Computers & Security 144 (2024) 103985

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

Adopting security practices in software development process: Security testing framework for sustainable smart cities

Yusuf Mothanna [a], Wael ElMedany [a], Mustafa Hammad [b], Riadh Ksantini [a], Mhd Saeed Sharif [c,*]

[a] College of Information Technology, University of Bahrain, Bahrain
[b] Department of Computer Science, Mutah University, Jordan
[c] Intelligent Technologies Research Group, Computer Science and Digital Technologies, ACE, UEL, London, UK

ARTICLE INFO

Keywords:
Smart city application
Software development process
Security testing
Security practices

ABSTRACT

The dependence on smart city applications has expanded in recent years. Consequently, the number of cyberattack attempts to exploit smart application vulnerabilities significantly increases. Therefore, improving smart application security during the software development process is mandatory to ensure sustainable smart cities. But the challenge is how to adopt security practices in the software development process. There are Several established and mature security testing frameworks exist that consider security requirements and testing during Several already established and mature security testing frameworks exist that consider security requirements and testing during Software Development Life Cycle (SDLC), but there is a unique challenges posed by smart city applications and the need for a comprehensive approach to address the evolving threat landscape in this context. This paper proposed a framework that adopts security testing practices in all phases of the software development process. The proposed framework identifies several security activities and steps that can be applied in each phase of the software development process.

1. Introduction

Nowadays, information and communication technology and smart city implementation play vital roles in business and daily life. Organizations and individuals depend on smart applications to achieve their goals in many aspects of life. Nevertheless, the proliferation of smart applications encourages cyber hackers to discover and exploit the vulnerabilities in software applications to perform malicious acts and cause harmful effects and impacts. The risks posed by hackers have increased tremendously in recent years. Organizations are thriving to handle these threats and confront these challenges. Therefore, enhancing smart city application security is essential to overcome the difficulties and maintain the implementation of smart cities. However, organizations have released that it is important to build proper defenses to improve software application security.

One of the biggest challenges organizations realize is integrating security practices in the software development process to improve security. Security practices should be considered and implemented during the development process from the first phase. In the traditional software development approaches, security practices are normally introduced in the final stages. This approach consists of various problems such as increasing the possibilities of generating additional costs, raising development times, and decreasing the level of security protection of the smart application.

Integrating proper security practices and activities in all phases of the software development process is necessary. The costs, time, and efforts of detecting and maintaining vulnerabilities in the last phases exponentially increase. Also, implementing security from the earlier stage of the software development process makes the application less vulnerable and more secure. Therefore, how to adopt security practices in the software development process?

While established frameworks like SAMM are indeed robust, they may not be tailored to the specific requirements and intricacies associated with smart city development. The main contribution of this work is a proposal of a new security testing framework for the software development process. The proposed framework aims to improve the security of smart city applications by considering security practices in the development process.

Our framework aims to fill this gap by providing a specialized approach that integrates security testing practices seamlessly into every phase of the software development process, ensuring the security of smart city applications. The proposed framework has four main phases of applying security testing for the smart application. The first phase intends to define security testing goals and enhance the security knowledge of the team. Then, identify security guidelines by applying risk assessment and analyzing security requirements. The third

* Corresponding author.
E-mail address: s.sharif@uel.ac.uk (M.S. Sharif).

https://doi.org/10.1016/j.cose.2024.103985
Received 1 March 2023; Received in revised form 8 June 2024; Accepted 2 July 2024
Available online 6 July 2024

phase focuses on executing the security testing according to security guidelines and analyzing testing results. The final phase describes the security activities to ensure continuous improvement of software application security. In addition, the study compares and discusses some common useful secure software development models. The following sections involve a background and related work, an overview of the proposed security testing framework for the software development process, outcomes of the proposed framework phases, security testing and discussion, and a conclusion.

2. Background and related works

Protecting confidential data and maintaining the integrity and availability of smart cities' services are essentials hence it is vital to incorporate effective security and privacy considerations within the smart cities design to maintain their sustainability (Haque et al., 2022; Kaluarachchi, 2022; Rao and Deebak, 2022). Several research works discussed the updated security challenges related to smart applications. For example, the research study (Cui et al., 2018) presents several approaches and techniques to protect smart applications (Al-Qarafi et al., 2022; Kaushal et al., 2022; Rana et al., 2023). Also, the authors consulted that the rapid development of smart city applications is required more efforts and studies to mitigate the security challenges of smart cities.

Moreover, the research work in Ijaz et al. (2016) presented security threats, vulnerabilities, and controls for smart cities as well as analyzed security best practices of smart cities. The work in Ullah et al. (2021) proposed a risk management framework based on technology, organization, and environment to assess and manage the risks related to smart cities. The authors presented a collection of significant risks and identified appropriate treatments to enhance citizens' safety, security, and privacy. The study mentioned some risks associated with smart city applications. Therefore, securing the smart city applications during the software development process is important.

Many models and standards have been studied and compared to improve security in software applications (Taheri et al., 2023; Khan et al., 2022; Ogbodo et al., 2022). Kara (2012) examined the Common Criteria (CC) secure software development approach with other security development models. The author proposed that CC is an international Information Technology security standard (ISO/IEC 15408) and can be implemented as developers' secure software development lifecycle guidance.

