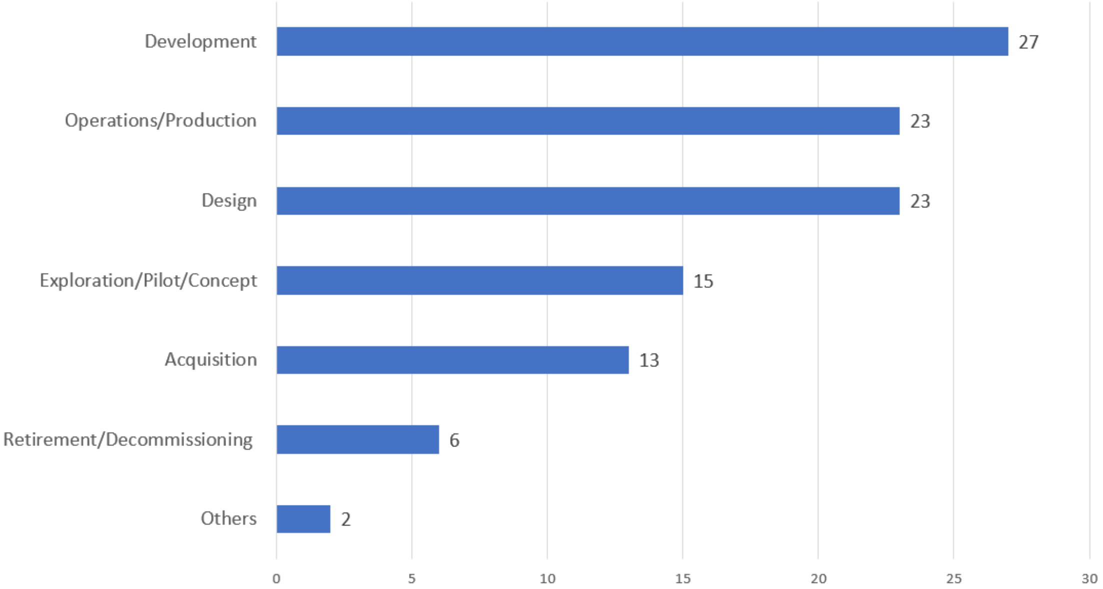


Figure 5. Responses to (Q2) What phases of software lifecycle does your project cover?

Figure 6 summarizes the responses to the 3rd question. The majority of the projects have very small team size of 2 to 5 people. A small minority were very large and has more than 50 people in the project.

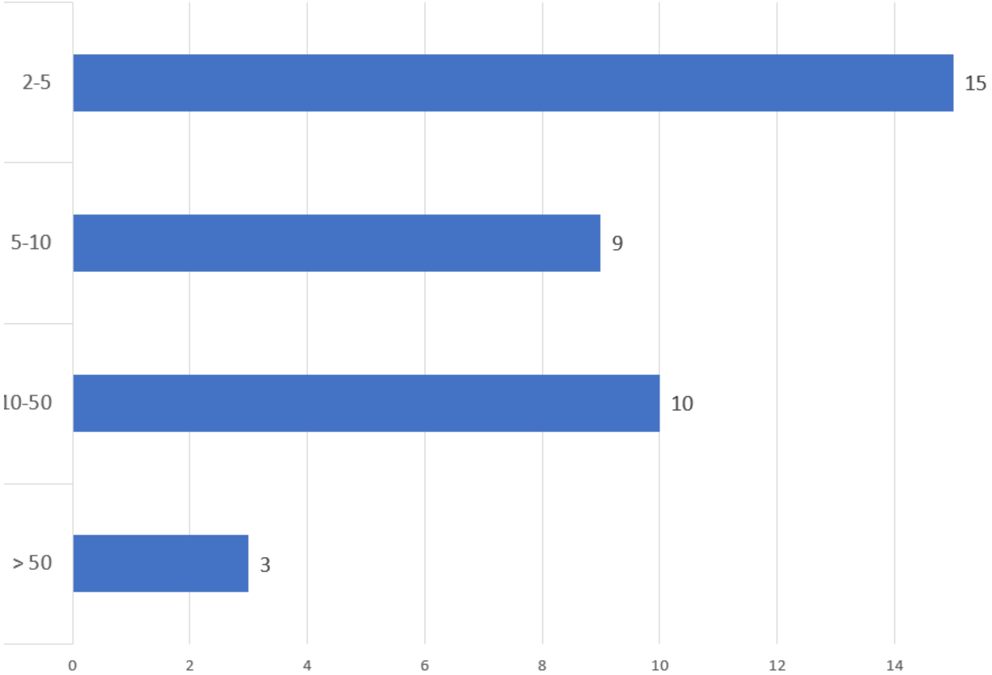


Figure 6. Responses to (Q3) What is the size of your project team?

4.1.2 Demographic Information (personal & company)

The demographic information about the respondents and their company was collected towards the end of the survey (questions 19 to 24) but is presented here before sharing the responses to the cybersecurity questions.

Figure 7 summarizes the responses to the question 19. Most of the respondents have taken on multiple project roles throughout their careers. There are 93 responses from the 37 respondents, giving an average of 2.5 different roles per respondent, besides their current role. The most common role assumed by the respondents is the developer role.

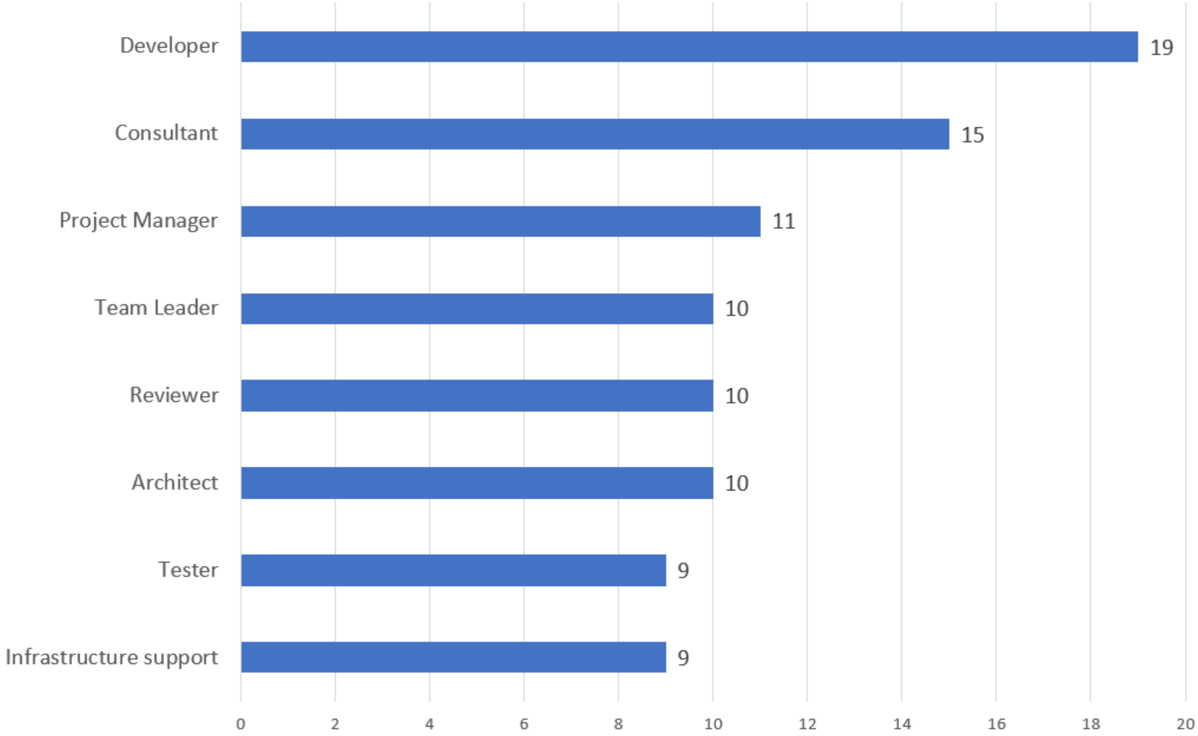


Figure 7. Responses to (Q19) Besides the current project, what roles have you played in other IT projects?

Figure 8 summarizes the responses to the question 20. More than half (22 out of 37) of the respondents have more than 10 years of IT experience, indicating quite an experienced pool of respondents.

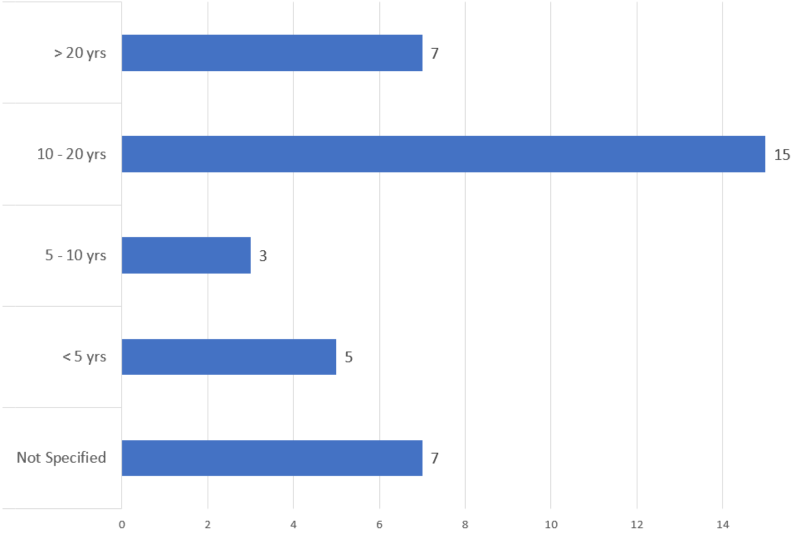


Figure 8. Responses to (Q20) How many years of IT experience do you have?

Figure 9 summarizes the responses to the question 21. The majority of the respondents work in large enterprises.

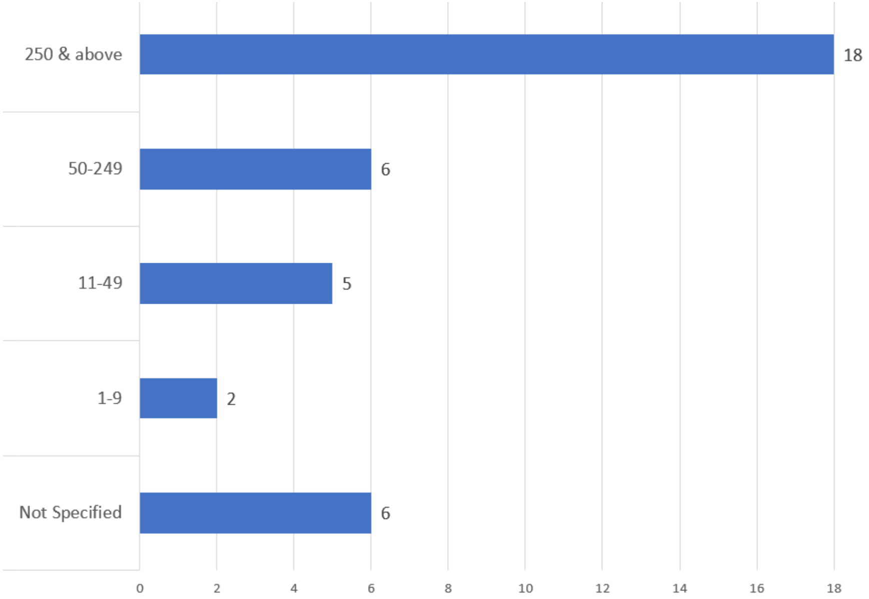


Figure 9. Responses to (Q21) What is the size (i.e. number of employees) of your company?

Figure 10 summarizes the responses to the question 22. The majority of the respondents work in MNCs and the next largest group of respondents work for the government/public sector.

