SECURITY CHECKLIST EVALUATION IN SOFTWARE DEVELOPMENT USING GOAL QUESTION METRIC APPROACH

Mazni binti Mohamed Jakeri, Universiti Teknologi Petronas
Mohd Fadzil Hassan, Universiti Teknologi Petronas
Aliza Sarlan, Universiti Teknologi Petronas
Amirudin Abdul Wahab, Menara Cyber Axis Jalan Impact

ABSTRACT

The checklist has been published and used by organisations as a guideline to track and monitor the practices applied in the software development process. This paper discussed the Goal Question Metric (GQM) approach in developing, measuring, and evaluating the security tasks for the security checklist implementation for one of the selected security activities in the design phase in the secure software development life cycle (SSDLC). The security checklist was developed by adapting security tasks from various resources to achieve the security activity's goal. The security checklist has been applied in the multiple-case study in the in-house web-based development teams in the Malaysian public sector to measure and evaluate the implementation of the security tasks. The findings indicated that additional steps need to be taken in order to maximise the number of security tasks performed and achieve its goals.

Keywords: Security Checklist, Security Tasks, Goal Question Metric, Software Development, Malaysian Public Sector Agencies

INTRODUCTION

Software development life cycle (SDLC) is a framework that describes activities performed throughout the development process and focuses completely on functionality and features. The secure software development life cycle (SSDLC) is set up by incorporating security-related activities to each phase of the existing development process (Mougoue, 2018; Mohaddes, 2015) to increase the security posture of the SDLC on which the activity is performed (Khan, 2009). For example, integrating risk analysis in the requirement phase, additional design in the design phase, code review in the development phase, and penetration test in the testing phase.

There are various approaches currently used for security integration. Established organisations have published secure frameworks with security activities integration as references for organisations and developers with the goal to reduce the number and severity of vulnerabilities in software (Microsoft Corporation, 2010). Examples are Cybersecurity Guidelines for SSDLC published by CyberSecurity Malaysia (CSM) (CyberSecurity Malaysia, 2019), Microsoft Security Development Lifecycle (MS SDL) by Microsoft Corporation (Microsoft Corporation, 2010), and Cigital Touchpoints (CT) by Cigital Inc. (McGraw, 2006) as a reference for organisations and developers. Each organisation presents a list of security activities that should be incorporated and implemented to produce secure applications. Therefore, there is a need to measure and evaluate the implementation of the security activities to achieve its goal.

This paper proposes the use of a security checklist to measure and evaluate the implementation of the selected security activity, which is an additional design. The main contribution of this paper is to develop a security checklist, to measure and evaluate the implementation of security activity by utilising the Goal Question Metric (GQM) approach. The

Citation Information: Jakeri, M.B.M., Hassan, M.F., Sarlan, A., & Wahab, A.A. (2022). Security checklist evaluation in software development using goal question metric approach. *Journal of Management Information and Decision Sciences, 25*(S2), 1-12.

security checklist was validated by three case studies in the in-house development teams in the Malaysian public sector agencies.

MATERIALS AND METHODS

This section of the paper presents the details of the proposed security checklist. To start with, the GQM approach is described. Then, the structures of the proposed security checklist are presented.

Goal Question Metric (GQM) Approach

The Goal Question Metric (GQM) approach, developed by Basili, et al., (1994) is a way to find out why and what to measure (Lindström, 2004). It consists of goals, questions, and metrics hierarchically as presented in Figure 1. A goal is referred to as a mission (Yahya, 2017). Then, the goal is refined into several questions and each question is then refined into several metrics (Lindström, 2004). One metric may be used to answer different questions under the same goal (Basili, 1994).

