Adherence to Secure Software Development Lifecycle

Alaa' Omar[1][a], Ahmad Alsadeh[2][b] and Mamoun Nawahdah[3][c]

[1]*Master in Software Engineering, Birzeit University, Almarj Str. 1, Birzeit, Palestine (State of)*

[2]*Electrical and Computer Engineering, Birzeit University, Almarj Str. 1, Birzeit, Palestine (State of)*

[3]*Computer Science, Birzeit University, Almarj Str. 1, Birzeit, Palestine (State of)*

Keywords: Secure Software Development, Software Security Engineering, Software Security Principles.

Abstract: Security in software development lifecycle (SDL) is a comprehensive development process for detecting, preventing security defects, and responding to the exploits. In this study, we investigate to what extent the software security principles are adopted in the Palestinian IT sector. Thus, we conducted an online self-administered questionnaire that targeted the Palestinian IT sector on a random sample of participants. The results revealed that most of the security practices are not fully applied by the surveyed enterprises. We found that the security background, company domain, budget, and timeline are influential factors that affect the adoption of security principles during the SDL. In addition, we found that software security is often neglected by most developers, although they are willing to comply with security principles when needed.

1 INTRODUCTION

Software security has become an important topic, especially with the growth of hacking tools and mechanisms (Bendovschi, 2015). Vulnerabilities are becoming more complex, and attacks become more effective. Even large companies have suffered from these attacks and lost money and customers due to these attacks (Tariq, 2018; Cashell et al., 2004).

Actually, Some of these attack incidents happened due to the lack of systematic secure software development process method followed by software development companies during the software development lifecycle (Council et al., 2007). Although most Software vendors confess the significance of undertaking the software security practice, they lack the proper guidance in comprehension of undertaking it (McGraw, 2004). The majority of the companies follow a flowed approach when dealing with security requirements rather than building security-in from the beginning (McGraw et al., 2009) and responding to the security breaches by just doing security patching.

However, there are some advocates for proactive software analysis to prevent vulnerabilities. McGraw (McGraw, 2006) proposed the Touchpoints model where the security activities are distributed among the SDL. Microsoft company developed a structured security process [1] to help developers understand and apply the security principles (Team, 2022). SAFECode [2] is a global industry forum for leading efforts to identify and promote best practices for secure and reliable software. Software Assurance Maturity Model (SAMM) [3] provides guidelines on how real-world security initiatives are organized, what activities they perform, and more. Building Security In Maturity Model (BSIMM) [4] provides a based line for common secure practices observed from high-performing organizations.

This paper aims to study the state-of-the-art practices of the Palestinian IT sector regarding adopting software security principles in SDL. The Palestinian IT sector is a growing sector and relatively small.

We derived the following research questions to support our study goal:

- **RQ1:** What security practices are applied during the SDL by software development teams in the IT Palestinian sector?
- **RQ2:** What are the factors that affect the adoption of the security principles in Palestine?

[a] https://orcid.org/0000-0001-6908-7894
[b] https://orcid.org/0000-0002-5893-4805
[c] https://orcid.org/0000-0003-2278-4101

[1] http://microsoft.com/sdl
[2] https://safecode.org/
[3] https://owasp.org/www-project-samm/
[4] https://www.bsimm.com/

Omar, A., Alsadeh, A. and Nawahdah, M.
Adherence to Secure Software Development Lifecycle.
DOI: 10.5220/0011263700003266
In *Proceedings of the 17th International Conference on Software Technologies (ICSOFT 2022)*, pages 410-417
ISBN: 978-989-758-588-3; ISSN: 2184-2833

To address these questions and validate them empirically, we developed a questionnaire containing both closed and open-ended questions. Software developers were surveyed with different roles, and software security experiences operating in different sectors, including the public and private sectors.