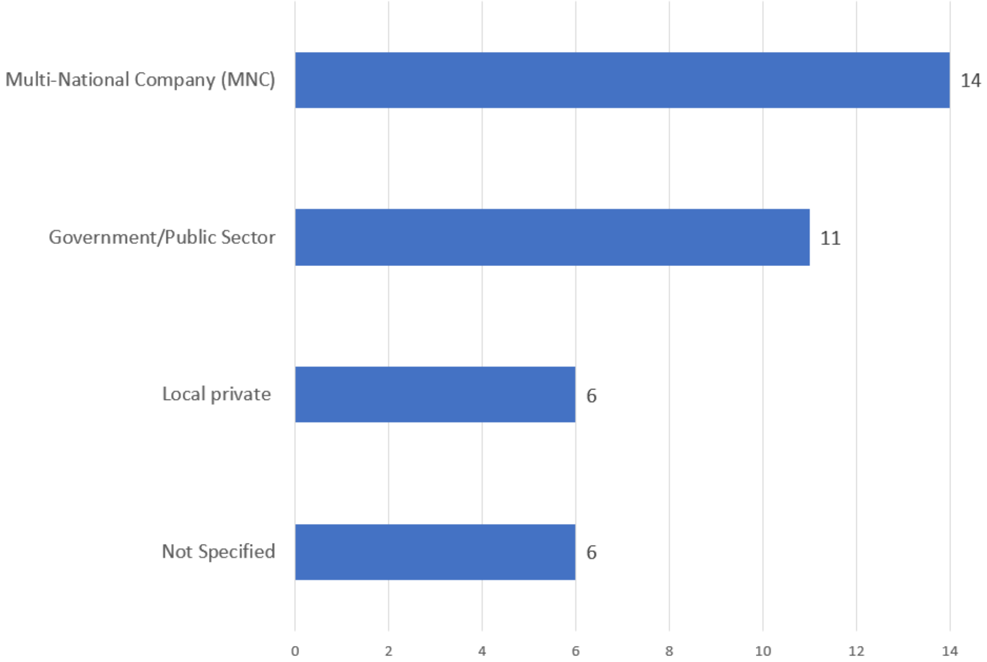


Figure 10. Responses to (Q22) Which category does your company fall under?

Figure 11 summarizes the responses to the question 23. The majority of the respondents work in the IT industry and in Defence/National Security.

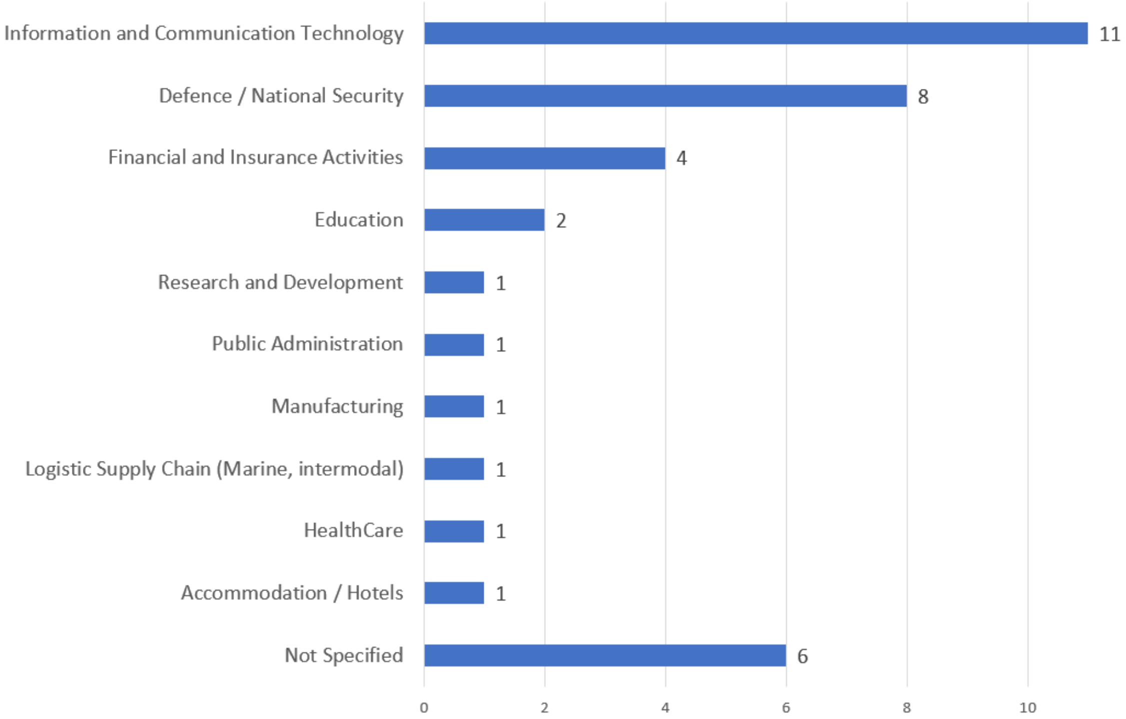


Figure 11. Responses to (Q23) Which industry is your company in?

Figure 12 summarizes the responses to the 20th question. The majority of the respondents are based in Singapore and the Netherlands.

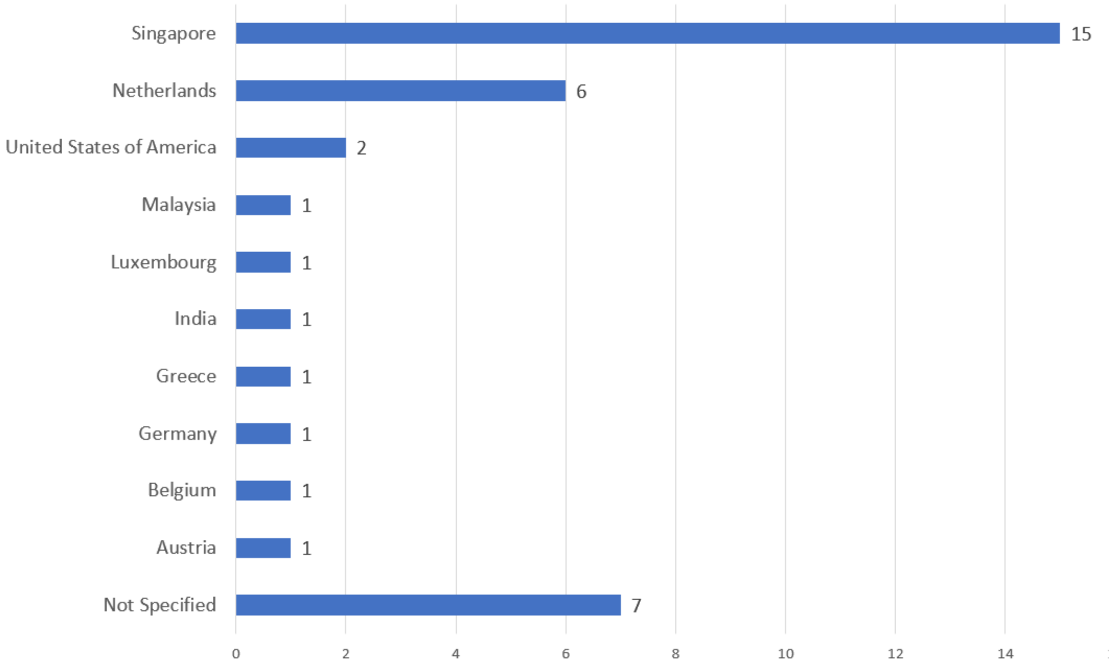


Figure 12. Responses to (Q24) Which country is your office in?

4.1.3 Cybersecurity Questions

The responses to the cybersecurity questions (4 to 18) are presented to in this section. Figure 13 summarizes the responses to the 4th question. About 27% (10 out of the 37) of the respondents indicated that they are aware of their company or project having encountered a cyber-attack on software that was developed or deployed.

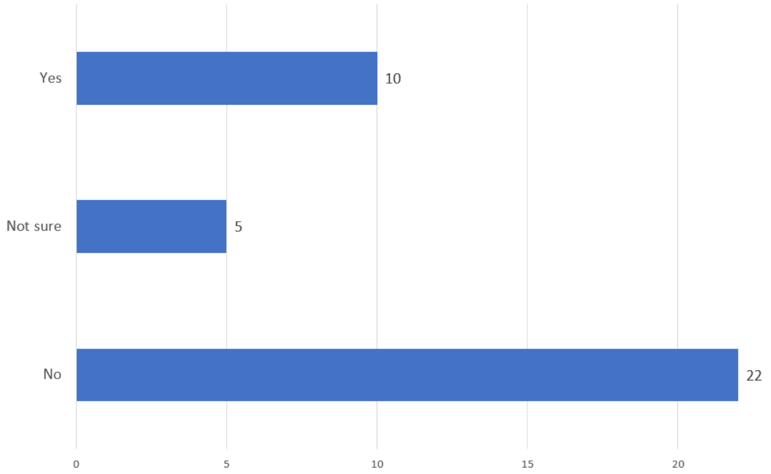


Figure 13. Responses to (Q4) Has your company/project ever experienced a cyber-attack related to software that was developed or deployed?

The 5th question was, “do you think adopting secure practices or a secure software development framework helps to prevent or reduce the occurrence of cyber-attacks?” All 37 respondents replied with a “Yes”.

Figure 14 summarizes the responses to question 6. Based on the responses, most of the companies have a cybersecurity team or a chief information security officer (CISO), being overall responsible for the security of the software developed. However, 10 out of 71 (about 14%) of the responses indicated there is no specific person responsible for the software security.

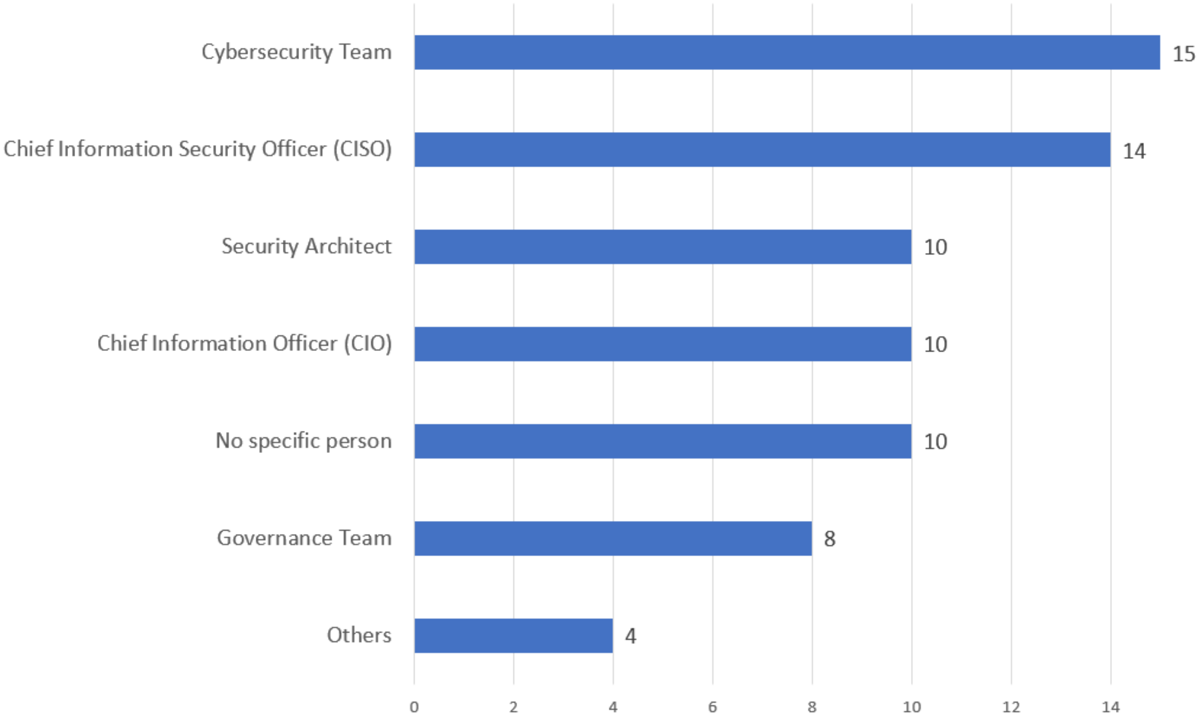


Figure 14. Responses to (Q6) Who in the company is overall responsible to ensure that developed/deployed software conform to the framework?

Figure 15 summarizes the responses to question 7. Microsoft Secure Development Lifecycle (SDL), OWASP's CLASP and NIST 800-160 on Systems Security Engineering are the more well-known frameworks among the participants. 4 of them indicated that they were unaware of any such frameworks. Among the 2 who indicted "Others", 1 did not specify while the other mentioned an initiative known as "Sheltered Harbour". Sheltered Harbour (2022) is not a secure development framework. It is, in fact, an industry-led initiative (prevalent in the United States) comprising financial institutions, core service providers and solution providers to enhance the stability and resiliency of the financial sector. It comprises infrastructure components and standards that help financial institutions ensure their business continuity in the event of a major disaster or incident like a cyber-attack.