However, the author suggested that adding some security functions could increase the security maturity of software development, such as the law, policy, procedure, and compliance. The researcher in Faizi and Rahman (2019) explained security integration in the software development process and various secure software development frameworks. Also, the authors provided guidelines for selecting the best framework. The paper concluded that based on the organization's characteristics, the development team must choose the best-fit methodology to be implemented. In Sharma and Misra (2017), authors explained several secure software developments models. The work discussed the model's methodologies to protect software application and their advantages and limitations.

Various studies are proposed new frameworks or models for securing software development (Akbar et al., 2022; Ghelani et al., 2022; Ansari et al., 2022; Taherdoost, 2022). Authors in Núñez et al. (2020) analyzed the available security software development models and proposed a new methodology for securing software applications which is Viewnext-UEx model. The proposed model is assessed with a real environment, which reduces vulnerabilities detection by 66 percentage. The authors validated that using the new methodology is improved the security and quality of software applications.

In Tung et al. (2016), authors proposed an integrated security testing framework to secure the software development process in each phase. The framework used security practices and activities to generate guidelines and integrated security testing tools to provide testing services then improve testing services based on the testing result analysis. The authors tested the framework and constructed a prototype system that integrated various security testing activities and tools. The results indicated that the framework could provide stable services with acceptable quality and perform efficient services during the software development process.

Other studies consider agile perspective in secure software lifecycle to enhance the protection and defense of software applications (Valdés-Rodríguez et al., 2023; Khan et al., 2022). Proposed research in de Vicente Mohino et al. (2019) described a new software development model that defined security practices in the development lifecycle by taking agile methodologies. The new models considered the security aspects in each development phase to detect and solve the vulnerabilities without extra time and cost. The paper described and compared the current secure software development lifecycle models at security activities, resources, artifacts, agile properties, and use in the software industry. The authors concluded that the model improved the software's quality and security, increased the added value to the software, and involved each team member in the process.

In Rindell et al. (2018), authors described the software development security approaches and mapped them with agile software development methods. The study provided a framework for adopting security practices with agile processes to align the security objective in the software development process.

Adopting security testing in the software development process ensured application protection. Authors in Mahendra and Khan (2016) reviewed security testing frameworks, methodologies, and techniques in each phase of the software development lifecycle. In addition, the article mentioned that most security testing methods are applied in different phases of the software development process. Also, security testing can be performed in design phased before the implementation. The paper concluded a need for a security testing framework process at the design phase of the software development process. On the other hand, authors in Lingham et al. (2020) discussed the techniques and approaches to define and eliminate software applications vulnerabilities. Also, the works presented in detail security features incorporated with the software development process, which is often ignored during the implementation.

Several secure software development lifecycle models are proposed to improve software application security. This paper reviewed three common known models, which are Microsoft Security Development Lifecycle (Microsoft SDL), Software Assurance Maturity Model (SAMM) from Open Web Application Security Project (OWASP), and Building Security In Maturity Model (BSIMM). Microsoft (2021) introduced security practices in all phases of the software development process, including guidance, best practices, tools, and techniques to reduce development costs.

The OWASP (Software, 2021) is an online community and proposed SAMM model with a collection of security practices mapped to software development functions. The model defined four business functions which are governance, contraction, verification, and operations. In addition, each business function has a set of security practices to reduce the risks of software applications. Building Security In Maturity Model (BSIMM) (Building, 2021) is a group of firms from different industries that are participated in studying real-world software security projects. BISMM proposed a security framework to assess the organization and prioritize the change required to increase the security maturity of the software development process. The framework is outlined in four domains, 12 practices, and 122 activities. Table 1 presents a comparison between security practices of Microsoft SDL, SAMM, and BISMM.

Based on (Microsoft, 2021; Software, 2021; Building, 2021) review, all models considered the security practices and activities in each phase of the software development process. The SAMM and BSMM models mentioned the policies, laws, regulations, and strategic direction in the

Table 1
Security practices of Microsoft SDL, SAMM, and BSIMM on software development process.

Software development phase	Microsoft SDL	SAMM	BSIMM
Requirements		- Policy and compliance	- Compliance and policy
	- Provide training	- Education and guidance	- Training
	- Define security requirements	- Security requirements and security architecture	- Architecture analysis
	- Define metrics and compliance report	- Strategy and metrics	- Strategy and metrics
Design	- Perform threat modeling	- Threat assessment	- Attack models
	- Establish design requirements	Design review	- Security features and Design
	- Define and use cryptography standards		
Implementation	- Perform Static Analysis Security Testing (SAST)	- Implementation review (Code Review)	- Code Review
	- Use approved tools and design review		
	- Manage the security risk of using third-party components		
Testing	- Perform Dynamic Analysis Security Testing (DAST)	- Security testing	- Security testing
	- Perform penetration testing		- Penetration testing
Maintenance	- Establish a standard incident response process	- Issue management and operational enablement	- Configuration management and vulnerability management
		- Environment hardening	- Software environment

requirements phase, but Microsoft SDL did not state them directly. The training, security requirements, compliance, and metrics are considered in all models. The security practices and activities are mostly similar in all models in the design phase. In the Implementations phase, Microsoft SDL is focused on risk management of the third party, but the security practices and activities are mostly similar in all models. Microsoft SDL in the testing phase focused on implementing dynamic analysis security testing. Security testing in SAMM is based on the defined security requirement and the organization-wide baseline.