To our knowledge, none of the studies in the literature have targeted the Palestinian IT sector regarding the adherence to the software security practices during the SDL. The main contributions of our research are:

- Evaluate empirically the software security adherence in the Palestinian IT Sector.
- Expose the factors that limit applying the security practices in the Palestinian IT sector.
- Compare our findings with other empirical research that targets similar objectives.

2 RELATED WORK

In this section, we provide a summary of the papers about empirical studies that are highly related to security adoption in the SDL; including studies that aim to investigate different factors that affect the adherence to security best practices.

F. Alghamdi (Alghamdi, 2020) investigated the attributes that affect the adherence to security practices during software development in software development companies. The results of her study showed that the larger the company is the increase in software security adoption is. Besides, custom-made software experiences a lack of awareness of security practices compared to public software companies. In-house and out-source companies have convergence on the adoption of security practices.

E. Venson et al. (Venson et al., 2019) investigated the impact of software security practices on the development estimation effort by conducting a close-ended questionnaire. The results showed that the professional social network contributed to a demographically diverse sampling frame. Their study revealed that security is a factor that motivates effort in software security. In addition, security practices are not taken into consideration in the way they should be while planning the software development initiatives.

H. Assal et al. (Venson et al., 2019) studied the security practices during the SDL and the actions of software development teams to ensure the security of their application, and how the software developer's experience and knowledge of software security practices can affect the adherence to those software security best practices. The results illustrated that the real-world security practices differed significantly from those found in literature, they are almost neglected as they increase the load on the team. The same authors also have studied the intersection between developers and software companies in the security process using an online survey tool (Assal and Chiasson, 2019) that companies both close-ended Likert Scale questions and open-ended short answers questions. The results of their study revealed that the participants deemed software security practices in software development in diversified ways. In addition, their results also showed that the developers are not blamed for neglecting the major role of applying software security principles, on the contrary, they are self-driven and willing to confirm security principles when needed.

P. Morrison et al. (Morrison et al., 2017) introduced a set of security best practices and empirically validated the driver list by conducting a survey on twelve open source projects that are expected to be users of security practices. Besides, the authors presented a set of security adherence metrics and build a model to assess how adherence metrics influence practice usage and how it is compared to the result of the conducted survey. Their study results showed a statistically significant correlation between security training and its adoption.

T. Thomas et al. (Thomas et al., 2018) showed that the application security work is done by software security experts isolated from the rest of the team, which leads to extra overhead in the communication for fixing vulnerabilities.

3 RESEARCH METHODOLOGY

A survey was conducted in Palestine and covers participants from multiple software companies and organizations that form the Palestinian IT sector. The majority of the participants hold a bachelor's degree (75%); the rest (25%) hold a master's degree. The participant's specialization fell into four categories (computer Engineering, Software Engineering, Computer Science, and Information Technology with (52%, 36%, 4%, 2%) respectively.

Our survey targeted three sectors: educational institutes, public sectors, and private sectors. Where the team size differs per sector, for example, 100% of the respondents who work in educational organizations have stated that the team size is between 1 to 5 employees, whereas in the public sector the team sizes fall within 1 to 5 with a percent of 66.6%. Finally, 59.37% have 1 to 5 years, 18.75% have 6 to ten, 9.37% have 11 to 15 years, and the rest 12.5% has more than 15 years of experience.

We conducted a survey of type self-administered questionnaire that mixes close-ended and open-ended questions to cover a wide range of security practices and gain a deep understanding of the participant's selected answers to the close-ended questions. The survey was developed to measure participants' awareness of how important is engaging software security best practices in software development from the first beginning (security in) and to reveal the factors that foster SDL adoption in the Palestinian IT sector from the vendor's point of view.

The survey was administered by one of the authors, who presented the survey to the participants enrolled in companies or organizations that have development teams, this includes project managers, software engineers/developers, testers, and others who work in software development. The companies varied in size and business domain, we served only one member of the same team from both genders and from different experiences and roles.