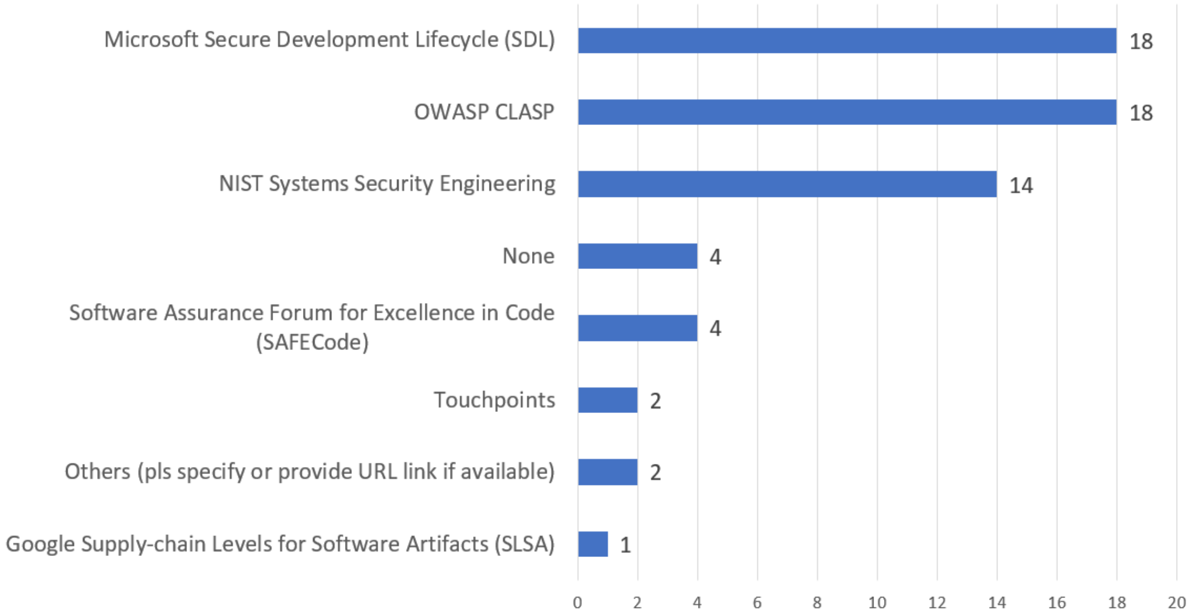


Figure 15. Responses to (Q7) What secure software development life cycle or secure software engineering (SSDLC/SSE) frameworks have you heard of?

Figure 16 summarizes the responses to question 8. Most of the participants indicated that their project used a custom framework, more than any of the published ones. 5 of them do not use any framework and one under “Others”, mentioned “Sheltered Harbour” as in question 7. The total number of responses is more than the valid respondents as some of them adopt more than a single framework.

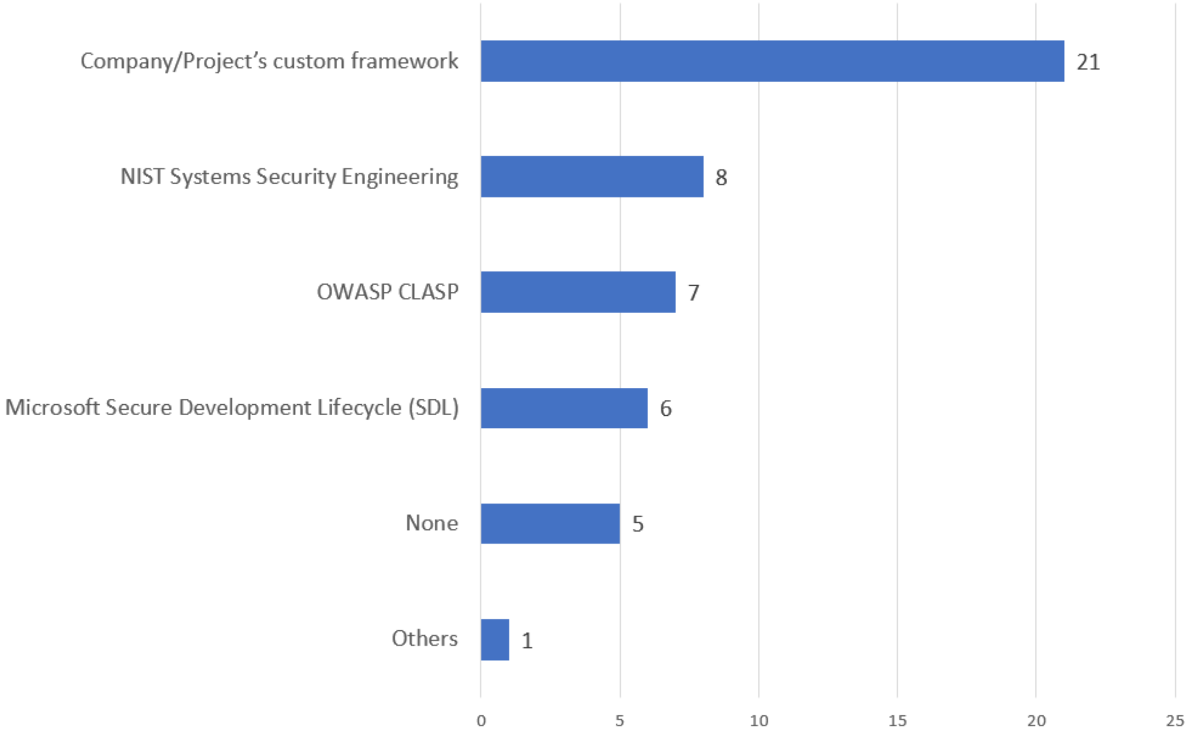


Figure 16. Responses to (Q8) What SSDLC/SSE framework is your project currently using?

Figure 17 and 18 summarizes the responses to question 9. Most of the respondents indicated that their projects carry out several secure development activities. Design review, code review and performing security risk assessment are the most common activities. This observation is similar for all responses, as well as, responses related only to custom frameworks. One respondent indicated, in verbatim, “running vulnerability scanner and fixing found CVEs” as a comment under “Others”.

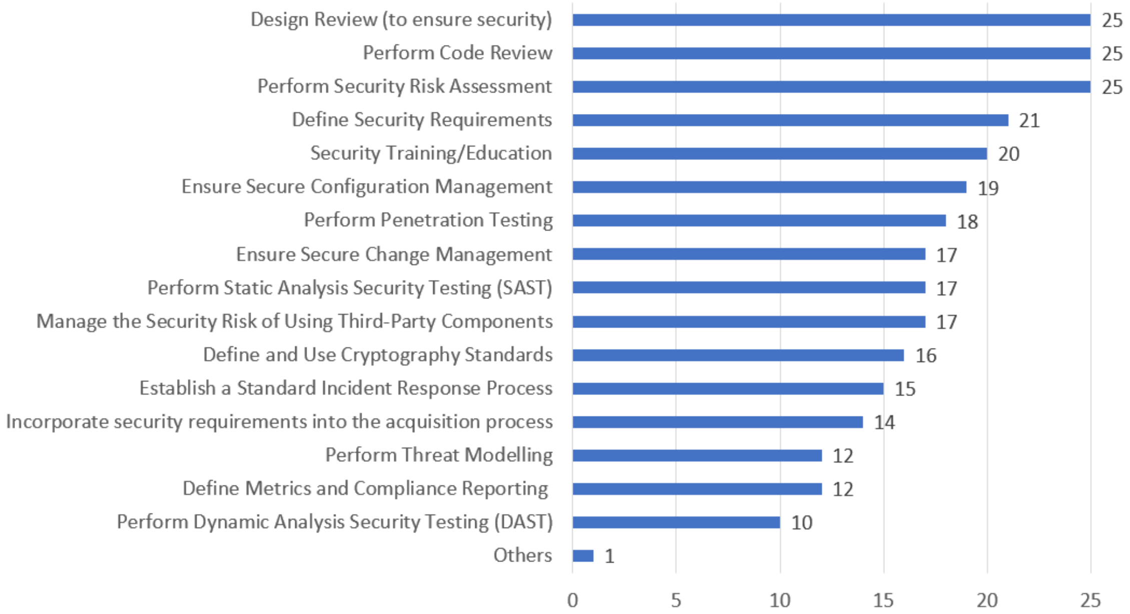


Figure 17. Responses to (Q9) What SSDLC stages/activities does your project follow? (all responses)

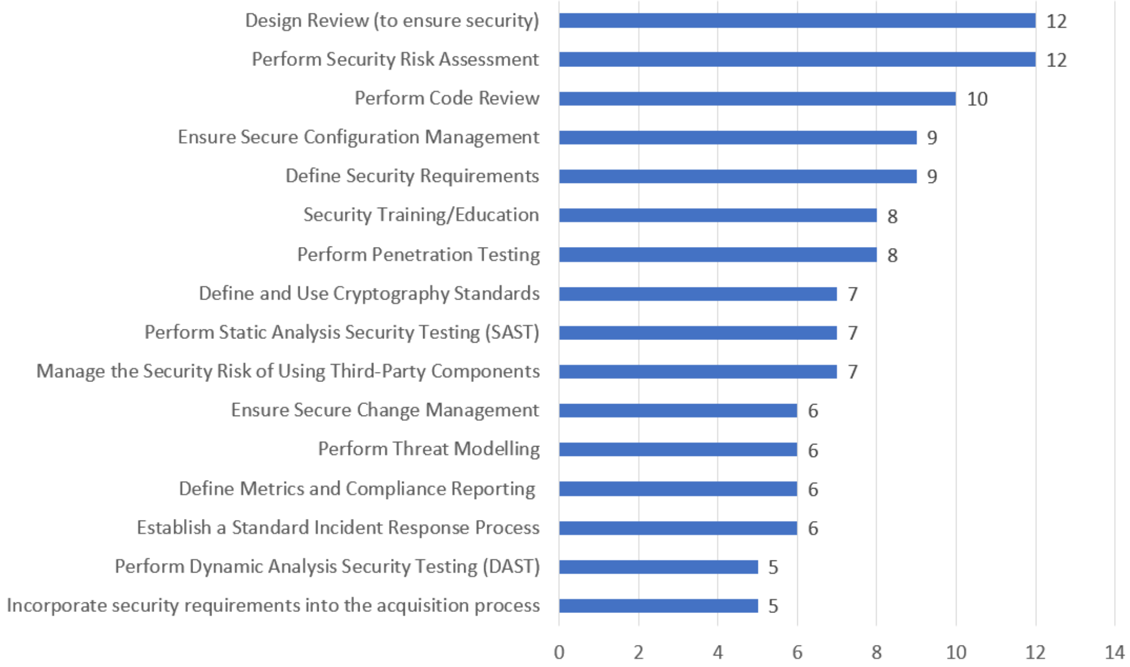


Figure 18. Responses to (Q9) What SSDLC stages/activities does your project follow? (responses only related to custom frameworks)

Figure 19 and 20 summarizes the responses to question 10. The respondents cited varied reasons for them adopting the secure development frameworks with a an even spread across the few options provided. The most cited reason is that the framework selected is comprehensive enough for their project size. The top few reasons cited are similar, taking into account all responses or responses related only to custom frameworks. One respondent indicated under "Others", the reason "company go live requirement".