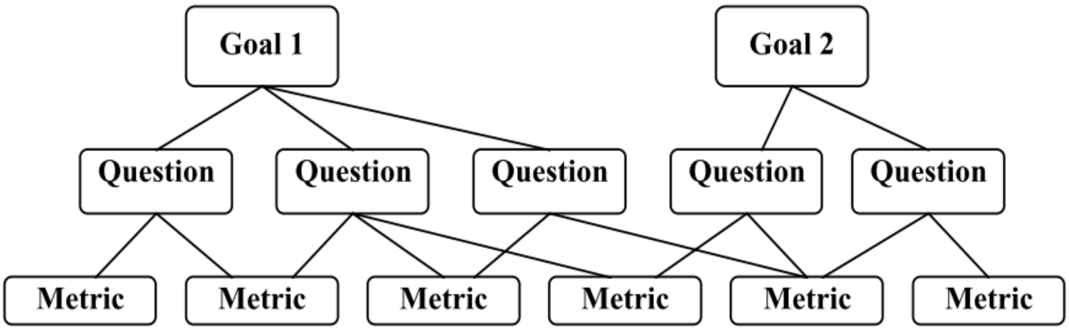

FIGURE 1
GQM MODEL

GQM has been used for a variety of purposes. For example, Ampatzoglou, et al., (2021) have applied GQM only to develop research questions based on the goal of their study whereas Gosain & Singh, (2020) used it to define the goal. GQM was used to set the goals and questions in analysing the business processes in order to identify the most suitable Robotic Process Automation (RPA) (Leshob, 2018); Ergasheva, et al., (2019) have applied the GQM approach entirely to develop and evaluate the software systems. The goals, questions, and metrics were defined through the interview of 67 people from the software development industries. Roseberry, et al., (2019) used GQM to evaluate the effectiveness of formal peer reviews (FPRs) at identifying defects in SDLC. The questions were formulated from the goal and the metrics were collected based on the questions.

Yahya, et al., (2017) have applied the GQM model to identify Security Metrics (SM) to evaluate the security control features of cloud storage systems for IT security practitioners. The security goals were based on the CSA Control Matrix (CCM) and other controls from the literature. Then the goals were refined into questions and the metrics derived from the questions. Halabi & Bellaiche, (2017) also used GQM to measure the performance of cloud security services and three types of security evaluation metrics were identified. In order to set up the Business Continuity Management (BCM) as well as towards the BCM compliance, Mansol, et al., (2016) adapted GQM to determine the goals and questions during the requirement stage. The organisational culture values collected through the survey distributed to 300 participants were used to define the metrics.

The Proposed Security Checklist

The GQM approach was adapted to develop the security checklist evaluation metrics for the security activities in the web-based application development. In this study, the additional design in the design phase as listed in the Secure SDLC by CSM (2019) was adapted to develop the security checklist. The security checklist for additional design consisted of four main goals which were:

- Goal 1: Using the latest version of selected programming languages to implement the design.
- Goal 2: Documenting all data type, format, range, and length.
- Goal 3: Improve the database security.
  - A. Database authentication
  - B. Database authorisation
  - C. Additional database security
- Goal 4: Interconnectivity.

Goal 1 is about the programming languages used to develop web-based applications. The goal is to use the latest version of programming languages in order to obtain security support from the provider while the older versions are exposed to unpatched security vulnerabilities. Goal 2 is on documenting all the data types, format, range, and length used in specifying the inputs. It is needed as a reference in the applications development. Goal 3 is divided into three goals which are database authentication, database authorisation, and additional database security. The main target is to improve database security from unauthorised access. Goal 4 aims to design upstream and downstream compatibility for software. This is especially necessary when it involves delegation of trust, single sign-on (SSO), token-based authentication, and cryptographic key sharing between applications (CyberSecurity, 2019).

The questions and the metrics for each question were adapted from (CyberSecurity, 2019; OWASP, 2010; ORACLE, 2020; MAMPU, 2007; Saive, 2020). The metrics represent the security tasks that need to be implemented to answer the questions and achieve the goals. Initially, the security checklist has 12 questions and 50 metrics. It consisted of subjective and objective metrics (Dichotomous). As for subjective metrics, the respondents were required to fill in their feedback while for objective metrics, they were required to select either ‘Y’ represents that the action has been taken or ‘N’ represents that no action has been taken.

The security checklist was reviewed and verified by a security expert from the Malaysian Administrative Modernisation and Management Planning Unit (MAMPU) through a structured interview session. He has wide experience in handling security incidents in the Malaysian public sector, data leakage protection management, and as an advisor for public sector cybersecurity. One question (Q2) and four additional metrics (M2, M33, M34, and M53) were added to strengthen the security checklist and to achieve its goals. The enhancement security checklist is shown in Table 1.