In order to present our findings, non-parametric tests were used, since the data is ordinal, and does not follow the bell shape. A Spearman's rank-order correlation was used to measure the association between variables. Next, for those who have a significant relationship, Mann-Whitney U-Test was used to determine which treatment has a significant relationship [5].

Regarding the open-ended questions, we applied the thematic coding process (Saldana, 2012). A set of codes were formulated. Next, the codes were grouped into themes, in a second cycle, and stored in Excel spreadsheets. Then the spreadsheets were analyzed.

4 RESULTS

We received a total of 40 responses. Survey respondents varied in gender, qualification, and years of experience. Table 1 shows the demographic information of the participants.

Table 1: Demographic Information.

Demographic variables		**Count**	**Percent**
Gender	Male	24	60%
	Female	16	40%
Qualification	Bachelor	30	75%
	Master	10	25%
Specialization	Computer Engineering	26	52%
	Software Engineering	18	36%
	Computer Science	2	4%
	Information Technology	1	2%
Experience	1–3 years	12	30%
	4–6 years	12	30%
	7–10 year	9	22.5%
	11 years or more	7	17.5%
Work sector	Private Sector	32	80%
	Public Sector	6	15%
	Educational Organization	2	5%
Team size	1–5	16	40%
	6–10	13	30%
	11–15	5	12%
	More than 15	5	12%

4.1 Answering RQ1

To address RQ1 which contains a mini list of the security best practices adopted from (Morrison et al., 2017), we represent the results for each security practice separately.

Table 2 enumerates a compiled list of software security practices taken from different security engineering processes, such as BSIMM, Microsoft SDL, and SAFECode. The practices were picked through content analysis, compiled into 16 best practices, and validated in a pilot data collection survey of 11 open source projects focused on security. The following are the results per practice, as shown in Figure 1:

P1) *Apply Security Requirements*: The majority of the respondents(28%) said that they do not apply security requirements during the software development, while (20%) do it once in the project, then comes weekly and daily with equal percentages (18%), and (8%) said they apply it monthly. Finally, only (5%) apply it quarterly.

P2) *Apply Data Classification Scheme*: A data classification Scheme helps organization classify their data, improve their security and prioritize them. Besides, it is important to ensure that the data is in regulatory compliance. Most of the respondents 40% neglected to apply data classification schemes in their projects even once. 18% of the respondents apply it either quarterly or monthly, 15% apply it weekly, and the rest 5% apply it daily.

P3) *Apply Threat Modeling*: This security practice is not applicable 38%. The rest of the percentages are distributed as follows: 18% do it weekly, 15% apply it once in the project and 13% apply it quarterly, whereas only 3% apply it daily.

P4) *Document Technical Stack*: The dominant choice among the others with 30%, followed by once in a project 25%, monthly 23%, weakly 13%, 5% quarterly, whereas, annually and daily are 3%.

P5) *Apply Secure Coding Standards*: The majority

[5] The dataset and analysis results can be found using the link https://github.com/eng-aomar/Security_in_practice

Table 2: Software Security Practices (Morrison et al., 2017).

ID	Security Practice	Description
P1	Apply Security Requirements	Think about and document security apprehensions before implementing software features.
P2	Apply Data Classification Scheme	Preserve and implement a data classification system. Distinguish and document security-sensitive data, personal information, financial information, and system credentials.
P3	Apply Threat Modeling	Expect, resolve, and notarize how and why adversaries may try to misapply the software.
P4	Document Technical Stack	Notarize the components used to build, test, deploy, and operate the software. Keep components up to date on security patches.
P5	Apply Secure Coding Standards	Implementing security-focused coding standards for each language and component used in building the program.
P6	Perform security testing	Consider security requirements, threat models and all other security-related information, and tools when designing and implementing a program test plan.
P7	Perform Penetration Testing	Organize a security-focused stress test for project software in your production environment Program project team.
P8	Perform Security Review	Conduct a security-focused review of all output, including, but not limited to, design, source code, software version, and documentation. Include reviewers who did not produce the final product under review.
P9	Publish Operations Guide	Document security concerns apply to administrators and users and support how the program is configured and running.
P10	Track Vulnerabilities	Tracking software vulnerabilities discovered in the program and prioritizing their solutions.
P11	Improve Development Process	Incorporate ”lessons learned” from security vulnerabilities and their resolutions into the project’s software development process.
P12	Perform Security Training	Make sure that project personnel is trained in security concepts and role-specific security techniques.