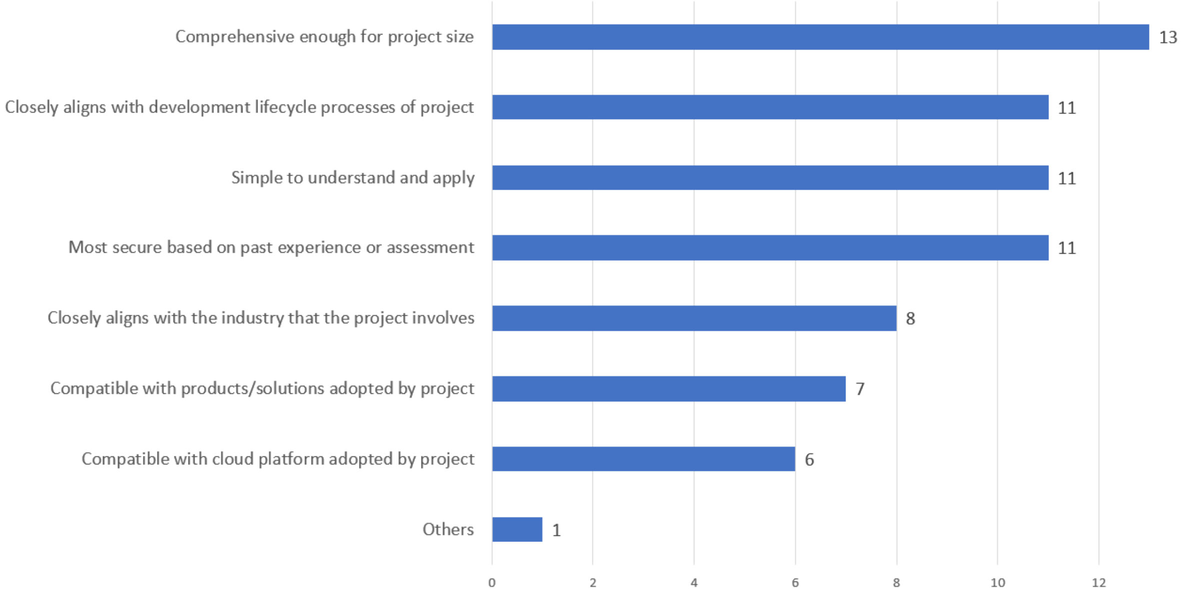


Figure 19. Responses to (Q10) Can you explain the reason(s) for your project adopting the mentioned framework? (all responses)

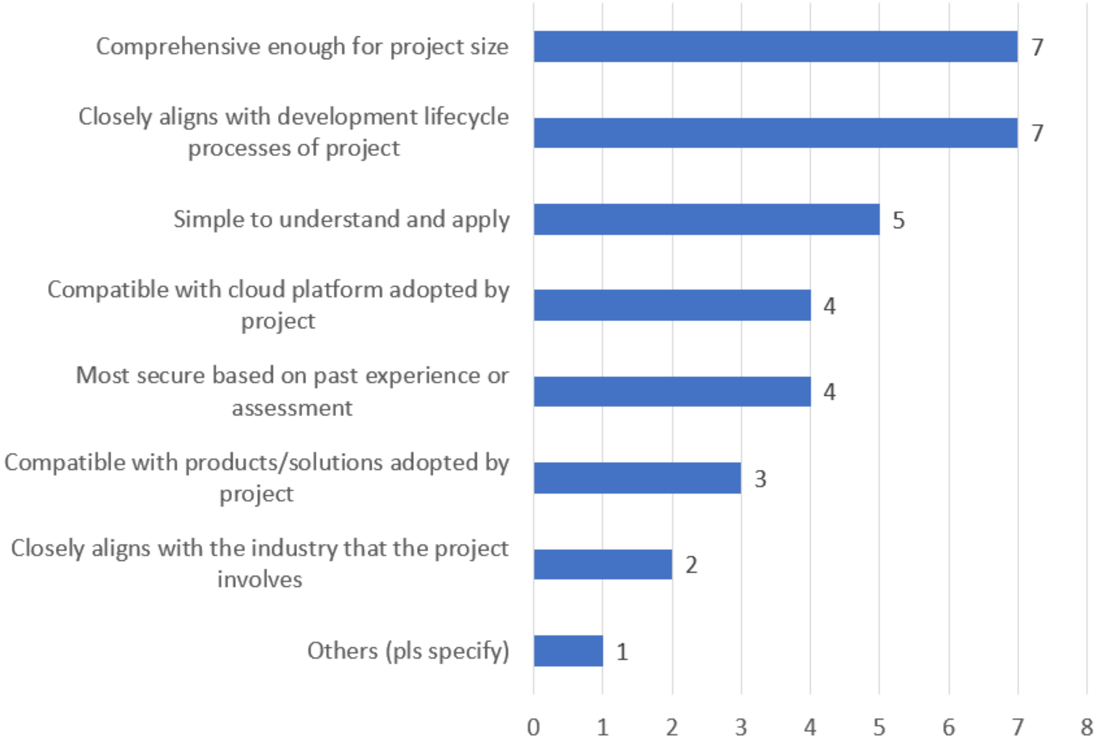


Figure 20. Responses to (Q10) Can you explain the reason(s) for your project adopting the mentioned framework? (responses only related to custom frameworks)

Figure 21 summarizes the responses to question 11. Among the 37 valid respondents, 7 did not select any option and one of them indicated "N/A" (not applicable) in the free text input. On hindsight this interview question should have included an option which explicitly mentions something to the effect of "no benefit observed".

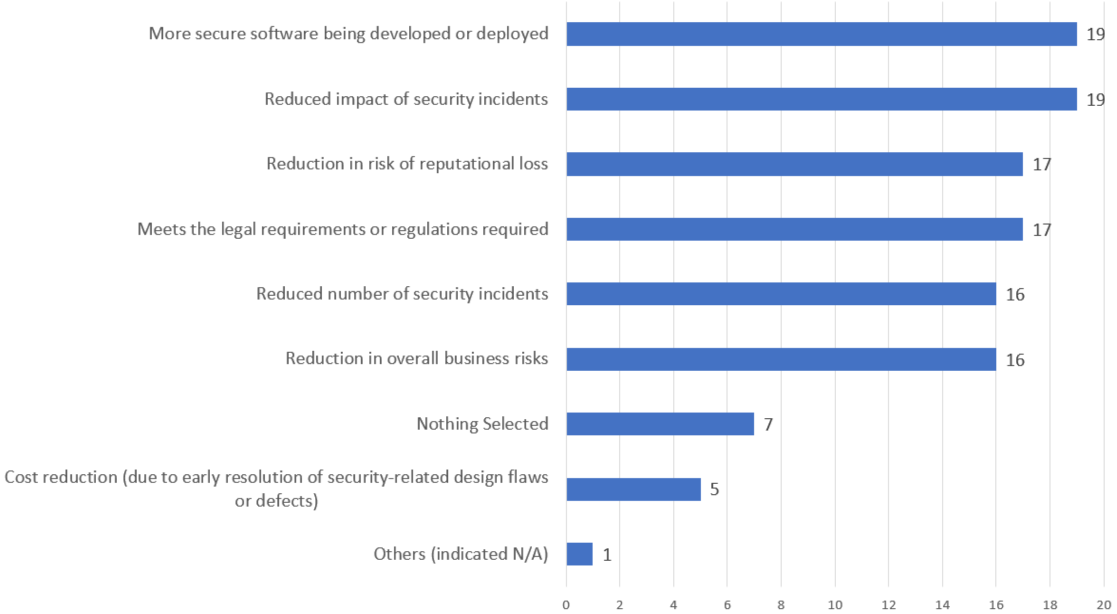


Figure 21. Responses to (Q11) Which of the following benefit(s) have you observed after adopting the mentioned framework?

Figure 22 summarizes the responses to question 12. Most of the respondent feel that it is necessary to adopt one of these security frameworks, while a small minority feel that it is not necessary. This could be due to, perhaps, the notion that adopting the necessary secure development activities will suffice instead of a complete framework.

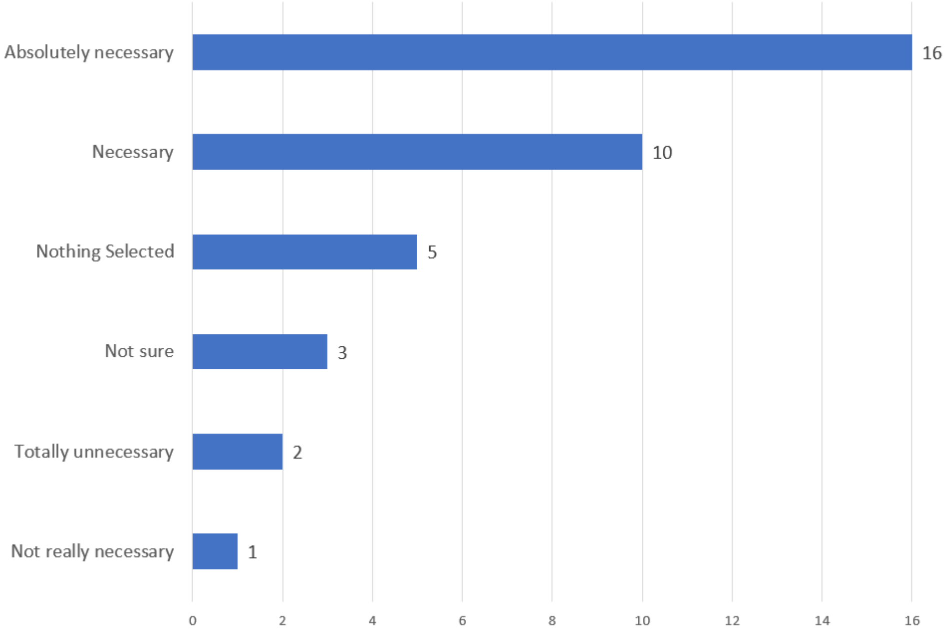


Figure 22. Responses to (Q12) How necessary do you think adopting one of these frameworks is?

Figure 23 summarizes the responses to question 13. Most of the respondents (22) indicated that their projects either fully or strictly enforced the security process while a small minority indicated that they were not or only lightly enforced.

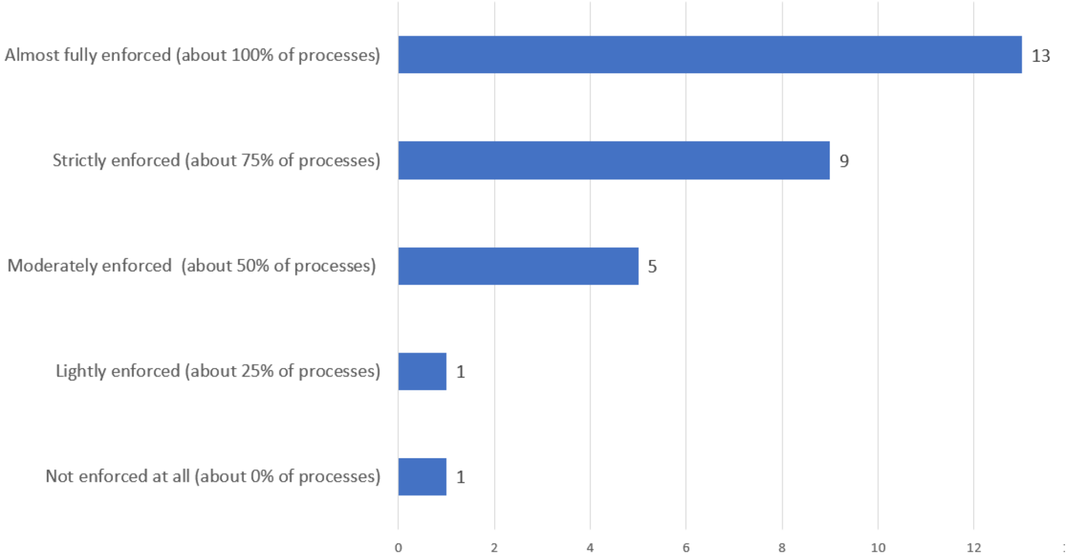


Figure 23. Responses to (Q13) How strictly does your project enforce the security processes of this framework?

Figure 24 summarizes the responses to question 14. Majority of the respondents indicated that their project almost fully enforced the security processes. The 2 verbatim responses under “others” were “resource to check and ensure” and “ease of use products”.