In BSMM, security testing is performed based on the needs and features following the quantality assurance process. The testing depends on the white box and fuzzy testing in this phase. All models prefer to perform penetration testing and use automated tools. Managing incident response helps address new threats that can emerge over time, and all models considered incident response management in the maintenance phase. SAMM and BSMM are focused on the environment and operation management, increasing the protection level by little damage in cased o vulnerabilities are exploited and ensuring the information quality and availability according to expectation.

Based on the three models review and descriptions, this paper proposed the security testing framework of the software development process. In addition, the article identified security requirements, including practices, activities, and tools in each phase of software development.

3. An overview of the proposed security testing framework of software development process

Security holds a significant role in the software development process and security testing is essential to secure software applications to prevent data breaches and security attacks.

3.1. Rationale for developing a new framework

This subsection aims to provide a comprehensive rationale for the development of our proposed framework. It delves into the specific challenges posed by smart city applications and justifies the necessity for a specialized approach. By addressing the limitations of existing frameworks, particularly SAMM, we aim to highlight how our framework bridges crucial gaps in ensuring the security of smart city software development.

3.1.1. Challenges in smart city applications

The expansion of smart city applications has brought forth unique challenges in terms of security. The interconnected nature of these applications and the increasing number of cyberattack attempts make conventional security testing frameworks insufficient. Smart city environments demand a more specialized and integrated approach to address the evolving threat landscape effectively.

3.1.2. Limitations of existing frameworks

While established frameworks like SAMM have proven effective in conventional software development, they may not adequately cater to the intricacies of smart city application security. Our analysis of SAMM and other frameworks revealed certain limitations, including:

- Lack of Tailored Approach: Existing frameworks may lack a tailored approach that comprehensively integrates security testing practices into all phases of the smart city software development process.
- Inadequate Coverage of Smart City-specific Risks: Smart city applications face unique risks that stem from their interconnectedness, reliance on IoT devices, and exposure to diverse data sources. Conventional frameworks may not sufficiently address these specific risks.

3.1.3. Bridging gaps with our framework

Our proposed framework is designed to overcome the identified limitations by providing:

- Comprehensive Integration: Unlike existing frameworks, our approach ensures the integration of security testing practices from the initial stages of development, covering all phases of the software development life cycle (SDLC).
- Smart City-specific Considerations: The framework takes into account the unique challenges posed by smart city applications, offering tailored security measures to mitigate specific risks associated with interconnected systems and IoT devices.
- Early Risk Mitigation: By embedding security testing in the early phases of development, our framework aims to proactively identify and mitigate potential security risks, reducing the likelihood of vulnerabilities persisting into the final product.

3.2. Proposed security testing framework

This paper proposes a security testing framework for the software development process. Based on the proposed framework, security testing is considered from an early stage of the development process. Fig. 1

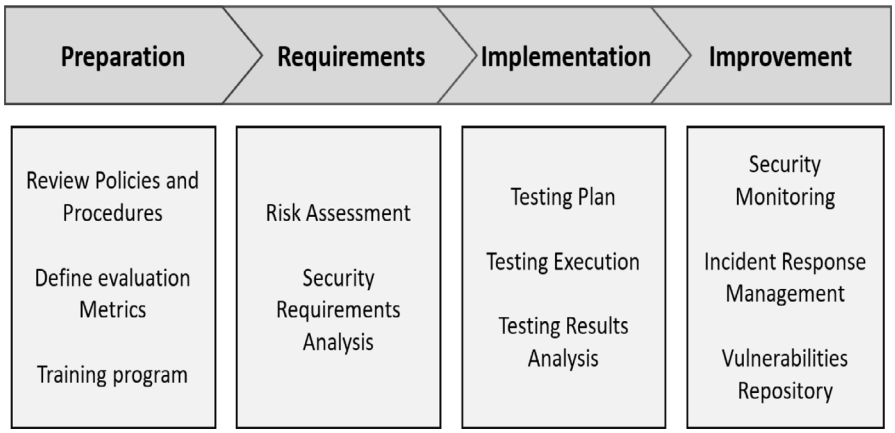

Fig. 1. An overview of security testing framework.

presents an overview of the main phases and practices of the proposed security testing framework.

The following is a description of each phase and its practices of the proposed framework.

3.2.1. Preparation phase

The preparation phase is essential to identify the security testing goals for the software development process and enhance the knowledge of software development teams. This phase involves three security practices which are reviewing policies and procedures, defining evaluation metrics, and implementing a training program (ISO, 2021a).

1. ***Review Policies and Procedures***
   Reviewing security policies and procedures can be conducted by implementing the following functions:

   - Emphasize alignment with smart city regulations and industry standards, such as PCI, DSS, HIPPA, and GDPR.
   - Specify how the review integrates smart city-specific security needs and objectives.
   - Define and assess the related policies, regulations, and standards.
   - Discuss the assessment results with other groups, such as legal, quality, and strategies.
   - Share the results with the top management for approval.

   Reviewing security policies and procedures is important to understand the legal and regulations requirements and to be aligned with security strategy objectives. Likewise, This review helps to consider the industry best practices such as Payment Card Industry Data Security Standard (PCI DSS), Healthcare Insurance Portability and Accountability Act. (HIPAA), and General Data Protection Regulation (GDPR) in the security testing goals.
2. ***Define Evaluation Metrics***
   Defining evaluation metrics for security testing can be performed by:

   - Tailor evaluation metrics to reflect smart city security goals and Key Performance Indicators (KPIs).
   - Highlight criteria relevant to the unique aspects of smart city software development.
   - Identify the Key Performance Indicator (KPI) of security goals.
   - Define the evaluation criteria of security tasks.
   - Execute a tracking system to ensure security tasks are completed and generate accurate reports.