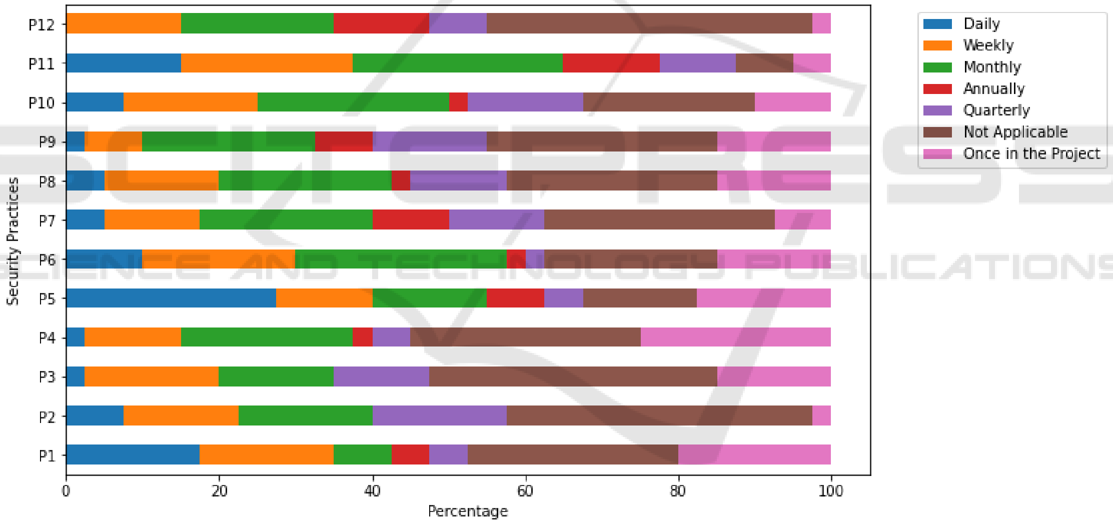

Figure 1: The answers to RQ1: security practices that are applied by software development teams in Palestine.

of the respondents 28% stated that they apply security coding standards daily, the reaming of the choices gained a relatively close percentage (around 15%), whereas the least percentage goes for the quarterly choice with only 5%.

P6) *Perform Security Testing*: The percentages for Perform Security testing monthly, not applicable, and weekly obtained as 28%, 23%, and 20%, whereas the options (”Once in the project” and ”Daily”) obtained 15%, and 10%, and the rest of the options obtained 3%.

P7) *Perform Penetration Testing*: Penetration testing is defined as an authorized cyber-attack done by specialized pen testers, this aims to find the possible risks and threats that abuse the system, and it measures the already implemented risk mitigation techniques applied in the current project under test, this is important to reveal hidden gaps. When the participants were asked about applying penetration testing in their current projects, the frequencies shows that ”Not Applicable” came first with a 30% of the votes, while ”Monthly” came in second place with a 23% percent, followed by 13% for the ”Quarterly”, whereas the other choices gain percentages 10% and less.

P8) *Perform Security Review*: The majority of the respondents 28% stated that they did not perform security reviews in their project even once, this percentage is followed by 23% have stated that they do it Monthly, and 15% do it either once in the project or Weekly. While 13% perform security reviews quarterly and only 5% do it daily.

