SECURETI: ADVANCED SDLC AND PROJECT MANAGEMENT TOOL FOR TI (PHILIPPINES)

Amelie Anne Gutierrez

School of Computing, Professional Science Master's in Cybersecurity,
Holy Angel University, Angeles City, Pampanga, Philippines

ABSTRACT

There are essential security considerations in the systems used by semiconductor companies like TI. Along with other semiconductor companies, TI has recognized that IT security is highly crucial during web application developers' system development life cycle (SDLC). The challenges faced by TI web developers were consolidated via questionnaires starting with how risk management and secure coding can be reinforced in SDLC; and how to achieve IT Security, PM and SDLC initiatives by developing a prototype which was evaluated considering the aforementioned goals. This study aimed to practice NIST strategies by integrating risk management checkpoints in the SDLC; enforce secure coding using static code analysis tool by developing a prototype application mapped with IT Security goals, project management and SDLC initiatives and evaluation of the impact of the proposed solution. This paper discussed how SecureTI was able to satisfy IT Security requirements in the SDLC and PM phases.

KEYWORDS

Security, SDLC, project management, NIST, static analysis.

1. INTRODUCTION

Semiconductors are almost everywhere. Widely known as Integrated Circuits (ICs), all computers, different means of communication, and all modes of transportation which get us from place to place, even the electronic devices that we are privileged to have for entertainment purposes like listening to a variety of music, watching a vast selection of movies, playing games; the medical equipment which are used in medical facilities, some military systems, and a lot more products of technology are all within our reach because of ICs. The convenience of technology that we are privileged to use every day would not be possible without semiconductors. Given this fact, semiconductor companies still continuously revolutionize new technologies that will help transform society for the better. These include the Internet of Things (IoT), virtual reality, energy efficient sensing, robotics, and Artificial Intelligence (AI) [1].

According to a Globe Newswire report that was presented by Research and Markets, one of the largest market research stores conducting analysis of industry sectors, Texas Instruments (TI) Inc. was included in the 2019 top ten semiconductor companies globally [2]. The products manufactured by TI focuses on analog and embedded processing, plus digital light processing (DLP) and education technology [3].

TI is currently situated in more than thirty (30) countries globally. There are two TI sites located in the Philippines, TI Baguio and TI Clark with roughly ten (10) web application developers supporting two local sites [3]. "A web-based application is a computer program consisting of

clients and web servers which runs on a web browser application"[4]. Web application developers are responsible for the system development of web application programs.

Application developers are critical members of a project management team. They are responsible for developing and translating the business need in order to resolve system related problems and innovations in the organization. The most common tasks of application developers include system/application planning and design, development and testing, deployment and support, troubleshooting, end user support and training, monitoring/implementing updates and security. They can also act as project managers who can exercise collaboration and communication within the organization [5].

The rapid pace of innovation most evident in semiconductor companies like TI undeniably requires radical security considerations in the systems/applications supported by their web application developers. TI has acknowledged that security and cyber security have become top concerns in this complex digital era due to different cyber attacks and data breach attempts. TI, along with other semiconductor companies, has recognized that IT security is highly essential during the system development life cycle (SDLC). "Designers and developers must take security seriously, conducting a thorough risk evaluation and selecting appropriate measures so that their application protects user privacy and defends consumers against fraudulent actions." [3]

This study aims to propose an advanced SDLC and project management tool to TI (Philippines) web application developers that is integrated with IT security considerations laying emphasis on risk management and secure coding by making best use of a static analysis tool. Risk Management is the implementation of strategies to manage and address a risk. A risk is defined as any potential harm which may arise from a current process or future event [6]. Secure coding helps in preventing cyber attacks because it eliminates the vulnerabilities in the source code starting from writing the code itself. Secure coding helps keep the code efficient and optimized, standard and with considerations to security [7].