   Define evaluation metrics aid in ensuring that the security goals are understood and applied during the entire development process.
3. ***Training program***
   A security training program has to be developed for the development teams by following steps:

   - Integrate examples and case studies related to smart city application security.
   - Emphasize the distinctive aspects of attack vectors and defense strategies in smart city contexts.
   - Create a training path of the team based on roles and responsibilities.
   - Conducted security awareness workshops or e-learning approaches as a baseline for everyone.
   - Perform training on risk assessment methodology and security testing tools.
   - Develop a portal that involves resources of protecting software such as white papers, tools guidelines, training materials, and updated vulnerabilities.

   Software development teams must know the security baseline and develop secure software. In addition, the team should understand the attack techniques and how to fix vulnerabilities. Providing training in the initial phase reduces software development mistakes. Also, the proper training increases the security Knowledge and skills of development teams, which minimizes software maintenance.

3.2.2. Requirements phase

The requirements phase helps establish security guidelines and analyzes the requirements based on business functions, security best practices, and risks assessment results. This phase consists of two steps which are risk assessment and security requirements analysis.

1. ***Risk assessment***
   Risk assessment is the process of identifying, assessing, and reducing risks to an acceptable level. This phase is crucial for smart city applications and involves comprehensive planning and execution to understand potential threats and vulnerabilities specific to this context (ISO, 2021c; NIST, 2021):

   - Explicitly state considerations for smart city assets, including IoT devices, interconnected networks, and citizen data.
   - Align risk criteria with the unique characteristics of smart city ecosystems.
   - Create a comprehensive plan of risk assessment that includes the following:

  - **Set criteria and risk appetite:** Establish criteria for prioritization, considering the unique risk appetite for smart city applications.
  - **Define scope and boundaries:** Determine the functions and people involved, allocating resources effectively for better optimization.
  - **Identify roles and responsibilities:** Clearly define roles and responsibilities of individuals involved in the risk assessment process.

- Conduct risk assessments, which include the following steps:

  - **Assets Identification:** Identify and categorize information assets critical to the smart city application, including but not limited to records, physical devices, software, communications services, general utilities, people, qualifications, skills, and experience.
  - **Risk Identification:** Define and extract a list of risk scenarios and consequences related to identified assets and functions, considering the comprehensive identification of assets, threats, and vulnerabilities.
  - **Risk Analysis:** Determine the likelihood of identified risk scenarios occurring, considering the causes and sources of risk and their potential impact.
  - **Risk Evaluation:** Classify and prioritize risk scenarios based on their severity and potential impact on the smart city application.

Risk assessment is the core step of the requirement phase. The risk assessment must be prepared carefully and accurately, especially since its results will reflect the whole security testing status. The risk assessment results help to identify appropriate actions and priorities for security testing of the software development process.

2. ***Security Requirements Analysis***
Security requirement analysis is essential for identifying security testing practices and activities specific to the software development process of smart city applications. Implementation steps include:

   - **Study risk assessment reports and security best practices:** Analyze risk assessment reports, incorporating insights into potential key security issues.
   - **Perform threat modeling:** Specify additional security issues through a proactive threat modeling process.
   - **Review legal and industry standards:** Consider legal requirements and industry standards, especially those relevant to smart city applications.
   - **Specify security requirements:** Develop security requirements based on the identified security issues and standard requirements, emphasizing proactive practices and enumerating security testing techniques.

The security requirements emphasize proactive practices by identifying security risks then enumerating security testing techniques.

3.2.3. *Implementation phase*

The implementation phase aims to execute the security testing on the software development processes from the initial stage according to the security guidelines. The results of the implementation phase are testing reports that include the methodologies, tools, findings, and correction status. This phase involves three steps which are testing plan, testing execution, and testing results analysis:

1. ***Testing plan***
The testing plan is developed according to security requirements and guidelines. The plan is documents describing the methodologies and approaches of software security testing. The following steps assist in building the plan documents (Radack et al., 2008):

   - Include examples and scenarios relevant to smart city applications.
   - Ensure that the testing plan reflects the diversity of security challenges posed by interconnected urban systems.
   - Comprehend the security specifications and standards outlined in the guidelines, and outline the testing workflow in detail.
   - Define the severity of the testing finding and actions priority.
   - Set the testing instructions and define the testing approaches and tools categorized into automatic, semi-automatic, manual, and code review.
   - Build the testing cases based on the security guideline. Each test case is associated with a specific security guideline.

The testing plan is the first step of the implementation phase to ensure that all security guidelines are considered during the testing. The testing plan details testing workflows, instructions, tools, and testing cases. Also, the well-developed plan provides the efficiency and effectiveness of the security testing execution.

2. ***Testing execution***
The major step of the implementation phase is the testing execution. The execution aims to inspect the software application during the development process to find security problems. The following activities help to achieve the execution intentions (Jammeh, 2020):

   - Specify smart city-specific security testing approaches addressing IoT vulnerabilities, dynamic network configurations, and challenges of real-time data processing.
   - Perform the testing case in each phase of the software development process and integrate the approved testing tools.
   - Apply the below security testing to identify the vulnerabilities in the software which are Static application security testing (SAST), Dynamic application security testing (DAST), vulnerability scanning, penetration testing, configuration management, compliance and infrastructure as code.