P9) *Publish Operations Guide*: We can see that the publish operations guide security principle is not widely adopted in the Palestinian IT sector, as 30% of the respondents said it is not applicable, while 23% apply it Monthly, 15% apply it either once in the project or quarterly, the rest 8% apply it either annually or weekly.

P10) *Track Vulnerabilities*: The majority of the participants (25%) mentioned they apply this principle monthly. This is close to those 23% who stated that this principle is not applicable. The rest of the options; Weekly, Quarterly, Once in the project, Daily, and Annually) obtained 18%, 15%, 10%, 8%, and 3% respectively.

P11) *Improve Development Process*: The improved development process principle implies engaging the lessons learned in refining and adjusting the SDL. In this context, the respondents implied, that this principle is applied Monthly at 28%, while Weekly and Daily come after with the percentages (23% and 15%), whereas the rest obtained 13% and less.

P12) *Perform Security Training*: The majority (43%) of the IT sectors in Palestine do not perform security training for their employees. While 20% stated that they do it monthly, 15% weekly, 8% quarterly, and only 3% do security training once in the project.

4.2 Answering RQ2

To address RQ2, we applied Spearman's rank-order correlation test to find the factors. Table 3 shows a compact correlation matrix that contains only the independent variable and their corespondent correlated security practice. There is a significant relationship between participants sex with security principles (P10 =Track Vulnerabilities, P11=Improve Development Process, and P12=Perform Security Training), with r(40)=.345, p=.015, r(40)= .370, p=.009, and r(40)=.264, p=.050 respectively. The direction of the correlation was positive.

In Table 4, Mann Whitney U-test was performed to compare the mean ranks between the two genders of participants (male and female). On P10, the 16 females have significantly higher mean ranks (25.28) than the 24 males (17.31), U = 115.00, p = 0.031. In addition, the 16 females have a significantly higher mean rank (25.63) than the 24 males, mean rank (17.08) on P11. The last principle shows that there is no significant relationship between sex and adopting this principle because p(0.099) is greater than (0.05). While the academic degree factor has no significant relationship with the security principle [P3] Apply Threat Modeling, as p(0.099) is greater than (0.05). Regarding the Specialization variable, the Kruskal-Wallis Test results show that there is no significant relationship between the specialization and [P4] Document Technical Stack security principle, with p(0.337) being greater than (0.05).

To identify which security self-rating level has a significant relationship with the mentioned security principles in Table 3, Kruskal-Wallis Test was performed, and the results show that there is a significant relationship with ([P3] Apply Threat Modeling,[P5] Apply Secure Coding Standards, [P7] Perform penetration testing, [P8] Perform Security review, [P10] Track vulnerabilities) with P= (**0.045, 0.005, 0.004, 0.020**) respectively. Doing the post-hock test (Mann Whitney U test) to determine which level has the significant relationship. The results of Mann Whitney test show that the 13 participants with security self-rated level equals (3) have significantly higher mean ranks on P[3],[P7],[P8],and [P10], with mean ranks **(108.5, 237.00, 237.5, 239.00)** and U= (17.5, 49, 48.5, 47.0), p = (0.049, 0.021, 0.019, 0.018) respectively.

Table 3: Correlation.

	Sex	Qualification	Specialization	Software security self-rated
P1				-.377**
P2				-.277*
P3		-.264*		-.359*
P4			-.269*	-.366*
P5				-.493**
P6				-.518**
P7				-.349*
P8				-.465**
P9				
P10	.345*			
P11	.376**			
P12	.264*			-.326*

*. Correlation is significant at the 0.05 level (1-tailed).
**. Correlation is significant at the 0.01 level (1-tailed).
P1-P12 (software security practises defined in Table 2.

4.3 Qualitative Analysis

In this section, the results of the qualitative analysis for the open-ended questions are presented. The re-

Table 4: Mann Whitney U Participants sex.