The verbatim responses weren’t very precise or complete. However, taking the context of the question into consideration, it is more likely that they meant the following:

- The resources required to check and ensure (is lacking)
- The ease of use of the products to facilitate the processes (is poor)

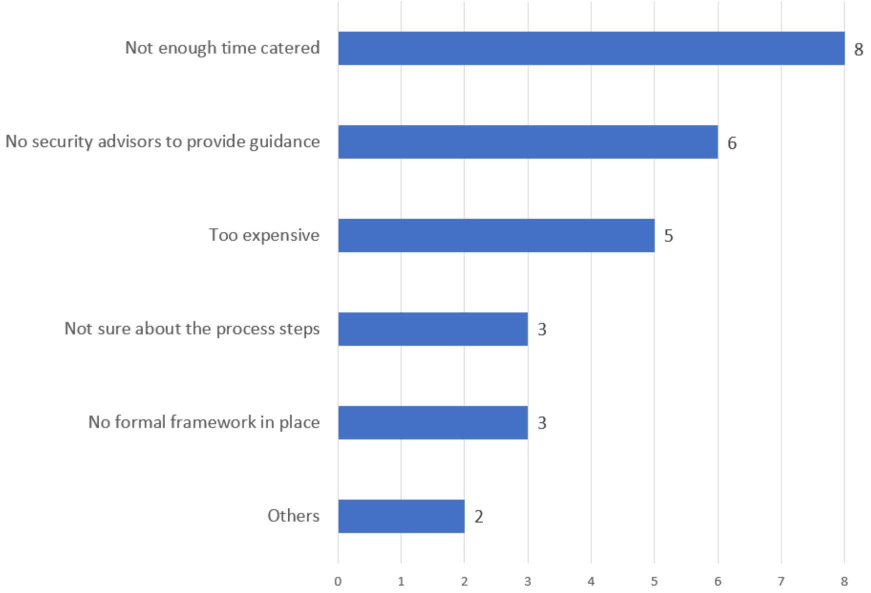


Figure 24. Responses to (Q14) What are the possible reasons for the framework NOT being “almost fully enforced”?

Figure 25 summarizes the responses to question 15. A significant number of the respondents reflected that their company started adopting a secure development framework more than three years ago.

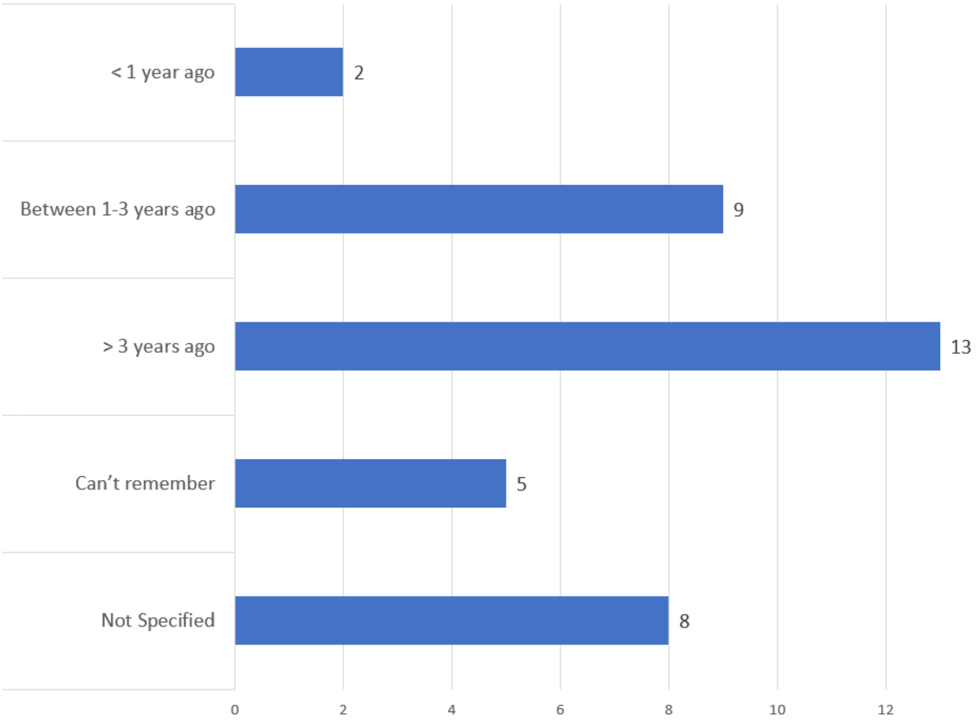


Figure 25. Responses to (Q15) When did your company start to adopt a secure software development life cycle or secure software engineering (SSDLC/SSE) framework?

Figure 26 summarizes the responses to question 16. The majority of the respondents indicated that they thought that there is indeed a greater need to adopt a secure development framework as companies increasingly rely on remote working.

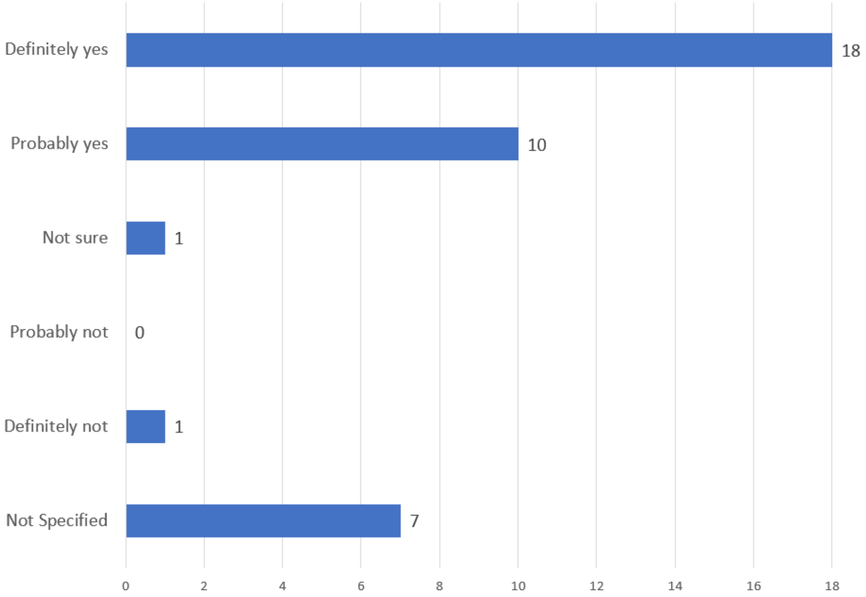


Figure 26. Responses to (Q16) Do you think there is a greater need to adopt an SSDLC/SSE framework as companies increasingly rely on remote working?

Figure 27 summarizes the responses to question 17. Participants were asked if they had been any changes or additions to the framework that was adopted in the last 2 years. They were also asked to briefly described the changes if there was any. Among the 10 respondents who indicated that there was some change to their adopted framework, 7 provided details about the changes that are summarized in the following few points:

- The framework was changed to account for increased compliance audit and stricter checks
- The framework was changed for specific vulnerability mitigation (e.g. ransomware and a Java-based logging utility, Log4j)
- The framework was changed to refine the process within
- The framework was updated due to an evolution of technology but is still the same

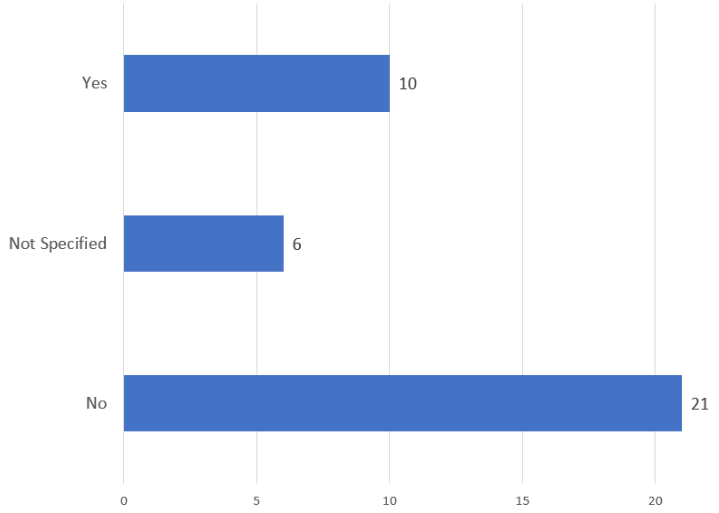


Figure 27. Responses to (Q17) Has there been any changes or additions to the framework that has been adopted in the last 2 years?

Participants were also asked what the reasons for the changes are. Figure 28 shows the reasons cited for any changes or additions. Based on the responses, the bulk of the changes or additions were due to new legal/government regulations.

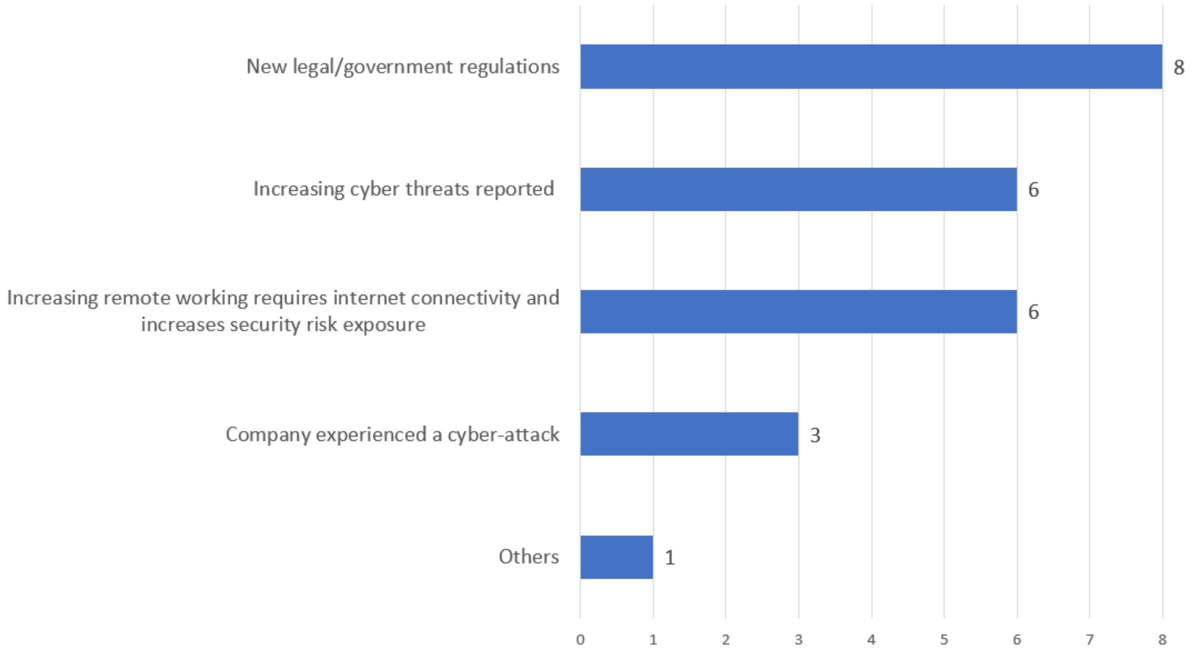


Figure 28. Responses to (Q18) What are the reasons for the changes?

4.1.4 Key Survey Findings

From these results, the key findings from the survey are summarised in the next few paragraphs with respect to the research questions.

Research Question 1:

A number of responses (44%) from the respondents indicated that their project used a custom framework, more than any of the cited frameworks (NIST 800-160, Microsoft SDL and OWAP CLASP). Some respondents (10%) indicated that no framework was adopted. Most of the respondents indicated that their projects either fully enforced (44%) or strictly enforced (31%) the security process, while the rest (about 24%) indicated that they were not or only lightly enforced. The main reasons cited for not fully enforcing the processes are insufficient time catered, the lack of guidance or that it is too expensive to do so.

Research Question 2:

The respondents cited varied reasons for them adopting the secure development frameworks with an even spread across the few options provided. The most cited reason is that the framework selected is comprehensive enough for their project size. Also frequently cited are reasons such as, it closely aligns with their development lifecycle processes or it is simple to understand and apply. The most cited benefits observed are that more secure software has been developed or that there is a reduced impact of security incidents.

Research Question 3:

The 3 most common reasons cited by the respondents for adopting custom frameworks are:

- They are comprehensive enough for their project size
- They closely align with the development lifecycle processes of their project
- They are simple to understand and apply

Also, specific to organisations that use custom frameworks, the 3 most common secure development activities the respondents mentioned are:

- Design Review
- Perform Security Risk Assessment
- Perform Code Review

Research Question 4:

A significant number of the respondents reflected that their company started adopting a secure development framework more than three years ago. The majority of the respondents also indicated that they feel there is indeed a greater need to adopt a secure development framework as companies increasingly rely on remote working.