Security testing approaches should be considered in various phases of software development. The testing helps reduce software vulnerabilities and improve the quality of the software.

3. ***Testing results analysis***
Different security testing approaches are applied in the testing execution step, generating various testing results. Therefore, combining and comparing the testing results are helpful to solve the problems effectively in different dimensions. The below are the main steps of the analysis (Radack et al., 2008):

   - Emphasize the relevance of analysis criteria to smart city security concerns.
   - Consider incorporating data analytics techniques tailored for the intricacies of smart city applications.
   - Define the objective of the security testing analysis and specify the criteria for comparing the testing result.
   - Store the testing results in a repository and integrate various testing results in a converged report.
   - Identify the new security issues according to data mining or statistics tools and develop an action list to solve the new issues.

Analysis of the security testing result improves the vulnerabilities detection of the software. This practice performs test results with severity and recommends fixing the problems. Moreover, the analysis results enhance the quality of security testing execution.

3.2.4. Improvement phase

Security testing is a continuous process. Therefore, the improvement phase is required to find unknown security issues. This phase involves three steps, which are security monitoring, incident response management, and vulnerabilities repository:

1. ***Security monitoring***
   Security monitoring is an automatic tool that gathers and inspects indicators of potential threats. The following steps help to develop security monitoring for the software environment (Dempsey et al., 2011):
   - Highlight the distinctive features of security monitoring in smart city environments, including real-time threat detection across diverse components.
   - Determine the requirements by identifying objectives, priorities, and workflows.
   - Implement an appropriate security monitoring tool based on the defined requirements.

   Monitoring the software environment is necessary to detect attacks immediately and support operation security and incident response.
2. ***Incident response management***
   The incident response focuses on handling security events to increase the efficiency of reactions rather than uninformed response. The following steps are helpful to establish incident response management (ISO, 2021b):
   - Address the dynamic nature of smart city threats and responses, ensuring that incident response plans are agile and adaptive.
   - Develop an incident response plan to address new threats.
   - Identify the incident response team and points of contact for security issues.
   - Establish incident response process, protocols, and communication channels.
   - Conduct the root cause analysis of the security issues and prepare incident response reports.

   The quick detection and response of security events minimize the significant impact of the software application. In addition, incident details are dissected to enhance the security posture of the software application.
3. ***Vulnerabilities repository***
   The vulnerability repository is a database of vulnerabilities developed and improved based on the implemented security practices during the software development process. The following activities assist in developing a vulnerability repository:
   - Emphasize continual learning from security events specific to smart city applications to proactively enhance security measures.
   - Review and analysis the new threat details from security monitoring and incident response reports.
   - Review the most common vulnerabilities from another repository.
   - Develop the repository to improve the software application security.

   Vulnerability repository focuses on learning from the security failure and error to improve the security measures and become proactive rather than reactive. The improvement phase is a continuous practice of securing the software development process that efficiently resolves vulnerabilities.

4. Outcomes of security testing framework

The proposed security testing framework has four main outcomes which security goals, security guidelines, security testing reports, and enhancement reports. The outcome of each phase of the proposed framework can be used as input of the next phase, as shown in Fig. 2.

Fig. 2 presents the security goals as an outcome of the preparation phase, security guidelines as an outcome of the requirements phase, security testing reports as an outcome of the implementation phase, and enhancement reports as an outcome of the Improvement phase.

1. **Security goals**
   The preparation phase focuses on defining the security goals of the security testing. In addition, this phase supports specifying a matrix with clear evaluation criteria and security goals KPI. Developing an evaluation metric is essential to define the acceptance criteria and achieve the security testing goals. Also, this phase aims to build team skills in security knowledge.
   Well-define security goals enhance specifying the requirements of security testing. Therefore, the security goals are the main input in the second phase of the framework, which is the requirements.
2. **Security Guidelines**
   The outcome of the requirements phase is security guidelines that include security methodologies, functions, technique specifications, tools, and coding practices to combat security issues. According to risk assessment and requirements analysis, understanding security issues is necessary to identify proper security guidelines.
   The appropriate security guidelines support applying effective security testing. Consequently, security guidelines consider as the most important source of the implementation phase.
3. **Testing Reports**
   The outcome of the implementation phase is security testing reports. The testing reports depend on a proper testing plan and effective testing execution. According to the testing execution results analysis, the testing reports are produced with details of security issues with actions to solve the problems. In addition, the testing reports include descriptions of challenges that appeared during the security testing implementation.
   The security testing reports are one of the main inputs of the next phase, which is the improvement phase. The reports help create monitoring use cases, develop an incident response plan, and build the vulnerabilities repository.
4. **Enhancement Reports**
   The outcome of the improvement phase is enhancement reports. The reports can be generated by getting information from three sources which are continuous security monitoring detections, incident response details, and vulnerabilities repository. The enhancement reports help mitigate future failures and errors. In addition, the reports support vulnerabilities reduction and proactive actions.
   The enhancement reports improve all previous phases of the proposed framework. The reports are important for current and future software applications to increase security posture.