Sex		N	Mean Rank	$\sum Ranks$	U	Sig.
P10	Male	24	17.31	415.50	115.5	0.031
	Female	16	25.28	494.50		
	Total	40				
P11	Male	24	17.08	410.00	110.00	0.021
	Female	16	25.63	410.00		
	Total	40				
P12	Male	24	18.13	435.00	135.00	0.099
	Female	16	47.04	2258.00		
	Total	40				
Academic Degree		N	Mean Rank	$\sum Ranks$	U	Sig.
P3	Bachelor	30	22.2	666.0	99.00	0.099
	Master	10	15.40	154.0		
	Total	40				

*. Correlation is significant at the 0.05 level (1-tailed).
P3. Apply Threat Modeling. P10. Track Vulnerabilities. P11. Improve Development Process. P12. Perform Security Training.

sults are grouped for each question separately. When the participants were asked "**What are your main priorities when doing development?**", the answers were divided around the main ideas shown in Table 5. The majority of the respondents (30%) imply that sticking to the requirement and starting from the high priority ones is at the top of their priorities when doing software development. While security lies on the second priority with a percentage of 22%, 20% for applying testing and quality assurance, 12% for performance, 10% for Writing clean code, and 6% for usability.

Table 5: Declared Current Priorities During Software Development.

Main Priority	**Declared Priority**	**Percent**
Requirement	15	30%
Security	11	22%
Quality	10	20%
Performance	6	12%
Clean Code	5	10%
Usability	3	6%
Total	**50**	

Q1. What are your main priorities when doing development?

When the respondents were asked, **"Do your priorities change when a deadline approaches?"**, Table 6 shows that 75% answered "No", while 15% declared that their priorities change. This change depends on the release and the requirement prioritization.

Table 7 shows the answers to the third question **"How does security fit in your priorities?"**, The majority 25% said that it is a top priority, while 15% said it is the least priority for different reasons, and only 5% said it lies in the middle of their priority. 7.5% think it is only important in the testing phase. 2.5% think security requirements are an integral part of the project, and 2.5% think security is about Authentication, Authorization, and Data integrity. 2.5% consider security in the system analyst phase of the SDL. 2.5% Apply security only during the implementation phase. While 2.5% think that security is included in the technologies used.

Table 6: Priorities Change During Software Development.

Keyword	**Declared Answers**	**Percent**
No	30	75%
Yes	6	20%
Depends on the release	2	7%
Requirements prioritization	2	7%
Total	**40**	

Q2. Do your priorities change when a deadline approaches?

Table 7: Where Software Security Fits in?

Keyword	**Count**	**Percent**
Priority	10	25%
Least priority	6	15%
Important in the testing	3	7.5%
In the Middle	2	5%
Apply minimum software security	1	2.5%
Authentication, Authorization and Data integrity	1	2.5%
Security requirement are Integral part of all projects	1	2.5%
Adopting security practices while preparing for system analysis	1	2.5%
Included within the technologies used	1	2.5%
Applied in the implementation	1	2.5%
Total	**40**	

Q3. How does security fit in your priorities?

In Table 8, we asked respondents **"Do you apply security tooling?"** if yes **"What tools do you use? and in which SDLC phase?"**. 21 answered "No", while 2 answered "Yes", but did not specifies in Which SDLC phase.4 stated that they use it in Implementation, by doing encryption and web services. While 2 uses `SonarQube`, 1 uses `Tenale`, 1 uses `Quyals`, 1 `GDB`, x64 in the testing phase, 1 uses Kali, and 1 uses `GitLab`.

5 DISCUSSION

We found that the adherence to the security principles during the SDL in the Palestinian IT sector is still immature. When the respondents were asked whether the security principles are applied, the choice "Not Applicable" was clearly dominant with a percent

Table 8: Apply Security Tools.