TABLE 1
SECURITY CHECKLIST FOR THE ADDITIONAL DESIGN IN THE DESIGN PHASE

Goal	Question and Metric			Reference
G1	Using the latest version of selected programming languages to implement the design.			
	Q1	What is the programming language used to implement the design?		
		M1	Name of the programming language used.	(CyberSecurity, 2019)
	Q2	Is it the latest version of the programming languages used?		
		M2	The version of the programming language used.	Additional metric
G2	Documenting all data type, format, range, and length.			
	Q3	Do all inputs are specified and documented with the data type, format, range, and length (e.g.: input name: loginID, type: text		

		maxlength: 20)?		
		M3	All inputs are specified and documented with the data type, format, range, and length (Y/N).	(CyberSecurity, 2019)
G3	Improve the database security.			
G3 (A)	A: Database authentication.			
	Q4	What are the security tasks involved in account locking?		
		M4	Numbers of repeated failed login attempts allowed (maximum 3 times)	(MAMPU, 2007)
		M5	The user's account is lock automatically or manually by the database administrator after a specified number of repeated failed login attempts (Y/N).	(ORACLE, 2020)
		M6	The user's account is unlocked automatically or manually by the database administrator (Y/N).	(ORACLE, 2020)
		M7	User's account logoff automatically if there is no activity within a specified period (Y/N).	(MAMPU, 2007)
		M8	Automatic active user revalidation after a specified period (Y/N).	(MAMPU, 2007)
		M9	Enforce user logoff and termination of all privileges for the user that has been transferred or retired or quit (Y/N).	
		M10	The same user ID cannot be used for more than one session (Y/N).	(MAMPU, 2007)
		M11	User privileges are suspended after 30 days of inactivity and deleted after 30 days of suspension of use (Y/N).	(MAMPU, 2007)
	Q5	Are the password lifetime and expiration implemented?		
		M12	Password lifetime and expiration implemented (Y/N).	(MAMPU, 2007)
		M13	The lifetime period for passwords is restricted (Y/N).	(MAMPU, 2007)
		M14	The user must change the password during the grace period (Y/N).	(ORACLE, 2020)
		M15	The user's account is locked after the password expires (Y/N).	(ORACLE, 2020)
	Q6	Is the password history saved?		
		M16	Password history for each user is saved (Y/N).	(ORACLE, 2020)
		M17	The password is NOT re-used (Y/N).	(ORACLE, 2020)
	Q7	Is the password complexity verification applied?		
		M18	The minimum password is twelve characters in length (Y/N).	(MAMPU, 2007)
		M19	The password is NOT equal to the user ID (Y/N).	(MAMPU, 2007)
		M20	The password includes at least one alphabet character, one numeric character, and one punctuation mark (Y/N).	(MAMPU, 2007)
		M21	The password did NOT match any word on an internal list of simple words like welcome, account, database, user, and so on (Y/N).	(ORACLE, 2020)
		M22	The password differs from the previous password by at least three characters (Y/N).	(ORACLE, 2020)
	Q8	Is additional password management applied?		
		M23	Password entry should be hidden on the user's screen. (e.g., on web forms use the input type "password") (Y/N).	(MAMPU, 2007)
		M24	Password reset and changing operations require the same level of controls as account creation and authentication (Y/N).	(OWASP, 2010)
		M25	If using email-based resets, only send email to a pre-registered address with a temporary link/password	(OWASP, 2010)