5. Security testing and discussion

Performing the proposed security testing framework has many advantages during the software development process. One of the framework's main strengths is the continuous integration of security testing

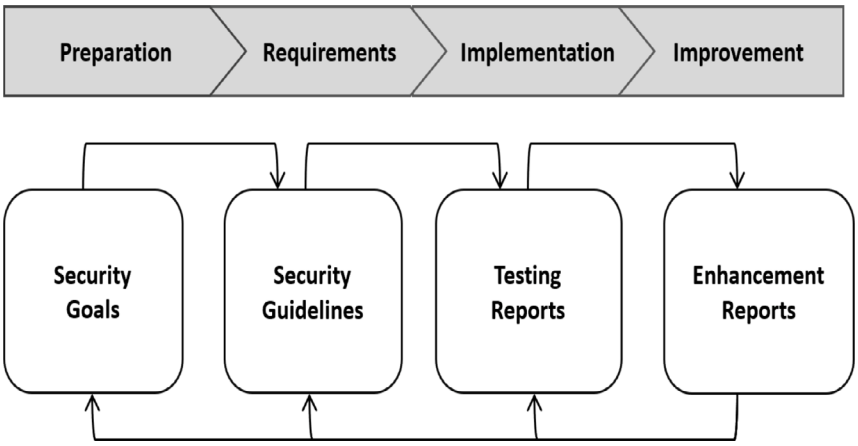

Fig. 2. Outcomes of security testing framework phases.

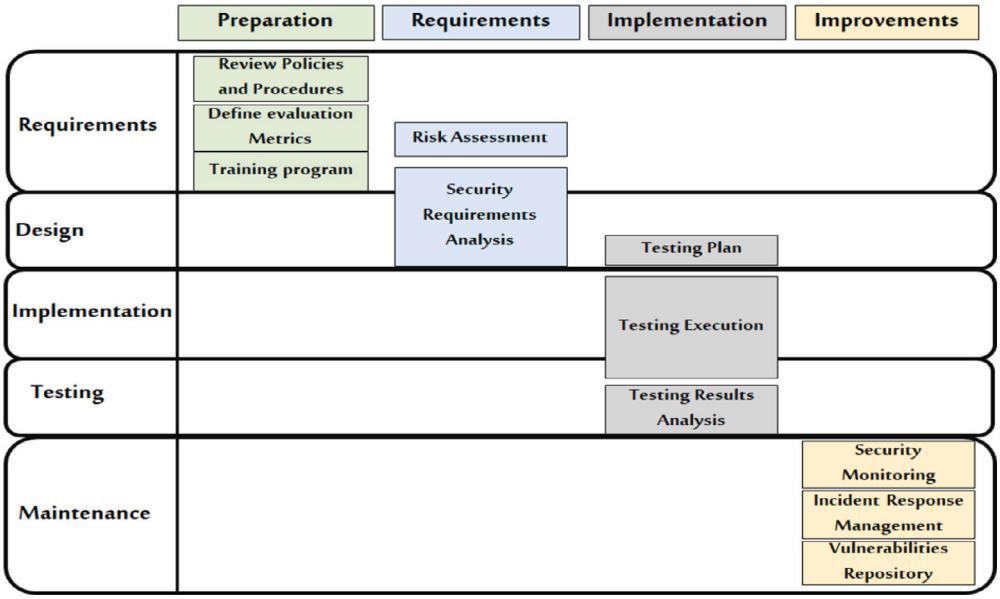

Fig. 3. Integration of proposed security testing practices with software development phases.

with all phases of software development. Fig. 3 describes the integration of security testing framework practices in software development phases.

As shown in Fig. 3, the software development phases are listed on the left side, and the proposed security testing phases are introduced at the top. Then, the proposed security testing practices are mapped with the software development phases, demonstrating integrating the testing practices with the development process. Therefore, all practices in the preparation and requirement phase of the testing framework are integrated with the requirements phase of software development except security requirements analysis which is between the requirements and design phases of software development. Also, the implementation phase practices of testing frameworks can be integrated with the design and implementation phase of software development where testing plan with implementation phase, testing execution with implementation and testing phase, and testing results analysis with the testing phase. Finally, the improvement phase practices are integrated with the maintenance phase of software development.

Other strengths of the security testing framework are flexibility and improvement. The proposed framework has many activities to define the security goals and requirements to customize the security testing. Furthermore, those activities are important for developing security testing approaches, for example, considering new techniques and technology. Therefore, the proposed framework considers flexibility to achieve the efficiency and effectiveness of security testing. On the other hand, the last phase of the framework focuses on improvement. Security improvement is a continuous process and essential to ensure high performance of security testing. Security monitoring, incident response, and vulnerabilities repository are activities that aim to enhance the quality of the software from the security perspective and keep feeding other security practices with new threats and known vulnerabilities.

The limitation of the proposed framework lies in the absence of experimental evaluation in this paper. As a potential avenue for future research, conducting experiments to validate the framework is recommended.

6. Conclusion

In the last few years, the implementation and dependency of smart city applications are increased. Consequently, improving application security is mandatory to ensure sustainable smart cities. Usually, security testing practices are considered in the last stages during the development process. This paper proposes a new security testing framework to develop secure applications considering all phases of the development process. The proposed framework adopts various security actions to define the security goals, develop security guidelines, execute security

testing and continuously improve the security of software applications. This paper has reviewed and compared many well-known secure software development lifecycle models. This study does not evaluate the new framework by applying an experiment. Therefore, future work can evaluate the proposed framework with an experiment to measure efficiency and study the applicability in cloud pipelines.