Out of the 37 respondents, 10 of them (27%) indicated that there have been changes to the adopted framework in the last 2 years. Some of the changes include framework amendments due to technology evolution, increased compliance audit, more refinement on the process and mitigation for security vulnerabilities. The 3 most common reasons for the changes are:

- new legal/government regulations;
- increasing cyber threats reported;
- increasing remote working requires internet connectivity and increases security risk exposure.

4.2 Interview Findings

Following the survey, participants of the survey were asked if they were willing to take part in the interviews. Among the 37 valid respondents, eventually only 3 of them participated in the interview. Another 6 interviewees were nominated based on personal connections and agreed to participate. In total, 9 IT professionals were interviewed. Three (3) were based in Singapore, 3 in the Netherlands, 2 in Luxembourg, and 1 in the UK. They are described in Table 4, which assigns an identifier (ID), P1 to P9, to each. The 5th interviewee is assigned 2 IDs (P5A and P5B) as he had recently joined a new employer and described the 2 recent projects from this former and current organization. To preserve their anonymity, "he", "his" or "him" will be used, regardless of gender, when referring to the interviewees individually, if required.

ID	Role	Based In	Org. Size	Industry	Type
P1	Product Owner	Netherlands	Small	Healthcare	Private Local
P2	Project Manager	Singapore	Large	Defence	Government
P3	Developer, Project Leader	Netherlands (Curacao)	Micro	Finance	Private Local
P4	Reviewer	Singapore	Large	Government	Government
P5A, P5B	Developer, SecDevOps	Luxembourg	Small	Banking, Human Resource	Private Local
P6	Capabilities Lead (i.e. Requirements Manager)	Singapore	Large	Technology	MNC
P7	Developer	Luxembourg	Large	IT	MNC
P8	Developer, Project Lead	Netherlands	Large	IT & Research	Non-profit

Table 4. Description of Interviewees

Coding and Inter-Coder Agreement

The interviews were recorded and manually transcribed. First, open coding was done to label the quotes of potential interest mentioned during the interviews. This has the effect of breaking up the interview transcripts into discrete parts for subsequent analysis. During coding, besides being open and unbiased in reading through the transcripts, software development and cybersecurity activities were specially looked out for. Finally, a total of 189 unique codes were derived and applied to 268 quotations in the 8 interview transcripts.

To ensure that the analysis of the data was reliable, the transcript for one of the interviews (the sample) was coded by an independent coder and the inter-coder agreement was measured. ATLAS.ti provides three options for measuring the inter-coder agreement – simple percent agreement, Holsti index and Krippendorff's family of alpha coefficients. As the simple percent agreement and the Holsti index do not take into account chance agreement, therefore the Krippendorff alpha coefficient was used as an indicator for scientific reporting.

A test coding was first done to ensure the stability of the semantic domains and codes defined, and to ensure that both coders understood the meaning of the codes. After the initial test coding, a final independent coding was done on the sample and the Krippendorff alpha measured was 0.778. As this value is close to the recommended value ($\alpha \geq 0.800$), we conclude that the coded data is reliable.

Interview Findings

Analysis of the codes and quotations in the interview transcripts yielded the findings described in the next few sections. The findings are presented grouped by research questions, which are addressed by one or more interview questions. Table 3 in the previous chapter shows how the research questions are mapped to the interview questions. The full set of semi-structured interview questions is shown in **Appendix 2**.

4.2.1 Knowledge and Attitude Toward Cybersecurity

Before asking the interviewees about their current practices, they were first asked what they understand about secure software development frameworks.

4 of the interviewees (P1, P2, P6, P8) could not describe or name any specific framework, although interviewee P8 did mention having heard of it.

> P8: "I have heard of it. But to be honest, I've never looked at it."

2 of the interviewees (P3, P7) managed to describe it but could not name any specific framework.

> P3: "The only one that I know is just the software security development lifecycle, which is like, requirements, design, implementation, testing and risk assessment."

> P7: "I think it's a framework, which ensures the development of our software will be secure. It will prevent the development from being attacked by other people."

2 of the interviewees (P4, P5) were able to describe it and managed to name some of the frameworks such as MS SDL, OWASP CLASP and NIST 800-160.

> P4: "they are security activities that fortify the application throughout the normal software development lifecycle"

> P5: "I think in general, secure software development is kind of a practice that tries to make sure that security of a piece of software is integrated at every level."

4.2.2 Current Practices and Reasons in the Adoption of Secure Frameworks

This section provides findings to three research questions:

- RQ1: What are the current practices regarding the adoption of secure software development frameworks?
- RQ2: What are the key considerations for organizations in deciding on the SSDLC frameworks and practices to adopt?"
- RQ3: For organisations that use custom frameworks, what are the reasons behind this, and which elements are common in these custom frameworks?

All of the interviewees indicated that their projects either follow a custom development framework or some secure development activities. None of them adopted a published framework in its entirety. Seven of the interviewees (P1, P2, P3, P6, P5B, P7, P8) expressed that their projects do not follow any specific secure development framework or they are unaware of it. The feedback from these seven interviewees are as follows and where applicable, words with brackets between the quotes are added for clarity and to provide context of the statement as part of the entire interview conversation:

- Two of the interviewees (P3 and P7) mentioned that their projects do not follow any specific framework and provided no reasons for that.
- P1 mentioned that his project does not follow any specific framework, however, they do adopt security practices as they are compelled to follow laws and regulations because of the patient data within their project's systems.
- P2 mentioned that his project, which uses COTS products, only adopts some secure activities to ensure that the software is safe to use. However, other projects in his company that develop in-house applications actually adopt a SSDLC framework as a reference and customise to make it more robust. They customise the framework as it is dependent on their business needs and which component of the framework fits into their criteria. Also due to increasing cyber-attacks, more activities have been added by the different industries, as well as, his company to ensure that the software is in a better position to be used.
- P5B explained that they have no policy in place currently because they are writing the policy and determining the rules for the next iteration of releases for their development.
- P6 pointed out that he believed the technical members in his team follow a secure development framework, although he is not aware of which framework nor the reasons for its adoption.
- P8 indicated that his project does not adopt any framework and provided an explanation for that - "No, this one (project) doesn't (adopt any framework). However, cybersecurity is of the utmost importance in this project. So from all the experience we have in the team with cybersecurity, we try to apply everything to the highest standards."

The remaining two interviewees (P4, P5A) indicated that their projects use a custom SSDLC framework:

- P4 indicated that his project uses a custom framework. In essence, their reason was to select the best activities that are reasonable from the standard frameworks and incorporate their own activities.
- P5A explained his project's practice and reason as such: "Security was always like a number one issue. It's something that's not optional. No written document saying this is the policy, but it was well understood that we were following it in test and development. So before we even started writing code, we would have the API specification written. Although we have no policy on our documentation, you can probably pull up the standard SSDLC and say, 'we did this and this and this.' If you compared us to a secure software development framework, we would probably match the definition of one of them. But we didn't have an official work document saying that we must carry out business this way."

All of the interviewees described varying methods and degrees of enforcement of the security processes that their project have adopted. Four of the interviewees (P2, P4, P6, P8) indicated strong enforcement in their processes, involving parties independent of the project team:

- P2 described a strict enforcement process whereby project team members are responsible for declaring all the information of the software security status of their project. At the same time, automated scanning with reports to show the status of the security analysis, will be review it by an independent reviewer to ensure that the system is safe for deployment. after deployment, there will be regular and automated scanning to ensure that the application that is safe for continued operation.
- P4 highlighted that their enforcement process includes two review milestones by independent reviewers to ensure that the process is being followed. There are also regular independent audits conducted.
- P6 mentioned that they have enforcement in the form of dedicated cybersecurity team reviews, which can take about a week. This entails the cybersecurity team reviewing the software developed by the project team and coming back with a list of detected vulnerabilities to be fixed. The project team will need to fix these vulnerabilities before being allowed to deploy the software.
- P8 described that they have a strict enforcement process that involves a check with penetration testing. They also utilize configuration management software to enforce security settings and the Git repository so everyone in the team can see when unauthorized changes were made.

Two of the interviewees (P1, P7) indicated some enforcement by internal reviews/control by the project team members:

- P1 mentioned that in this project, only 2 (the most senior) out of 10 developers have permission to bypass security requirements. And that is rarely done.
- P7 highlighted that their enforcement involves checking by internal reviewers from the project team.

Three of the interviewees (P3, P5A, P5B) indicated no enforcement at that point in time:

- P3 described that there is hardly any enforcement and the company depends on him as developer and project lead to take the initiative to implement secure practices.
- P5A declared that there is no enforcement but they rely on the team members to play their part and and they generally do, as they are working in a slow-moving environment where they can do repeated testing and development is focused on protecting people's privacy. He doesn't think it's a good idea to have an independent reviewer to do all the checks. Instead, it is better for everyone to be more security-minded and motivated to play their part to deliver a secure product.
- P5B declared that they they have no enforcement now as they are in the midst of coming up with the the security policies and putting in place the processes.

4.2.3 Common Practices Adopted

As the reasons have already been explained in the previous section (4.2.2), this section will cover the common elements or activities mentioned by the interviewees in the adoption of secure frameworks or activities. It will provide the findings to the last part of the research question:

- RQ3: For organisations that use custom frameworks, what are the reasons behind this, and which elements are common in these custom frameworks?

All the interviewees cited several secure development practices that their projects adopt. The number of activities cited by each interviewee ranged from 3 to 7. Table 5 lists the activities mentioned and their frequency. The table shows that some kind of security testing and code reviews are the most frequently adopted activities.

Activity	Frequency	Remarks
Testing	10	generic - 2 penetration - 4 black-box - 1 manual - 1 white-box - 1 automated - 1
Code review	6	generic - 1 automated - 2 manual - 2 static code analysis - 1
Authentication (i.e. using proper authentication techniques to access applications)	3	generic - 2 multi-factor - 1
OS hardening	2	
Patching	2	
Vulnerability scanning & assessment	2	
Design review	2	
Protect from insider threats	1	
API specification	1	
Security configuration and settings	1	
Audit logging	1	
Audits (by external auditors)	1	
Code obfuscation	1	
Fix security vulnerabilities	1	
Implement secure architecture	1	
Inaccessible from internet	1	
Secure coding practice	1	
Secure data deletion	1	
Test-driven development	1	
Threat risk assessment	1	
Use VPN Software	1	

Table 5. Secure Development Practices Mentioned In Interviews

4.2.5 Changes and Trends in Adoption of Secure Development Frameworks/Activities

This section provides findings to the last research question:

- RQ4: What are the historical trends in the adoption of secure software development frameworks? What are the major changes in the past, and what are the potential future developments?

All the interviewees except three of them (P3, P5B, P7) described some recent changes or additions to the secure development framework or activities that they have adopted. All the changes described were implemented to tighten security. These changes and the specific reasons behind them are:

- P1 desribed the introduction of a decentralized network, that is quite new and unique to the healthcare industry. The decentralized architecture mitigates the risks and reduces the impact of any potetial security breach. If one of the health care institutions encounters a security issue, then only one of the nodes in that network will be affected.
- P2 mentioned that the OWASP dependency checker was introduced to scan the software libraries that are used in their IT systems. The OWASP dependency checker is able to scan deployed open-source software libraries and highlight those with known vulnerabilities or have reached their end of support date.
- P4 commented they have additional standards that they have to comply with additional tests to carry out to take into account the increased utilization of mobile apps. Previously, only web-based apps were deployed. Additional standards and tests were also introduced to take into account the increased migration of applications to the public cloud.
- P5A indicated the increased adoption of high-fidelity testing in an environment that is as close to the customer's actually environment. his project also increasingly relied on centralised configuration to help with security using containers (like Docker).
- P6 indicated his company had recently switched to an apparently more secure VPN provider recently. They have also made privacy settings more strict.
- P8 mentioned that they are looking into Security Information and Event Management (SIEM) systems, that had the potential to reduce the demand on the team providing support for the systems, freeing them to work on other activities.

All of the interviewees described varying past and future trends in the adoption of secure software development frameworks and practices. Five of the interviewees (P2, P3, P4, P7, P8) forecasted an increasing awareness and/or adoption of security frameworks or activities:

- P2 forecasted that more companies will adopt security frameworks and practices because of the increased cyberattacks that have been happening in recent years. So security must be there in order to protect business operations and organization's data, in order to ensure that it is safe for customers to use their applications. This is even more important with the proliferation of new technologies like IoT, which greatly increases the attack surface of an organization's network.