			(Y/N).	
		M26	Temporary passwords and links should have a short expiration time (Y/N).	(OWASP, 2010)
		M27	Enforce the changing of temporary passwords on the next use (Y/N).	(OWASP, 2010)
		M28	Notify users when a password reset occurs (Y/N).	(OWASP, 2010)
		M29	Enforce password changes based on requirements established in policy or regulation. Critical systems may require more frequent changes. The time between resets must be administratively controlled (Y/N).	(MAMPU, 2007)
		M30	Enforce password changes for the first-time login or after the password has been reset (Y/N).	(MAMPU, 2007)
		M31	Disable "remember me" functionality for password fields (Y/N).	(OWASP, 2010)
		M32	The last use (successful or unsuccessful) of a user account should be reported to the user at their next successful login (Y/N).	(MAMPU, 2007)
G3 (B)	B: Database Authorisation			
	Q9	Is secure database management via web login interface implemented?		(Saive, 2016)
		M33	Is the database management via web login interface performed (e.g., PhpMyAdmin) (Y/N)?	
		M34	Is there any policy permitting database management via web login interface (Y/N)?	Additional metric
		M35	The access is restricted to a specific IP range (Y/N).	Additional metric
		M36	The default login URL is changed (Y/N).	
		M37	HTTPS is enabled (Y/N).	
	Q10	Is secure database management implemented?		(Saive, 2016)
		M38	Password protection to the database (Y/N).	
		M39	Disable root login to the database (Y/N).	
	Q11	Do database privileges implemented?		(Saive, 2016)
		M40	Database-specific privileges to database administrator is setup (Y/N).	
		M41	Database-specific privileges to the users are setup (Y/N).	
G3 (C)	C: Additional database security			
	Q12	Is additional database security performed?		(OWASP, 2010)
		M42	Connection strings should NOT be hardcoded within the application (Y/N).	
		M43	Connection strings should be stored in a separate configuration file on a trusted system (Y/N).	
		M44	The connection string should be encrypted (Y/N).	
		M45	Close the connection as soon as possible (Y/N).	
		M46	Remove or change all default database administrative passwords (Y/N).	
		M47	Utilize strong passwords/phrases or implement multi-factor authentication (Y/N).	
		M48	Disable any default accounts that are not required to support business requirements (Y/N).	
G4	Interconnectivity			
	Q13	Is interconnectivity applied?		(CyberSecurity, 2019)
		M49	SSO is applied (Y/N).	
		M50	Token-based authentication is applied (Y/N).	
		M51	Cryptographic key sharing between applications is applied (Y/N).	
		M52	Upstream and downstream compatibility of software should be explicitly designed (Y/N).	

		M53	Authentication and authorization t to NAS connections are applied (Y/N).	
		M54	NAS access is restricted to a specific IP range (Y/N).	Additional metric

Data Collection

The target samples were three in-house web-based applications development teams in the Malaysian public sector agencies, namely Teams 1, 2, and 3. Each team was represented by IT Officer and Assistant IT Officer, except for Team 2 that only consisted of two IT Officers. The IT Officer for Team 1 was the Project Manager (PM) as well as the system analyst, while the Assistant IT Officer was the programmer. As for Team 2, one IT Officer was the PM, while the other IT Officer was the system analyst and the programmer. For Team 3, the team has two IT Officers, who were the PM and the system analyst, as well as an Assistant IT Officer as the programmer. They had 5 to 15 years of experience in in-house web-based applications development. Each team was given the security checklist from 14th September 2020 and was collected on 9th March 2021. They were given almost six months to implement the listed security tasks. The implementation status was monitored through phone calls, email, and WhatsApp.

RESULTS AND DISCUSSION

This section presents the results and discusses the findings gained from the multiple-case study.

Results

TABLE 2
RESPONSES TO THE METRICS

Goal	Question and Metric			Team 1	Team 2	Team 3
G1	Using the latest version of selected programming languages to implement the design.					
	Q1	What is the programming language used to implement the design?				
		M1	Name of the programming language used.	PHP	PHP	PHP
	Q2	Is it the latest version of the programming languages used?				
		M2	The version of the programming language used.	7.3	5	5.6
G2	Documenting all data type, format, range, and length.					
	Q3	Do all inputs are specified and documented with the data type, format, range, and length (e.g.: input name: loginID, type: text maxlength: 20)?				
		M3	All inputs are specified and documented with the data type, format, range, and length (Y/N).	Y	N	Y
G3	Improve the database security.					
G3 (A)	A: Database authentication.					
	Q4	What are the security tasks involved in account locking?				
		M4	Numbers of repeated failed login attempts allowed (maximum 3 times)	N	N	Y
		M5	The user's account is lock automatically or manually by the database administrator after a specified number of repeated failed login attempts (Y/N).	N	N	N
		M6	The user's account is unlocked automatically or manually by the database administrator (Y/N).	N	N	N