CRediT authorship contribution statement

Yusuf Mothanna: Writing – original draft, Software, Formal analysis, Data curation, Conceptualization. **Wael ElMedany:** Writing – review & editing, Validation, Project administration, Methodology, Investigation. **Mustafa Hammad:** Writing – review & editing, Visualization, Methodology. **Riadh Ksantini:** Writing – original draft, Visualization, Supervision, Investigation. **Mhd Saeed Sharif:** Writing – review & editing, Supervision, Resources, Project administration, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

Akbar, M.A., Smolander, K., Mahmood, S., Alsanad, A., 2022. Toward successful DevSecOps in software development organizations: A decision-making framework. Inf. Softw. Technol. 147, 106894.

Al-Qarafi, A., Alrowais, F., S. Alotaibi, S., Nemri, N., Al-Wesabi, F.N., Al Duhayyim, M., Marzouk, R., Othman, M., Al-Shabi, M., 2022. Optimal machine learning based privacy preserving blockchain assisted internet of things with smart cities environment. Appl. Sci. 12 (12), 5893.

Ansari, M.T.J., Pandey, D., Alenezi, M., 2022. STORE: Security threat oriented requirements engineering methodology. J. King Saud Univ.-Comput. Inf. Sci. 34 (2), 191–203.

2021. Building security in maturity model. https://www.bsimm.com/framework.html. (Accessed 02 December 2021).

Cui, L., Xie, G., Qu, Y., Gao, L., Yang, Y., 2018. Security and privacy in smart cities: Challenges and opportunities. IEEE Access 6, 46134–46145.

de Vicente Mohino, J., Bermejo Higuera, J., Bermejo Higuera, J.R., Sicilia Montalvo, J.A., 2019. The application of a new secure software development life cycle (S-SDLC) with agile methodologies. Electronics 8 (11), 1218.

Dempsey, K.L., Johnson, L.A., Scholl, M.A., Stine, K.M., Jones, A.C., Orebaugh, A., Chawla, N.S., Johnston, R., et al., 2011. Information security continuous monitoring (ISCM) for federal information systems and organizations.

Faizi, S., Rahman, S., 2019. Choosing the best-fit lifecycle framework while addressing functionality and security issues. In: CATA. pp. 107–116.

Ghelani, D., Hua, T.K., Koduru, S.K.R., 2022. A model-driven approach for online banking application using angularjs framework. Am. J. Inf. Sci. Technol. 6 (3), 52–63.

Haque, A.B., Bhushan, B., Dhiman, G., 2022. Conceptualizing smart city applications: Requirements, architecture, security issues, and emerging trends. Expert Syst. 39 (5), e12753.

Ijaz, S., Shah, M.A., Khan, A., Ahmed, M., 2016. Smart cities: A survey on security concerns. Int. J. Adv. Comput. Sci. Appl. 7 (2), 612–625.

2021a. ISO/IEC 27001:2013 information technology - security techniques - information security management systems – requirements. https://www.iso.org/standard/54534.html. (Accessed 015 December 2021).

2021b. ISO/IEC 27035-2:2016 information technology - security techniques - information security incident management - Part 2: Guidelines to plan and prepare for incident response. https://www.iso.org/standard/62071.html. (Accessed 18 December 2021).

2021c. ISO 31000:2018 risk management – guidelines. https://www.iso.org/standard/65694.html. (Accessed 17 December 2021).

Jammeh, B., 2020. DevSecOps: Security expertise a key to automated testing in ci/cd pipeline.

Kaluarachchi, Y., 2022. Implementing data-driven smart city applications for future cities. Smart Cities 5 (2), 455–474.

Kara, M., 2012. Review on common criteria as a secure software development model. Int. J. Comput. Sci. Inf. Technol. 4 (2), 83.

Kaushal, R.K., Bhardwaj, R., Kumar, N., Aljohani, A.A., Gupta, S.K., Singh, P., Purohit, N., 2022. Using mobile computing to provide a smart and secure Internet of Things (IoT) framework for medical applications. Wirel. Commun. Mob. Comput. 2022, 1–13.

Khan, R.A., Khan, S.U., Khan, H.U., Ilyas, M., 2022. Systematic literature review on security risks and its practices in secure software development. Ieee Access 10, 5456–5481.

Lingham, A.D., Kin, N.T.K., Jing, C.W., Loong, C.H., et al., 2020. Implementation of security features in software development phases. arXiv preprint arXiv:2012.13108.

Mahendra, N., Khan, S.A., 2016. A categorized review on software security testing. Int. J. Comput. Appl. 154 (1), 21–25.

2021. Microsoft security development lifecycle. https://www.microsoft.com/en-us/securityengineering/sdl/. (Accessed 03 December 2021).

2021. NIST risk management framework. https://csrc.nist.gov/projects/risk-management/about-rmf. (Accessed 18 December 2021).

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.

Ogbodo, E.U., Abu-Mahfouz, A.M., Kurien, A.M., 2022. A survey on 5G and LPWAN-IoT for improved smart cities and remote area applications: From the aspect of architecture and security. Sensors 22 (16), 6313.

Radack, S., et al., 2008. Guide to Information Security Testing and Assessment. Tech. Rep., Technical report, National Institute of Standards and Technology.

Rana, S.K., Rana, A.K., Rana, S.K., Sharma, V., Lilhore, U.K., Khalaf, O.I., Galletta, A., 2023. Decentralized model to protect digital evidence via smart contracts using layer 2 polygon blockchain. IEEE Access.

Rao, P.M., Deebak, B., 2022. Security and privacy issues in smart cities/industries: technologies, applications, and challenges. J. Ambient Intell. Humaniz. Comput. 1–37.