- P3 observed that there many documentation about software development these days cover the security aspects as well. As such, awareness of secure software development is slowly growing.
- P4 observed a growing awareness of application security especially since the OWASP standard was published in 2005 and felt that that was the start of the (security) activities being instilled in the whole development lifecycle. For his organization, more applications are being moved from internal to being internet-facing so there is an increasing need to reduce the potential cyber-attacks and monitor the vulnerabilities of the numerous 3rd party components used. He projected that increasing digitalization and use of agile in development will tend to see business needs take precedence over security. Also software development is so prevalent and easily accessible to even the school kids, that we can expect an increase in the development of software applications and a corresponding rise in security challenges
- P7 observed that there had been more compliance with GDPR or country-specific regulations in recent years and predicted that there will be more of such regulations that his company have to comply with as they expands to serve customers in more countries. He also projected that they will be increasing policies and rules to be adopted in the future to strengthen the security practices, although it may not be lead to the increased adoption of entire secure development frameworks.
- P8 forecasted that there will a continued rise of cyber-attacks and threats and a corresponding increase of security activities that organization will have to keep up with.

Two of the interviewees (P5, P6) projected more security being built into tools that can help development teams deliver more secure software:

- P5 observed that there are a lot more specialised services/technology for security and 3rd party tools that build in security straightaway e.g. products from HashiCorp (like Vault, Consul) and Kubernetes etc.
- P6 projected that there is going to be a lot more model-driven and UI-based code development, where security is actually taken care of by the underlying tools used. When compiling or before software deployment, there will also be automated code security checks by the tools. There will also be increasing adoption of automated vulnerability detection (using AI) and some automated intrusion test.

Finally, P1 observed that that companies tend to ignore or not invest enough in good security until they encounter a security breach or issue that gets public attention. Business considerations tend to take precedence which makes security an after-thought. Cybersecurity solutions coming from either science or private companies tend not to be largely adopted and are kind of niche. There are solutions that work but they're not implemented on a large enough scale to make a difference.

5. Discussion

5.1 Discussion of Results

This section discusses the results and findings from both the survey and interviews and compares it with earlier research, where applicable.

5.1.1 Adoption of Frameworks

From this research it can be seen that the majority of the projects and companies are taking cybersecurity more seriously by adopting secure practices, though many are still not adopting a formal SSDLC framework. The findings from the survey and interviews yielded the following numbers:

- Survey: formal framework (43.8%), custom framework (43.8%), none/others (12.5%)
- Interview: formal framework (0%), custom framework (22.2%), none (77.8%)

The difference in percentage between the survey and interview findings could be due to the multiple choices provided in the survey which acted as pointers to help the respondents name the formal frameworks and claim that their projects adopt one of them. Without the options to choose from, the interviewees found it harder to recall the formal frameworks. For example, one of the interviewees pointed out that the technical members in his team follow a SSDLC framework but he is not aware which framework they adopted. Another interviewee (P8) mentioned during the interview, “I have heard of it. But to be honest, I’ve never looked at it.”

It should be noted that there is a fine line between projects that adopt a custom framework versus those who do not use a framework but select specific security activities. In essence, these two categories of projects do not adopt a formal framework. When grouped together, we find that those who do not adopt a formal framework, surpass those who do. Overall, this finding is in agreement with an earlier research that found that only about 30% of its survey participants said they use a formal SSDLC framework (Geer, 2010).

Other than NIST 800-160, OWASP CLASP and Microsoft SDL, none of the survey respondents or interviewees have adopted other formal frameworks. Neither has any of them heard of any other formal frameworks, including the Viewnext-UEx model mentioned by Núñez et al (2020). One of the survey respondents did mention an initiative known as “Sheltered Harbour” (Sheltered Harbour, 2022) which is not strictly a secure development framework.

5.1.2 Adoption Considerations

From both the survey and the interviews, it is clear that among those who do adopt a framework, the majority use a custom framework. The 3 most common reasons cited by the survey respondents for adopting the frameworks (standard or custom) are:

- They are comprehensive enough for their project size

- They closely align with the development lifecycle processes of their project
- They are simple to understand and apply

In addition, one of the interviewees (P4) explained that the reason for their project using a custom framework was to select the activities that are reasonable from the standard frameworks and also incorporate their own.

5.1.3 Enforcement of Processes

The other equally important finding is that although some of the survey respondents indicated that their projects fully enforced the security processes, the majority (55%) indicated that they were not fully enforced. The main reasons cited for not fully enforcing the processes were: insufficient time catered, the lack of guidance or that it is too expensive to do so.

From the interviews, we observed that 5 of the 9 interviewees reported less stringent enforcement via internal controls within the team or not at all. The other 4 interviewees describe stronger controls that exhibit some independence between the reviewers and the project team.

These reasons point to the fact that many projects are still torn between their efforts at security and their other pressing project objectives. This is similarly articulated by Assal et al (2019) which showed that many developers are motivated to develop secure software but are deterred when they have to deal with competing priorities or the lack of resources.

5.1.4 Common Activities Adopted

During the interviews, the following set of security-related activities were mentioned more than once (listed in descending order of frequency mentioned):

- testing (penetration, black-box, white-box etc.)
- code review
- using proper authentication
- operating system hardening (although not strictly considered software development)
- patching
- vulnerability scanning and assessment
- design review

The most common secure development activities (corresponding top 7 selected for comparison) cited in the survey were:

- design review
- security risk assessment
- code review
- define security requirements
- security training/education
- secure configuration management
- penetration testing

From the comparison, it is observed that design review, code review and penetration testing are the 3 most common secure development activities.

5.1.5 Changes and Trends

Some of the changes to the adopted framework mentioned in the survey are amendments due to technology evolution, increased compliance audit, more refinement on the process and mitigation for security vulnerabilities. The 3 most common reasons quoted for the changes are:

- new legal/government regulations
- increasing cyber threats reported
- increasing remote working requires internet connectivity and increases security risk exposure

Some of the changes and trends quoted by the interviewees are:

- new tools introduced to scan open-source software libraries for known vulnerabilities
- additional standards and tests to take into account increased utilization of mobile apps and increased migration of applications to the public cloud.
- made privacy settings more strict
- looking into Security Information and Event Management (SIEM) systems

The responses from the survey respondents were selected from the multiple choices provided while the interviewees highlighted the changes in their own words, hence they seem quite different. However, some correlation can observed between the reasons highlighted in the survey and the changes mentioned by the interviewees. For example, new legal/government regulations may be a likely reason for the making of privacy settings more strict. Also, increasing cyber threats reported is a potential driver for new tools to scan software libraries for vulnerabilities and the need to look into deploying SIEM systems.

5.2 Limitations

This research has faced several limitations such as:

1. Convenience sampling. Sampling is a major challenge when conducting surveys. The survey and interview invitations were extended to known contacts and “contacts of contacts”. It was challenging to obtain a good random and representative sample of the target population in this research. Another constraint of the convenience sampling is that it was difficult to get a bigger sample size and recruit participants from more diverse countries/industries. Most of the respondents were from Singapore and the Netherlands due to the author’s personal connections and network. As a result of this skewed sample, external validity and generalizability of the findings cannot be ascertained for this research.

2. Quality of the online survey. Much more information could have been collected. However, a balance was struck between the amount of information gathered and the

survey completion time. Any additional desired information not obtained during the survey was elicited from the participants during the subsequent interviews.

3. Validity of survey questions. Another challenge concerns the construct of the questions and the pre-selection of frameworks as selection options. To mitigate any potential issues, pilots of the survey were conducted to ensure its clarity and to find possible defects and an “other” option were included for participants to fill in other frameworks. Based on these pilots, the survey was enhanced before its publication.

4. During the interviews, some interviewees provided responses that may be related but do not really answer the question in a satisfactory manner e.g. when the question was about trends in the adoption of security frameworks and practices, some of the interviewees mentioned closely-related cybersecurity trends instead.

6. Conclusion

This study has looked mainly into the adoption of secure software development frameworks. Starting by first reviewing prior studies in this area, a survey and interviews were subsequently conducted to discover the most current adoption practices and the considerations behind them. It has also looked at the activities prescribed by 6 formal SSDLC frameworks and observed that the majority of organizations has still not adopted one of these frameworks in its entirety.

Not every organization believes in adopting a formal framework. Most would rather adopt a custom framework or choose a few key secure activities in their software development. Adding to the complication is the fact that all the 6 formal frameworks recommend different activities as summarized in Table 1. Though MS SDL, CLASP, Touchpoints and NIST 800-160 are the most comprehensive across all stages of the software development lifecycle, they are not as comprehensive as SLSA in ensuring integrity of the software supply chain, which is a critical area to protect. As discovered by Geer (2010) and again in this research, customizing a framework or selecting some secure activities is still a sensible option for organizations with varying resources and specific needs to address the continuously emerging cyber threats.

Another research with software developers has shown that many of them are motivated to develop secure software but are deterred when they have to deal with competing priorities or the lack of resources (Assal et al, 2019). A crucial conclusion from that research and this, is that when it comes to security, organizations need to have a clear objective and consistent message to the IT professionals in their projects. It will be futile to adopt the most comprehensive SSDLC framework or set of security activities but not conscientiously enforce the processes prescribed. A more lightweight but closely enforced regiment in software security might instead yield more effective outcomes. A research by Weir, Becker, et al (2020) that experimented with a series of lightweight interventions to improve development teams' awareness and their motivation to improve software security showed that the interventions can have a long term beneficial effect on the security of the software developed. These techniques have the potential to improve software security on a wider scale.

One potential approach to improve the adoption rate of formal SSDLC frameworks is to build in some options within the framework that proposes the secure activities to be executed, based on the risk assessment of the projects adopting it. Along with the options, an appropriate risk assessment framework that measures factors such as the system exposure and the "impact if compromised" needs to be developed. Lower-risk projects can then select the option with more lightweight interventions, while high-risk projects need to follow the option with the most stringent set of secure development activities. These should be formulated by knowledgeable practitioners, and based on research and empirical evidence; rather than leaving it to individual organizations, who may not have the expertise, to casually decide based on convenience.

6.1 Future Research

From the challenges encountered while putting this thesis together and taking the limitations into account, the following points ought to be considered for further research:

- The survey and interview can be repeated with a more random and representative sample of the target population. Also more participants and from more diverse countries/industries should be recruited for the study.
- Besides the project team members and developers, the study can be expanded to capture the perspectives of the senior management or C-level executives of organizations e.g. CIO, CISO etc.
- Besides a survey and interviews, other methods like observation and delving into the codes and artefacts can be done to provide empirical evidence that can be compared with the perceptions of the survey respondents and interviewees.

References

Assal, H., & Chiasson, S. (2019). Think secure from the beginning': A Survey with Software Developers. CHI '19: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, 289, 1-13. https://doi.org/10.1145/3290605.3300519

Bromiley, M. (2019, August). SANS 2019 Incident Response (IR) Survey: It’s Time for a Change. SANS.

Canalys. (2021, March). Now and next for the cybersecurity industry – part 1.

Centre for Cyber Security. (2021). SolarWinds: State-sponsored global software supply chain attack. (1st ed.)

Cyber Security Agency of Singapore. (2017, Nov). Security-by-Design Framework (Version 1.0). https://www.csa.gov.sg/legislation/supplementary-references

De Win, B., Scandariato, R., Buyens, K., Grégoire, J., & Joosen, W. (2009). On the secure software development process: CLASP, SDL and Touchpoints compared. Information and Software Technology, 51(7), 1152-1171. https://doi.org/10.1016/j.infsof.2008.01.010

Forrester. (2021). Beyond Boundaries: The Future Of Cybersecurity In The New World Of Work.

Gasiba, T. E., Lechner, U., Pinto-Albuquerque, M., & Mendez Fernandez, D. (2020). Awareness of secure coding guidelines in the industry - A first data analysis. 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 345-352. https://doi.org/10.1109/TrustCom50675.2020.00055

Geer, D. (2010). Are Companies Actually Using Secure Development Life Cycles? Computer, 43(6), 12–16. https://doi.org/10.1109/MC.2010.159

Howard, M., & Lipner, S. (2006). The Security Development Lifecycle. Microsoft Press.