		M7	User's account logoff automatically if there is no activity within a specified period (Y/N).	N	Y	N
		M8	Automatic active user revalidation after a specified period (Y/N).	Y	N	N
		M9	Enforce user logoff and termination of all privileges for the user that has been transferred or retired or quit (Y/N).	Y	Y	N
		M10	The same user ID cannot be used for more than one session (Y/N).	N	N	N
		M11	User privileges are suspended after 30 days of inactivity and deleted after 30 days of suspension of use (Y/N).	N	N	N
	Q5	Are the password lifetime and expiration implemented?				
		M12	Password lifetime and expiration implemented (Y/N).	N	N	N
		M13	The lifetime period for passwords is restricted (Y/N).	N	N	N
		M14	The user must change the password during the grace period (Y/N).	N	N	N
		M15	The user's account is locked after the password expires (Y/N).	N	N	N
	Q6	Is the password history saved?				
		M16	Password history for each user is saved (Y/N).	N	N	N
		M17	The password is NOT re-used (Y/N).	N	N	N
	Q7	Is the password complexity verification applied?				
		M18	The minimum password is twelve characters in length (Y/N).	Y	N	Y
		M19	The password is NOT equal to the user ID (Y/N).	Y	N	Y
		M20	The password includes at least one alphabet character, one numeric character, and one punctuation mark (Y/N).	Y	N	Y
		M21	The password did NOT match any word on an internal list of simple words like welcome, account, database, user, and so on (Y/N).	Y	N	Y
		M22	The password differs from the previous password by at least three characters (Y/N).	N	N	Y
	Q8	Is additional password management applied?				
		M23	Password entry should be hidden on the user's screen. (e.g., on web forms use the input type "password") (Y/N).	Y	Y	Y
		M24	Password reset and changing operations require the same level of controls as account creation and authentication (Y/N).	Y	N	N
		M25	If using email-based resets, only send email to a pre-registered address with a temporary link/password (Y/N).	Y	N	N
		M26	Temporary passwords and links should have a short expiration time (Y/N).	N	N	N
		M27	Enforce the changing of temporary passwords on the next use (Y/N).	N	Y	N
		M28	Notify users when a password reset occurs (Y/N).	N	N	N
		M29	Enforce password changes based on requirements established in policy or regulation. Critical systems may require more frequent changes. The time between resets must be administratively controlled	N	N	N

Citation Information: Jakeri, M.B.M., Hassan, M.F., Sarlan, A., & Wahab, A.A. (2021). Security checklist evaluation in software development using goal question metric approach. *Journal of Management Information and Decision Sciences, 24*(S6), 1-12.

			(Y/N).			
		M30	Enforce password changes for the first-time login or after the password has been reset (Y/N).	N	Y	N
		M31	Disable "remember me" functionality for password fields (Y/N).	Y	Y	Y
		M32	The last use (successful or unsuccessful) of a user account should be reported to the user at their next successful login (Y/N).	N	N	N
G3 (B)	B: Database Authorisation					
	Q9	Is secure database management via web login interface implemented?				
		M33	Is the database management via web login interface performed (e.g., PhpMyAdmin) (Y/N)?	Y	Y	Y
		M34	Is there any policy permitting database management via web login interface (Y/N)?	N	N	Y
		M35	The access is restricted to a specific IP range (Y/N).	Y	N	Y
		M36	The default login URL is changed (Y/N).	N	Y	Y
		M37	HTTPS is enabled (Y/N).	N	Y	N
	Q10	Is secure database management implemented?				
		M38	Password protection to the database (Y/N).	Y	Y	Y
		M39	Disable root login to the database (Y/N).	Y	N	Y
	Q11	Do database privileges implemented?				
		M40	Database-specific privileges to database administrator is setup (Y/N).	Y	Y	Y
		M41	Database-specific privileges to the users are setup (Y/N).	Y	N	Y
G3 (C)	C: Additional database security					
	Q12	Is additional database security performed?				
		M42	Connection strings should NOT be hardcoded within the application (Y/N).	Y	N	Y
		M43	Connection strings should be stored in a separate configuration file on a trusted system (Y/N).	Y	N	Y
		M44	The connection string should be encrypted (Y/N).	N	N	Y
		M45	Close the connection as soon as possible (Y/N).	Y	N	Y
		M46	Remove or change all default database administrative passwords (Y/N).	Y	Y	Y
		M47	Utilize strong passwords/phrases or implement multi-factor authentication (Y/N).	Y	Y	Y
		M48	Disable any default accounts that are not required to support business requirements (Y/N).	Y	Y	Y
G4	Interconnectivity					
	Q13	Is interconnectivity applied?				
		M49	SSO is applied (Y/N).	N	N	N
		M50	Token-based authentication is applied (Y/N).	N	N	N
		M51	Cryptographic key sharing between applications is applied (Y/N).	N	N	N
		M52	Upstream and downstream compatibility of software should be explicitly designed (Y/N).	N	N	N
		M53	Authentication and authorization t to NAS connections are applied (Y/N).	Y	Y	N
		M54	NAS access is restricted to a specific IP	N	N	N

			range (Y/N).			