Keyword	Count	Software Development Phase
No	21	NA
Yes	2	NA
HTTPS Certificate	1	Deployment
Data Encryption	2	Implementation
SonarQube	2	CI
Tenable	1	NA
Quyals	1	NA
Web services	2	Implementation
GDB, x64	1	Testing
Kali	1	NA
GitLab	1	After Sprint
Total	**40**	

Q4. Do you apply Security Tooling? If yes, What tools do you use? and in which SDLC phase?

of 61.5% in most security principles. The principle (Perform Security Training) came first with a percent 43%, this high percentage reflects the current culture found in the Palestinian IT sector that does not give the needed training and attention. In the second place comes the security principle (Applying Threat Modeling) 38%, this leads to conclude that security in the Palestinian IT sector is not a built-in process. When it comes to security, an unplanned operation is done in an ad-hoc manner with the absence of a systematic approach to be engaged as an integral part of the SDL.

Table 9 and Table 10 show a summary of our results compared to previous studies. Our results show that the top two practices that are “daily” applied are: “Apply Secure Coding Standards and, Apply Security Requirements” with percentages of 28% and 18%.

Table 9: RQ1 Dominant Choice Comparison.

Study Ref.	Dominant choice	Percentage
Our Study	Not Applicable	61.5%
(Venson et al., 2019)	Daily	31%
(Morrison et al., 2017)	Weekly	31%

Compared to similar studies, the **RQ1** results of Morrison et al. (Morrison et al., 2017) show that the top two **daily** reported practices are **Apply Secure Coding Standards and, Track Vulnerability** with **45% and 42%** respectively, and **42%** said that **Publish Operation Guide** is **Not Applicable**. While, the results of Venson et al. (Venson et al., 2019) show that the top two **daily** reported practices are **Apply Secure Coding Standards and, Apply Security Tooling** with **54% and 36%** respectively, and **22%** said that **Publish Operation Guide** is **Not Applicable**.

From Table 10, we can see that in all of the mentioned studies, the security principle **Apply Secure Coding Standards** is applied Daily with the highest percentage, but in the Palestinian market, the percentage is relatively smaller compared to Morrison (Morrison et al., 2017), and Venson findings (Venson et al., 2019). This percentage reflects the current culture in the Palestinian companies and the percentage of awareness to apply security coding standards during the implementation phase, which is relatively modest and needs more attention. This low commitment from the Palestinian market is reasonable and it comes in a context consistent with the rest of the results, especially if we know that the **Perform Security Training** is not Applied with a percentage of 43%.

Table 10: The Top Two Security Practices Applied **Daily** Comparison.

Study Ref.	Applying security principle	Percentage
Our Study	Secure coding standards	28%
	Security requirements	18%
(Venson et al., 2019)	Secure coding standards	54%
	Security tools	36%
(Morrison et al., 2017)	Secure coding standards	45%
	Track vulnerability	42%

By closely examining our study results, especially the open-ended questions, and by immersing ourselves in the experiences reported by the participants, we realized the factors that seem to shape their practices that may not be sufficiently taken into account by best practices. We present each worker and conflict with best practices, if applicable.

1. *Security Background*: There is a lack of knowledge of the principles of security in SDL. Some of the participants misunderstand the security roles within the SDLs. They think security is just about authentication, authorization, and data integrity. Others think that it is only considered in web development. And a group of the participants thinks it is embedded with the technology stack that is used by the team. The results clearly proved that the majority of participants do not receive appropriate training during their employment even once. This lack of knowledge is clearly reflected in the extent of implementation of security during the SDL in a negative way.
2. *Company Domain*: Our results revealed that the domain of the companies plays an important role in reshaping the culture inside these companies regarding the adoption of security principles and the awareness of it is important. Participants working in enterprises that adopt a security culture, tend to use tools and apply security principles more than those who stated that security is the least priority in their work.
3. *Budget and Timeline*: The lack of time offered for developers to accomplish their tasks is limiting them from taking care of security issues. Most

of the developers focus on meeting the requirements and customer satisfaction delivering their preferred features on time with the least cost. Besides, the limitation of the project budget offered to the software projects often stands against the adoption of security in SDL as the main goal.