Rindell, K., Hyrynsalmi, S., Leppänen, V., 2018. Aligning security objectives with agile software development. In: Proceedings of the 19th International Conference on Agile Software Development: Companion. pp. 1–9.

Sharma, A., Misra, P.K., 2017. Aspects of enhancing security in software development life cycle. Adv. Comput. Sci. Technol. 10 (2), 203–210.

2021. Software assurance maturity model. https://owaspsamm.org/model/. (Accessed 01 December 2021).

Taherdoost, H., 2022. A critical review of blockchain acceptance models—blockchain technology adoption frameworks and applications. Computers 11 (2), 24.

Taheri, R., Ahmed, H., Arslan, E., 2023. Deep learning for the security of software-defined networks: a review. Cluster Comput. 26 (5), 3089–3112.

Tung, Y.-H., Lo, S.-C., Shih, J.-F., Lin, H.-F., 2016. An integrated security testing framework for secure software development life cycle. In: 2016 18th Asia-Pacific Network Operations and Management Symposium. APNOMS, IEEE, pp. 1–4.

Ullah, F., Qayyum, S., Thaheem, M.J., Al-Turjman, F., Sepasgozar, S.M., 2021. Risk management in sustainable smart cities governance: A TOE framework. Technol. Forecast. Soc. Change 167, 120743.

Valdés-Rodríguez, Y., Hochstetter-Diez, J., Díaz-Arancibia, J., Cadena-Martínez, R., 2023. Towards the integration of security practices in agile software development: A systematic mapping review. Appl. Sci. 13 (7), 4578.

Yusuf Mohamed Ali Muthanna Is a Ph.D. student in Computer Science, college of It, University of Bahrain.

Wael Elmedany holds a PhD degree in Electrical Engineering, Manchester University, UK, 1999; MSc degree in computer communications, Menoufia University, Egypt, 1991; BSc degree in Electronic Engineering, Menoufia University, Egypt 1987. He is the founding and managing editor of International Journal of Computing and Digital Systems (IJCDS). He is the founder and Chair of MobiApps, DPNoC, and WoTBD workshops series. El-Medany is a senior IEEE member, member of editorial boards and TPC member of many international journals and conferences, and acts as chairperson in many conferences. His research interests in ASIC design, FPGA, embedded systems, remote monitoring systems, and reconfigurable computing.

Mustafa M. Hammad, Ph.D. Associate Professor Department of Computer Science, Faculty of IT Mutah University Al-Karak, Jordan

Riadh Ksantini received the M.Sc. and Ph.D. degrees in Computer Science from the Université de Sherbrooke, Sherbrooke, QC, Canada, in 2003 and 2007, respectively. From 2001 to 2007, he was a graduate research associate with Bell Canada Laboratories and the research center MOIVRE (MOdElisationen Imagerie, Vision et REseaux de neurones). Presently, he is Associate Professor at the Department of Computer Science, College of IT, University of Bahrain, Adjunct Associate Professor at the School of Computer Science, within the Faculty of Science of the University of Windsor, Windsor, Ontario, Canada, and Adjunct Professor at the Department of Computer Science,

Université du Québec à Montréal (UQAM). Prior to that, he was Postdoctoral Research Associate in the Ecole de Technologie Superieure, University du Quebec, Montreal, Canada. He has also served as Visiting Fellow Research Scientist at the Canadian Space agency, and Postdoctoral Research Associate in the School of Computer Science, within the Faculty of Science of the University of Windsor. In 2008, he was awarded a fellowship (of excellence) for postdoctoral research from the granting agency - Fonds quebecois de la recherche sur la nature et les technologies - (FQRNT). His PhD was evaluated and ranked third Ph.D. in Quebec for 2007 by the committee of Information Technology and Communications. His research interests include Artificial Intelligence, Machine/Deep Learning, Pattern Recognition and Computer Vision. His research work on Artificial Intelligence and Data Science has always involved collaboration between academia and industry. More than 70 Articles stemming from his research work have been published in several prestigious journals and conferences.

Dr Saeed Sharif is the Leader of Intelligent Technologies Research Group, Course Leader for Msc Computer Science and Course Leader for Msc Computer Science with industrial Placement at the school of architecture, computing, and engineering. He has a PhD in artificial intelligence from Brunel University London. His research interests and expertise include Artificial Intelligence, Innovative Tele-Health, Medical Technology, Digital Health Care and Medical Assistive Technology, Medical Image Analysis and Visualization, Intelligent Diagnosis Systems, Smart Biomedical Image, and Bio Signal Acquisition (MRI, PET, EEG), Nanotechnology, Medical Biotechnology, Security, and Big Data Analysis. He is working closely with clinicians and policy makers nationally and internationally to improve the clinical settings and the healthcare systems. He has led the research development of different research projects associated with many industries and NHS trusts. He has developed his research with the goal of improving the effectiveness, efficiency of medical and health care systems.

Dr Saeed has published a significant number of research articles in highly reputable journals, international conferences, and book chapters, such as Elsevier Applied Soft Computing, Computer Methods and Programs in Biomedicine etc. He is a member of the technical committee of several international conferences such as IEEE International Conference on Computer Systems and Applications and IEEE International Conference on Computer and Information Technology. He has participated in many national and international conferences e.g., ICIP. He has served as a reviewer for many journals e.g., IET IPJ, Elsevier CBMJ. He is Guest Editor at international journals. He received many academic and research awards. He is a Fellow of The UK Higher Education Academy, and member of the British Computer Society.