The proposed web application serves as a project management tool which incorporates IT security controls in each of the SDLC phases using NIST Special Publication (SP) 800 37 Revision 2, Risk Management Framework for Information Systems and Organizations and NIST Special Publication (SP) 800 160 Volume 1, Systems Security Engineering as guidelines for the integration of risk management and security tasks in the system development cycle.

In general, the web application developers of TI (Philippines) are faced with the current SDLC predicaments based on the pre-development survey conducted.

- How will risk management be integrated in the SDLC?
- How to re-assess the importance of secure coding in web application development?
- Is there an existing standard secure coding tool being utilized by the TI (Philippines) web application developers in the SDLC? If none, how will this be enforced and what tool will be used?
- What strategy can be done in order to align IT security, project management, and SDLC initiatives? Can this be achieved using a web application?
- What is the impact of using a single web application tool to achieve IT security goals, project management initiatives, and an effective SDLC model?

2. CONCEPTUAL FRAMEWORK AND MODEL

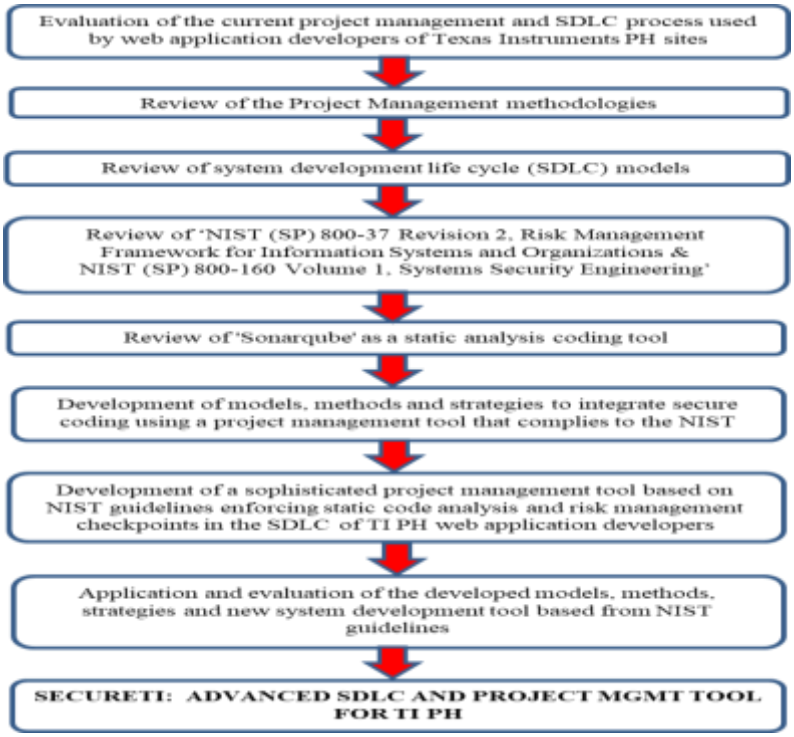

Figure 1. Conceptual Framework

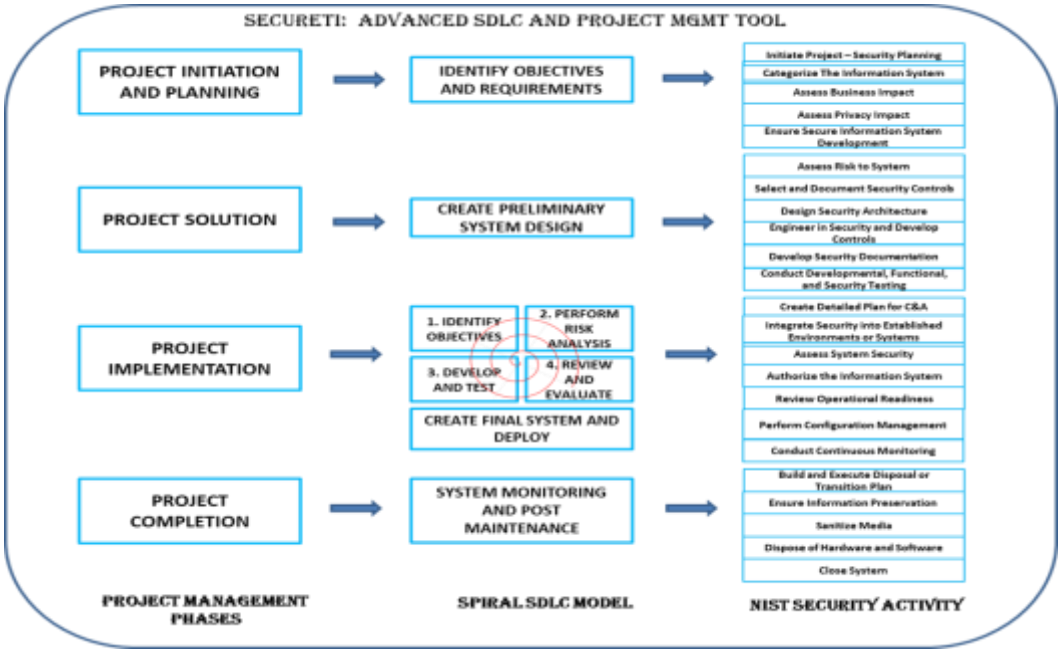

Figure 2. Conceptual Model

3. RELATED LITERATURE

This section reviews the scholarly publications and previous studies that are related to the overview of Texas Instruments as a semiconductor company; review of the project management phases and determining the appropriate project management methodology to be utilized in this study; comparative analysis of the different system development life cycle (SDLC) models and how this can be aligned with project management goals; importance of secure coding in application development by utilizing static analysis tools and overview of the NIST.SP.800 160v1 Systems Security Engineering and NIST.SP.800 37r2 Risk Management Framework for Information Systems and Organizations guidelines. These related publications serve as the foundations for SecureTI, an advanced SDLC and project management tool for TI PH web application developers.