6 THREATS TO VALIDITY

The size of the sample may be relatively small (40 participants). However, this sample is acceptable since the Palestinian IT sector is relatively small. On the other hand, the results of this research are limited to the Palestinian IT sector, and hence, it could not be generalized but can be replicated in other countries.

1. *Internal Validity*: Few participants know the authors in person. Thus, their responses to this fact may be affected by the participant trying to satisfy the author by choosing the answer that corresponds with the survey context. However, the number of them is limited, as the survey was delivered to companies without revealing personal information about the author.
2. *Construct Validity*: The participants could guess the research questions from the context of the survey title. Therefore, some participants' might answers to the survey may be affected. However, we think that the number of participants affected by their responses, if any, was few since we are dealing with mature and independent participants with their opinions.
3. *External Validity*: The sample as described in section 3 was limited to the Palestinian IT sector. However, the study took into consideration the diversity of the sample selection as software developers from various sectors, experiences, and company sizes, which ensures that the sample is as representative as possible.

7 CONCLUSION

The adherence to secure SDL of the Palestinian IT market is still modest, immature, and unsystematic, as non of the surveyed companies followed one of the known security models, such as Microsoft SDL, rather than, security in SDL is taken into consideration based on the developer skills and knowledge. Moreover, most of the security practices are not applied even during the project timeline. Successful adoption of a secure software development process requires company cultural change, in addition to developers' training and adopting the technical practices. Thus, it is important for software development companies to adopt their own customized secure SDL.

REFERENCES

Alghamdi, F. (2020). Motivational company's characteristics to secure software. In *2020 3rd International Conference on Computer Applications & Information Security (ICCAIS)*, pages 1–5. IEEE.

Assal, H. and Chiasson, S. (2019). 'think secure from the beginning' a survey with software developers. In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–13.

Bendovschi, A. (2015). Cyber-attacks – trends, patterns and security countermeasures. *Procedia Economics and Finance*, 28:24–31. 7th INTERNATIONAL CONFERENCE ON FINANCIAL CRIMINOLOGY 2015, 7th ICFC 2015, 13-14 April 2015,Wadham College, Oxford University, United Kingdom.

Cashell, B., Jackson, W. D., Jickling, M., and Webel, B. (2004). The economic impact of cyber-attacks. *Congressional research service documents, CRS RL32331 (Washington DC)*, 2.

Council, N. R. et al. (2007). *Software for dependable systems: Sufficient evidence?* National Academies Press.

McGraw, G. (2004). Software security. *IEEE Security & Privacy*, 2(2):80–83.

McGraw, G. (2006). *Software security:Building Security in*. Addison-Wesley Professional.

McGraw, G., Chess, B., and Migues, S. (2009). Building security in maturity model. *Fortify & Cigital*.

Morrison, P., Smith, B. H., and Williams, L. (2017). Surveying security practice adherence in software development. In *Proceedings of the Hot Topics in Science of Security: Symposium and Bootcamp*, pages 85–94.

Saldana, J. (2012). *The Coding Manual for Qualitative Researchers*. SAGE Publications.

Tariq, N. (2018). Impact of cyberattacks on financial institutions. *Journal of Internet Banking and Commerce*, 23(2):1–11.

Team, M. (2022). About microsoft sdl. https://www.microsoft.com/en-us/securityengineering/sdl/about.

Thomas, T. W., Tabassum, M., Chu, B., and Lipford, H. (2018). Security during application development: An application security expert perspective. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–12.

Venson, E., Alfayez, R., Gomes, M. M., Figueiredo, R. M., and Boehm, B. (2019). The impact of software security practices on development effort: An initial survey. In *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 1–12. IEEE.