As shown in Table 2, Team 1 used the latest version of PHP for web-based application development compared to Teams 2 and 3 (M2). As for G2, Teams 1 and 3 documented and specified all inputs used in the application (M3) but not performed by Team 2.

G3 is about improving database security. G3(A) is more focused on database authentication. For Q4, Team 3 was the only team that restricted the number of repeated failed login attempts allowed to a maximum of three times (M4). The other two teams did not limit the number of failed login attempts to the developed web-based application. Although Team 3 applied M4, the user’s account was not locked automatically or manually by the administrator (M5). Therefore, M6 was also not employed. Teams 1 and 3 did not perform M5 and M6 since M4 was not applied. Team 3 applied M7 where the idle user was logoff automatically if there was no activity within a specified period. Although Team 1 did not implement M7, they revalidate the active user after a specified period in order to authenticate the user. Teams 1 and 2 terminated all privileges for the user which were transferred or retired or quit (M9) but were not implemented for the user’s account which was inactive after 30 days (M11). All teams did not restrict the use of user ID to one session only.

Q5 and Q6 were about password lifetime and expiration, and password history respectively, and were not implemented by all teams. Team 2 did not apply password complexity verification at all (Q7). Team 1 applied all the listed metrics except for M22 where the password did not differ from the previous password by at least three characters while team 3 applied all the metrics. Q8 is on additional password management for authentication. The response showed that M23 (hiding the password on the users’ screen) and M31 (disabling "remember me" functionality for password fields) were applied by every team whereas M26, M28, M29, and M32 were not applied by all teams.

G3(B) was more focused on database authorisation on the database connection (Q9) and privileges (Q10) management. All teams used PhpMyAdmin to manage the MySQL database (M33). Teams 1 and 2 have no policy permitting the database management via web login interface (M34) even though it was implemented. Team 1 has restricted the database connection to a specific IP range (M35), protected the database with a password (M38), and disabled root login to the database (M39), however, the default login URL was not changed (M36) and HTTPS was not enabled (M37). Team 2 has changed the default login URL (M36), enabled HTTPS (M37), however, it can be accessed from any IP address (M35). The database was protected with a password (M38) but used the default root login (M39). Team 3 applied all the metrics but did not enable the HTTPS to the PhpMyAdmin (M37). All teams have set up database administrator privileges (M40). Teams 1 and 3 have also set up the database privileges for the users (M41) but not applied by Team 2.

G3(C) was on the additional database security. Teams 1 and 3 applied all the listed security tasks except on encrypting the connection string (M44) for Team 1. Team 2 only executed certain metrics which were removed or changed all the default administrator password (M46), utilised strong password (M47), and disabled the default accounts that were not required to support business requirements (M48).

G4 was about interconnectivity with the aim to explicitly design the upstream and downstream compatibility of software that involved single sign-on (SSO)(M49), token-based authentication (M50), and cryptographic key sharing between applications (M51). However, all teams were not applying the interconnectivity. Therefore, the upstream and downstream compatibility of software was not applicable (M52). Teams 1 and 2 applied authentication and authorisation to Network Access Storage (NAS) connection (M53) but were not restricted to a specific IP range (M54). Team 3 was not provided with the NAS facility, therefore they gave ‘N’ as the response.

Citation Information: Jakeri, M.B.M., Hassan, M.F., Sarlan, A., & Wahab, A.A. (2021). Security checklist evaluation in software development using goal question metric approach. *Journal of Management Information and Decision Sciences, 24*(S6), 1-12.