When it comes to the security and efficiency of a web application, project management is a very crucial aspect in building the IT security foundations during system development. Hitachi Systems Security highlighted the five benefits of project management in Cybersecurity which include streamlined project execution with sufficient budget and acceptable scheduling; strategic alignment since the project is aligned with the organization goals; optimized resource allocation to make sure that critical projects are assigned with the right resources; continuous improvement by having proper documentation about the advantages and disadvantages of a project considering the lessons learned after its implementation; and identification of potential risks of a project while communicating it to the stakeholders with the feasibility study of the project given the risks [8]. Abhishek Gupta from the Computer Engineering Department of California State University also highlights the importance of project management for business firms when it comes to resource management, project tracking and monitoring, project stakeholders and project management team conversations. Figure 3 shows a generic project life cycle [9] which is currently being followed by TI (Philippines) web application developers.

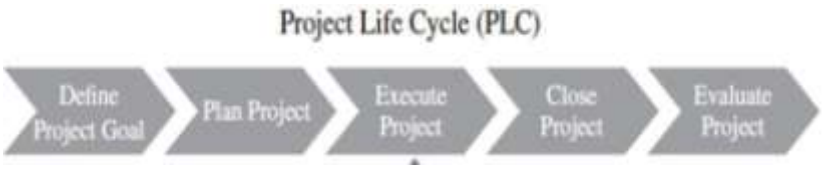

Figure 3. A Generic Project Life Cycle

In order to determine which project management methodology must be incorporated in the proposed web application, a study entitled 'Agile Project Management Tools: A Brief Comparative View' by Deniz Ozkan and Alok Mishra from Turkey was reviewed. According to the study, using agile methodology in software development lowers the cost, leads to better quality, productivity and higher satisfaction [10]. One of the principles of Agile manifesto mentioned in the study, "Responding to Change Over Following a Plan", would be used as a norm in this study to develop an advanced version of existing project management tools. A project management tool aims to help employees work on the same project and database; get instantaneous feedback on project updates; and communicate useful information to stakeholders on project status[11]. Table 1 shows the summary of the comparative analysis of different Agile Project Management tools [10]