Huang, Y. W., Yu, F., Hang, C., Tsai, C. H., Lee, D. T., & Kuo, S.Y. (2004). Securing Web Application Code by Static Analysis and Runtime Protection. https://doi.org/10.1145/988672.988679

Kirlappos, I., Beautement, A., & Sasse, M. A. (2013). "Comply or Die" Is Dead: Long live security-aware principal agents. Financial Cryptography and Data Security, 7862, 70-82. https://doi.org/10.1007/978-3-642-41320-9_5

McGraw, G. (2006). Software Security: Building Security In. Addison Wesley.

Myagmar, S., Lee, A. J., & Yurcik, W. (2005). Threat modeling as a basis for security requirements. In: Symposium on Requirements Engineering for Information Security (SREIS).

Núñez, J. C. S., Lindo, A. C., & Rodríguez, P. G. (2020). A Preventive Secure Software Development Model for a Software Factory: A Case Study. IEEE Access, 8, 77653-77665. https://doi.org/10.1109/access.2020.2989113

OpenSSF. (2022). Safeguarding artifact integrity across any software supply chain. https://slsa.dev/

OWASP. (2006). Comprehensive, lightweight application security process. http://www.owasp.org

Rauf, I., Petre, M., Tun, T., Lopez, T., Lunn, P., Van der Linden, D., Towse, J., Sharp, H., Levine, M., Rashid, A., & Nuseibeh, B. (2022). The Case for Adaptive Security Interventions. ACM Transactions on Software Engineering and Methodology, 31(1), 1-52. https://doi.org/10.1145/3471930

Rials, W. (2021). Top Cybersecurity Trends for 2021 and Beyond. Homeland Security Affairs: Pracademic Affairs 1, Article 3. www.hsaj.org/articles/17153

Ross, R., McEvilley, M., & Oren, J. (2016). NIST Special Publication 800-160, Volume 1. Systems Security Engineering: Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems. National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-160v1

Shah, S., & Mehtre, B. M. (2015). An overview of vulnerability assessment and penetration testing techniques. Journal of Computer Virology and Hacking Techniques, 11, 27–49. https://doi.org/10.1007/s11416-014-0231-x

Sheltered Harbour Organization. (2022). Sheltered Harbour. https://shelteredharbor.org/

Software Assurance Forum for Excellence in Code (SAFECode). (2018). Fundamental Practices for Secure Software Development. (3rd ed.)

Souppaya, M., Scarfone, K., & Dodson, D. (2022). NIST Special Publication 800-218. Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities. National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-218

Tessian. (2021). Back to Work: Security Behaviors Report.

Weir, C., Becker, I., Noble, J., Blair, L., Sasse, M.A., & Rashid, A. (2020). Interventions for long-term software security: Creating a lightweight program of assurance techniques for developers. Software: Practice and Experience. 50(3), 275-298. https://doi.org/10.1002/spe.2774

Weir, C., Hermann, B., & Fahl, S. (2020). From Needs to Actions to Secure Apps? The Effect of Requirements and Developer Practices on App Security. USENIX Security Symposium.

Weir, C., Migues, S., Ware, M., & Williams, L. (2021). Infiltrating Security into Development: Exploring the World's Largest Software Security Study. Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. 1326–1336. https://doi.org/10.1145/3468264.3473926

Wolf, F., Aviv, A. J., & Kuber, R. (2021). Security Obstacles and Motivations for Small Businesses from a CISO's Perspective. 30th USENIX Security Symposium (USENIX Security 21), 1199--1216.

Appendix 1 – Survey Questions

Start of Block: Project Information

Section 1 Project Information

The next few questions relate to the project that you are currently working on or the most recent project done within the last 2 years.

Q1 What is your role in the project?

- □ Developer
- □ Tester
- □ Reviewer
- □ Consultant
- □ Architect
- □ Infrastructure support
- □ Team Leader
- □ Project Manager
- □ Others (pls specify) ____

Q2 What phases of software lifecycle does your project cover (regardless of whether for a new system or an existing one)?

□ Exploration/Pilot/Concept

□ Acquisition

□ Design

□ Development

□ Operations/Production

□ Retirement/Decommissioning

□ Others (pls specify) ____

Q3 What is the size (i.e. number of team members) of your project team?

○ 1

○ 2-5

○ 5-10

○ 10-50

○ > 50

End of Block: Project Information

Start of Block: Cyber Questions

Section 2 Cybersecurity Questions

The next few questions relate to your encounters and experience with cybersecurity matters in your project and/or company.

Q4 Has your company/project ever experienced a cyber-attack related to software that was developed or deployed?

○ Yes

○ No

○ Not sure

Q5 Do you think adopting secure practices or a secure software development framework helps to prevent or reduce the occurrence of cyber-attacks?

○ Yes

○ No

○ Not sure

Q6 Who in the company is overall responsible to ensure that developed/deployed software conform to the framework?

□ Chief Information Officer (CIO)

□ Chief Information Security Officer (CISO)

□ Cybersecurity Team

□ Governance Team

□ Security Architect

□ No specific person

□ Others (pls specify) ____

Page Break

Q7 What secure software development life cycle or secure software engineering (SSDLC/SSE) frameworks have you heard of?

□ Microsoft Security Development Lifecycle (SDL)

□ NIST Systems Security Engineering

□ Google Supply-chain Levels for Software Artifacts (SLSA)

□ OWASP Comprehensive, Lightweight Application Security Process (CLASP)

□ Software Assurance Forum for Excellence in Code (SAFECode)

□ Touchpoints

□ Others (pls specify or provide URL link if available)

____

Q8 What SSDLC/SSE framework is your project currently using?

□ Company/Project’s custom framework

□ Microsoft Security Development Lifecycle (SDL)

□ NIST Systems Security Engineering

□ Google Supply-chain Levels for Software Artifacts (SLSA)

□ OWASP Comprehensive, Lightweight Application Security Process (CLASP)

□ Software Assurance Forum for Excellence in Code (SAFECode)

□ Touchpoints

□ Others (pls specify or provide URL link if available)

____

Page Break

Q9 What SSDLC stages/activities does your project follow?
(Please select all that applies)

□ Security Training/Education

□ Define Security Requirements

□ Define Metrics and Compliance Reporting

□ Perform Security Risk Assessment

□ Perform Threat Modelling

□ Design Review (to ensure security)

□ Incorporate security requirements into the acquisition process

□ Define and Use Cryptography Standards

□ Manage the Security Risk of Using Third-Party Components

□ Perform Static Analysis Security Testing (SAST)

□ Perform Dynamic Analysis Security Testing (DAST)

□ Perform Code Review

□ Perform Penetration Testing

□ Ensure Secure Configuration Management

□ Ensure Secure Change Management

□ Establish a Standard Incident Response Process

□ Others (pls specify) ____

Page Break

Q10 Can you explain the reason(s) for your project adopting the mentioned framework? (Please select all that applies)

□ Simple to understand and apply

□ Comprehensive enough for project size

□ Compatible with cloud platform adopted by project

□ Compatible with products/solutions adopted by project

□ Most secure based on past experience or assessment

□ Closely aligns with development lifecycle processes of project

□ Closely aligns with the industry that the project involves

□ Others (pls specify) ____

Q11 Which of the following benefit(s) have you observed after adopting the mentioned framework?

□ Cost reduction (due to early resolution of security-related design flaws or defects)

□ Reduced number of security incidents

□ More secure software being developed or deployed

□ Reduced impact of security incidents

□ Reduction in risk of reputational loss

□ Reduction in overall business risks

□ Meets the legal requirements or regulations required

□ Others (pls specify) ____

Page Break

Q12 How necessary do you think adopting one of these frameworks is?

○ Totally unnecessary

○ Not really necessary

○ Not sure

○ Necessary

○ Absolutely necessary

Q13 How strictly does your project enforce the security processes of this framework?

○ Not enforced at all (about 0% of processes)

○ Lightly enforced (about 25% of processes)

○ Moderately enforced (about 50% of processes)

○ Strictly enforced (about 75% of processes)

○ Almost fully enforced (about 100% of processes)

Q14 What are the possible reasons for the framework NOT being “almost fully enforced”?

□ No formal framework in place

□ Not enough time catered

□ Too expensive

□ Not sure about the process steps

□ No security advisors to provide guidance

□ Others (pls specify) ____

Page Break

Q15 When did your company start to adopt a secure software development life cycle or secure software engineering (SSDLC/SSE) framework?

○ < 1 year ago

○ Between 1-3 years ago

○ > 3 years ago

○ Can't remember

Q16 Do you think there is a greater need to adopt an SSDLC/SSE framework as companies increasingly rely on remote working?

○ Definitely not

○ Probably not

○ Not sure

○ Probably yes

○ Definitely yes

Q17 Has there been any changes or additions to the framework that has been adopted in the last 2 years?

○ No

○ Yes (pls explain the changes briefly)

____

Q18 What are the reasons for the changes?

□ New legal/government regulations

□ Company experienced a cyber-attack

□ Increasing cyber threats reported

□ Increasing remote working requires internet connectivity and increases security risk exposure

□ Others (pls specify) ____

End of Block: Cyber Questions

Start of Block: Demographic

Section 3 Demographic Information

The next few questions relate to your experience and general company information. There is no specific question that will allow you or your company to be identified.

Q19
Besides the current project, what roles have you played in other IT projects?

□ Developer

□ Tester

□ Reviewer

□ Consultant

□ Architect

□ Infrastructure support

□ Team Leader

□ Project Manager

□ Others (pls specify) ____

Q20 How many years of IT experience do you have?

____

Q21 What is the size (i.e. number of employees) of your company?

○ 1-9

○ 11-49

○ 50-249

○ 250 & above

Page Break

Q22 Which category does your company fall under?

○ Multi-National Company (MNC)

○ Local private

○ Government/Public Sector

○ Non-profit

○ Others (pls specify) ____

Q23 Which industry is your company in?

○ Agriculture

○ Manufacturing

○ Education

○ Oil & Petroleum Refining/Retail

○ Construction

○ Accommodation / Hotels

○ Financial and Insurance Activities

○ Public Administration

○ Defence / National Security

○ HealthCare

○ Information and Communication Technology

○ Food & Beverage Service

○ Utilities Supply (Electricity, Gas or Water)

○ Others ____

Q24 Which country is your office in (please state the country where your official office is located, even if working remotely)?

▼ Afghanistan ... Zimbabwe

End of Block: Demographic

Start of Block: Concluding Questions

Section 4 Concluding Questions

Q25 Can I contact you if any further clarification is needed?

○ Yes

○ No

Q26 Do you want a soft-copy of the final research report sent to you?

○ Yes

○ No

Q27 Please provide your email address if you answered “Yes” to any of the last 2 questions.

____

End of Block: Concluding Questions

Appendix 2 – Interview Questions

Section 1: Demographic Information

1. What is your Company Size?
2. What industry is your company in?
3. Type of Company (MNC, local private, government, non-profit etc.)?
4. Which country is your office in?
5. What is your role in the project?
6. How many years of IT experience do you have?
7. What phases of software lifecycle does your project cover (regardless of whether for a new system or an existing one) e.g. Design, Development, Operation?
8. What is the size of the project team?

Section 2: CyberSecurity Questions

9. Could you describe in your own words what are secure software development frameworks?
10. Do you think adopting secure practices or a secure software development framework helps to prevent or reduce the occurrence of cyber-attacks?
11. What secure software development life cycle or secure software engineering (SSDLC/SSE) frameworks have you heard of?
12. What SSDLC/SSE framework is your project currently using? Can you explain the reason for your project adopting the mentioned framework?
    - The majority of our survey respondents use custom frameworks. What do you think could be the reasons behind this?
    - Does your company have a specific policy regarding secure software development? (If yes) Can you tell me more about this policy?
13. What are the major stages/activities that your project follows from these frameworks?
    - What do you think are the key and important stages/activities that must be done to improve the security posture?
14. How strictly does your project enforce the security processes of this framework?
    - How does your company/project enforce the processes?
    - If the framework is NOT almost fully enforced, what are the possible reasons?
15. Has there been any changes or additions to the framework that has been adopted in your project in the last few years? If so, can you describe the major changes and the reasons for the change?

16. In your professional opinion and based on your experience, how would you describe the past trend and future development of adopting secure software development frameworks or practices?