DISCUSSIONS

TABLE 3
SUMMARISATION OF THE METRICS IMPLEMENTATION RATES

Metric	Team 1		Team 2		Team 3	
	Frequency	%	Frequency	%	Frequency	%
Yes	23	45.1	14	27.5	23	45.1
No	28	54.9	37	72.5	28	54.9
Total	51	100	51	100	51	100

In Table 3, the total number of the metric is 51 without including the M1, M2, and M33. M1 refers to the programming languages used to develop the in-house web-based application, M2 refers to the latest version of the programming languages being used while M33 is a query on database management via a web login interface.

The latest PHP version to date is PHP 8.0 released on 26 November 2020 (PHP Group, 2021). Team 1 was using PHP 7.3 where the security support ending on 6th December 2021 while PHP version lower than 7.3 was no longer supported by the PHP group. Therefore, Teams 2 and 3 should upgrade the PHP version as soon as possible, as they may be exposed to unpatched security vulnerabilities.

The percentage of security tasks implementation for additional design was less than 50%. The percentage rate for Team 1 and 3 were 45.5% while team 2 was 27.5%. Even though Teams 1 and 2 needed to apply the listed security tasks to the application that has been developed, Team 1 was able to integrate more security tasks in the application compared to Team 2. However, the low implementation was also faced by Team 3 although the application was still in the requirement phase during the first meeting held. Figure 2 shows the percentage of the implementation of the metrics with response ‘Y’ on the goals.

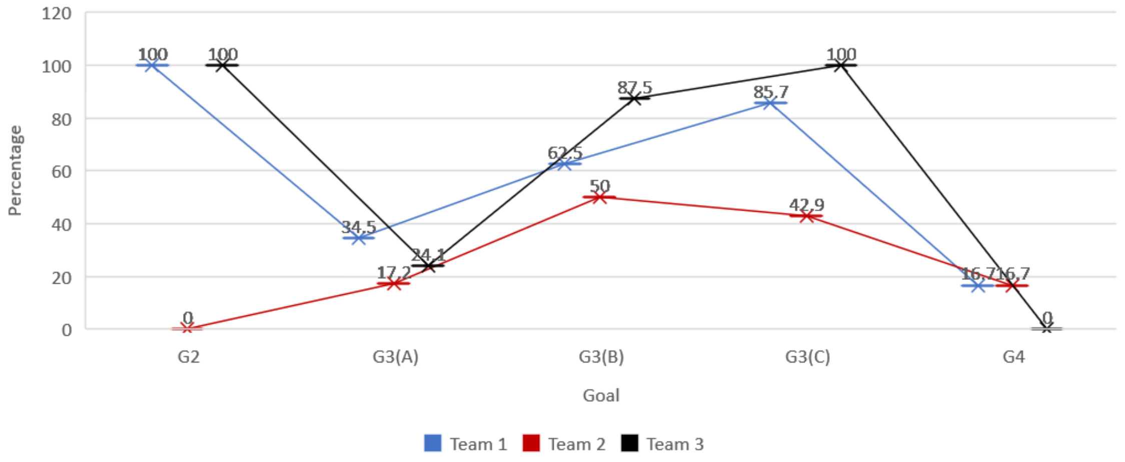

FIGURE 2
PERCENTAGE OF THE METRICS WITH ‘Y’

The above figure shows that the implementation of the metrics for G2 was 100% for Teams 1 and 3 as they documented and specified all the inputs used in the web-based application development. The implementation in database authentication (G3(A)) was very low, and immediate attention should be given. Database authentication is crucial to ensure that externals (e.g., human actors and external applications) are who or what they appear to be and, as a result, preventing security breaches by the impostor (Firesmith, 2003).

The database authorisation (G3(B)) implementation was more than 50% for all teams. This indicated that the teams, especially Teams 1 and 3 knew the importance of limiting the access and privileges to the authenticated users only. As for additional database security

(G3(C)), Team 1 has applied 85.7% of the security tasks. The only security task which has not been implemented was M44 (connection string should be encrypted). Team 3 successfully performed G3(C). However, the percentage of implementation by Team 2 was only 42.9%. The team only managed to implement three out of seven of the listed security tasks.

Teams 1 and 2 have the Network Access Storage (NAS) provided to store data. Although the authentication and authorisation connections were applied, the users were able to connect to NAS from anywhere in the network since the NAS access was not restricted to a specific IP address.