Table 1. Comparison of Agile Project Management Tools

Name	Platform Based	Web Based	Online	Cloud Based	Burn Down Chart	Agile Boards	Milestones	Resource Management	Time Tracking	Bug Tracking	Tasks	Integration	Reports	Documents	Version Control	Workspaces	User role	Pricing	Free Version
Jira		✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓		✓	✓	
Active Collab			✓	✓					✓		✓		✓	✓			✓	✓	
Agilo for Scrum	✓					✓					✓		✓	✓			✓	✓	
SpiraTeam by Inflectra	✓		✓		✓	✓	✓	✓	✓	✓	✓		✓	✓		✓	✓	✓	
Pivotal Tracker			✓			✓			✓				✓	✓		✓	✓	✓	✓
VSTS			✓		✓	✓	✓		✓		✓		✓	✓	✓		✓	✓	✓
Icescrum	✓		✓	✓	✓	✓				✓		✓	✓		✓		✓	✓	
SprintGrounds			✓		✓		✓			✓	✓	✓	✓		✓		✓	✓	✓
VersionOne			✓			✓	✓		✓	✓	✓	✓	✓	✓			✓	✓	✓
Taiga	✓	✓				✓	✓				✓	✓	✓	✓	✓		✓		✓
Agilelan			✓	✓		✓				✓	✓		✓	✓				✓	
Wrike	✓		✓	✓	✓	✓	✓	✓	✓		✓			✓			✓	✓	✓
Trello			✓	✓		✓	✓		✓		✓		✓	✓				✓	✓
Axosoft			✓	✓	✓	✓	✓	✓				✓						✓	
Planbox			✓	✓			✓	✓	✓		✓							✓	✓
Asana	✓		✓			✓	✓	✓			✓		✓					✓	✓

Based from the results of this previous research, the top five (5) most common characteristics of Agile Project Management tools are:

- Accessed online
- Pricing – there is cost involvement
- Use agile boards/task boards which show visual progress of project tasks
- Have task definition and monitoring
- With reporting functionality

Determining the most common characteristics of agile project management tools would be very important in this paper to be able to deliver a project management tool that is built from the comparative analysis of different existing agile project management tools and integrating these core features into an advanced SDLC and project management tool for TI (Philippines) web application developers which would not incur any cost for the organization.

Another significant element in this study, the system development life cycle (SDLC) is a conceptual model of project management which focuses on planning which identify and organize all activities required in the development of a system; analysis which lists down what is expected from the system; design, where the architecture of the system is being created and examined; implementation which is the pilot execution of the developed software with user testing; operation is the actual implementation in the production environment; maintenance involves supporting the deployed system in production and making sure that is always available; and lastly, the elimination/discard of the system which is disposition[12]. Figure 4 shows the different phases of systems development life cycle [9]

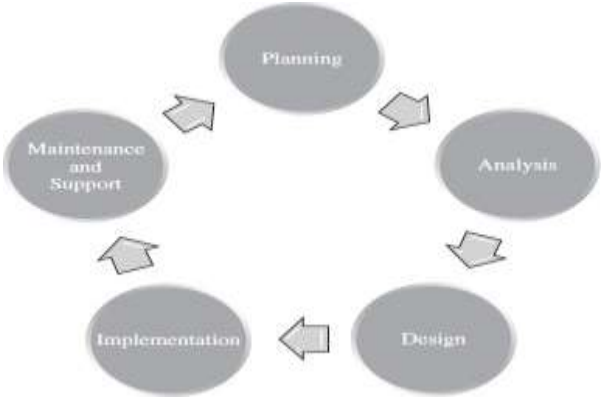

Figure 4. The Systems Development Life Cycle

Brian Evans, CISSP, CISM, CISA, CGEIT, the Senior Managing Consultant of Security Services expressed the importance of SDLC (Systems Development Life Cycle) to ensure that security controls and requirements are implemented in a system or application. According to him, there should be defined and enforced checkpoints that integrate IT security checks in every project phase [13]. SDLC deals with people and their processes, the technology as a whole which include infrastructure, software, and change management. In his article entitled 'The System Development Life Cycle: A Phased Approach to Application Security', he outlined the security reviews in each of the SDLC phases:

Project Initiation phase is where the project initiator prepares a formal request to the application team presenting the project objectives, stakeholders, criticality or business impact, and planned timeline.

Functional Specifications come from the user's perspective which includes data flow diagrams, data definitions, screen definitions, input data resources, report definitions, control and security requirements, and system interface requirements. Backup, restart and recovery, contingency requirements, hardware requirements, service level requirements capacity requirements, conversion requirements are also considered in this phase.

Security teams must participate in the post implementation review to confirm that the security features are effective. The security decisions are documented and the application is finalized.

Having achieved some understanding of the SDLC and project life cycles, their alignment was scrutinized further by Iryna Meschankina's article entitled 'The Software Development Life Cycle: Phases and Methodologies – Aligning SDLC with Project Management'. According to her, good project managers must be able to know how to combine managerial practices with SDLC. There must be a platform wherein project managers and developers can be able to communicate the status of a project to its stakeholders and deliver project milestone for each SDLC step. For example, Initiation – the first phase of project management is where the initial project scope and schedule are defined. Requirements gathering in SDLC are also a part of project initiation. The next stage is Planning, which deals with more intricate project benefit approximation. Execution integrates several SDLC phases at the same time which include Implementation (Coding), Testing, and Deployment. The Closeout is a phase to monitor the system performance and transition for future use. Figure 5 describes a mapping of the system development life cycle and project management phases [14].

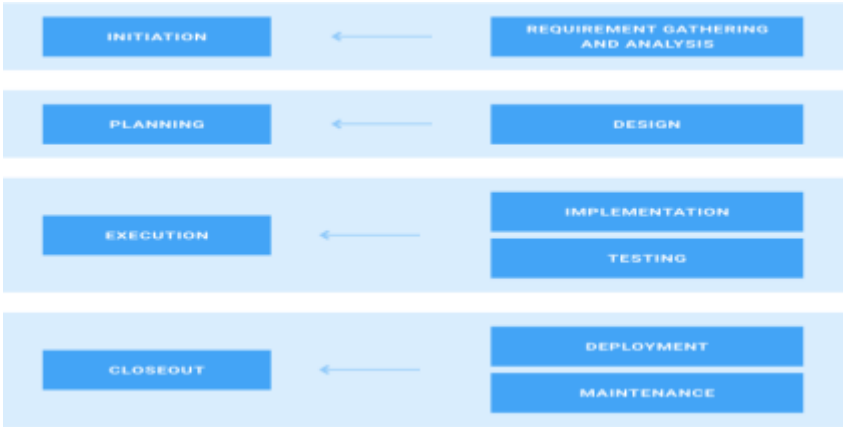

Figure 5. Alignment of the SDLC and Project Life Cycle Phases

To effectively apply the most efficient SDLC methodology in this study, the different properties of SDLC methods has been deliberated. In a journal entitled "Software Development Methods – Properties and Advances", the software development models are examined.

After the analysis of the different SDLC Model properties, it is concluded that only one of the SDLC models gives emphasis to risk assumptions and assessments in the SDLC phases, this is the Spiral SDLC model. The Spiral SDLC model combines waterfall and prototyping models. Risk management is one of primary objectives of this paper and the related studies show that spiral model is the best option for SecureTI, an advanced SDLC and project management tool for TI (Philippines) web application developers.