CONCLUSION

This paper has presented the GQM approach to define and develop, measure, and evaluate the security checklist for the selected security activity which is an additional design in the design phase. GQM was used to construct the security goal and the questions and metrics were defined from the literature. The security checklist has been used in the case studies and performed by the teams in the given period. The results revealed the low implementation of the listed security task. Therefore, a more planned and realistic development timeline and the ability to focus on the application development were suggested by the teams to increase the percentage of the security tasks implementation.

ACKNOWLEDGEMENT

The authors would like to thank the Malaysian Administrative Modernisation and Management Planning Unit (MAMPU), Universiti Teknologi Petronas (UTP), and the Malaysian public sector for supporting this research.

REFERENCES

Mougoue, “Software integrity, SSDLC 101: What is the secure software development life cycle?,” (2016).

Mohaddes, H., & Tabatabaei, I. (2015). “Effects of software security on software development life cycle and related security issues,” *Int. J. Comput. Intell. Inf. Secur.*, *6*(8), 4–12.

Khan, M.U.A., & Zulkernine, M. (2009). “A survey on requirements and design methods for secure software development,” *Ontario*.

Microsoft Corporation, *Simplified implementation of the SDL*. Microsoft Corporation, (2010).

Cyber Security Malaysia, “Cyber security guideline for secure software development life cycle (SSDLC).”(2019), 1–60.

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

Basili, V.R., Caldiera, G., & Rombach, H.D. (1994). “The goal question metric approach,” *Encycl. Softw. Eng., 2*, 528–532.

Lindström, B. (2004). “A software measurement case study using GQM,” *Communication*, 72.

Yahya, R.J., Walters, & Wills, G.B. (2019). “Using Goal-Question-Metric (GQM) approach to assess security in cloud storage,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, 223–240.

Ampatzoglou, E.M., Arvanitou, A., Ampatzoglou, P., Avgeriou, A.A., Tsintzira, & Chatzigeorgiou, A. (2020). “Architectural decision-making as a financial investment: An industrial case study,” *Inf. Softw. Technol., 129,* 106412.

Gosain, & Singh, J. (2020). “Comprehensive complexity metric for data warehouse multidimensional model understandability,” *IET Softw, 14*(3), 275–282.

Leshob, A.B., & Renard, L. (2018). “Towards a process analysis approach to adopt robotic process automation,” *Proc. - 2018 IEEE 15th Int. Conf. E-bus. Eng. ICEBE* 46–53.

Ergasheva, S., Kruglov, A., & Shulhan, I. (2019). “Development and evaluation of GQM method to improve adaptive systems,” *CEUR Workshop Proc. 2525*.

Roseberry, K., Sheppard, M.A., Wallis, R., & Yang, Y. (2019). “Does every formal peer review really need to take place? An industrial case study,” *Procedia Comput. Sci., 153,* 45–54.

Halabi, T., & Bellaiche, M. (2017). “Towards quantification and evaluation of security of cloud service providers,” *J. Inf. Secur. Appl, 33,* 55–65.

Mansol, N.H., Alwi, N.H.M., & Ismail, W. (2016). "Managing organizational culture requirement for business continuity management (BCM) implementation using Goal-Question-Metric (GQM) approach," *J. Teknol*, *78* (12–3), 13–22.

OWASP Foundation Inc, "OWASP secure coding practices quick reference guide version 2.0," (2010). 1–17.

ORACLE Corporation, "Database security guide," *ORACLE Corporation*, 2020.

MAMPU, "Information technology instructions." (2007). 61, Crossref , Google scholar , Indexed at

Saive, R. (2016). "4 useful tips to secure PhpMyAdmin login interface," *TecMint*,

The PHP Group, "Supported Versions," 2021. https://www.php.net/supported-versions.php (accessed Mar. 23, 2021).

Firesmith, G. (2003). "Engineering security requirements," *J. Object Technol*, *2*(1), 53–68.

Received: 14-Dec-2021, Manuscript No. JMIDS-21-7746; **Editor assigned:** 16-Dec-2021, PreQC No. JMIDS-21-7746(PQ); **Reviewed:** 27-Dec-2021, QC No. JMIDS-21-7746; **Revised:** 07-Jan-2022, Manuscript No. JMIDS-21-7746(R); **Published:** 14-Jan-2022