Jeff Williams, founder and major contributor in the Open Web Application Security Project (OWASP) mentioned in the OWASP Code Review guide: "The code is your only advantage over the hackers. Don't give up this advantage and rely only on external penetration testing. Use the – code". NIST Cybersecurity White Paper draft entitled 'Mitigating the Risk of Software Vulnerabilities by Adopting an SSDF', also stressed that secure coding tools reduce the number of risks in the application. Cost is also saved by eliminating vulnerabilities during the source code creation. According to a journal entitled "Security Risks in the Software Development Lifecycle" by Mamdouh Alenezi and Sadiq Almuairfi, risks and code errors should be eliminated during the development cycle and security risk assessment must be integrated into each phase of SDLC. The journal stressed the importance of ensuring code and software quality with a high level of security by implementing the best practices in risk management all throughout the development process.

One of the best practices in the study is the review/analysis of human readable code in order to pinpoint vulnerabilities and verify security requirements compliance. Mohammad Nadeem, Product Development Manager of Oracle, wrote a review highlighting 'Sonarqube' as a static code analysis tool. Nadeem discussed about how the quality of code can be evaluated. Software high quality characteristics are described by the following:

- standard coding
- bad coding practices were not used
- inexistence of any potential bugs, security risks and performance issues
- no duplicate code
- simple code logic
- good documentation and comments

- unit tests in code
- good design and architecture principles

SonarQube is a combination of static and dynamic analysis tool which collects and analyses source code, measuring quality and providing reports for applications. Styling details and critical design errors are evaluated by SonarQube. SonarQube allows developers to access and monitor code analysis by providing data which consider styling errors, code defects to design inefficiencies, code duplication, potential bugs, lack of test cases, and others. It aids the developer, project managers and also higher managerial levels to see the relevant details based on their perspective. The Sonarqube dashboard addresses bugs, test cases, and duplications, documentation, and also architecture.

NIST SP 800 37, Risk Management Framework for Information Systems and Organizations A System Life Cycle Approach for Security and Privacy updates is aligned with the NIST Cybersecurity Framework. It has integration of privacy risk management processes, supply chain risk management –processes and system life cycle security engineering processes. Revision 2 contains RMF tasks for information system owners to help in conducting system level risk management. –It characterizes the Risk Management Framework (RMF) with guidelines in applying risk management to information systems and organizations. –This publication highlights a well defined process for managing security and privacy risks. These risks may include information security – categorization, authorizations and continuous –monitoring, control selection, assessment, . The RMF also highlights almost real time risk management. There is implementation of continuous monitoring process, which is used as reference by leaders and – executives of an organization in making efficient and cost effective decisions about the system with significance to security and privacy in the SDLC. Execution of the RMF tasks reduces risks in the processes and management processes level. It also promotes accountability for the control implementation within an organization.

On the other hand, NIST SP 800 160 Volume 1 Systems Security Engineering Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems highlights engineering driven initiatives and recommends actions to develop trustworthy and supportable systems and all of its components. The International Organization for Standardization (ISO) systems and software engineering international standards, Institute of Electrical and Electronics Engineers (IEEE), International Electrotechnical Commission (IEC) are used as references by this publication. NIST SP 800 160 aims to resolve issues encountered during the SDLC.

Another research presented in the International Journal of Innovative Research in Computer and Communication Engineering by conducting a full text search using keywords with the keywords “software”, “security”, “SDLC” is the study entitled “Incorporating Security into SDLC Phases Using Security Analysis”. Kulkarni and Joshi incorporated security concerns when creating the software requirements like authentication, session, logging, and auditing requirements. They also studied integrity, availability, and application controls with system configuration, compliance, and security requirements. In software design, the proponents of this related study added security into the system design by developing a threat model and performing a security architecture review.

The related literature mentioned in this study particularly by Brian Evans presented different methods on how IT Security can be integrated in each phase of SDLC. Similarly, a detailed analysis of how to incorporate Security into SDLC phases was conducted by the researchers from the International Journal of Innovative Research in Computer and Communication Engineering. Another article by Meschankina tried to map project life cycle phases with SDLC phases. The main objective of this study is to incorporate IT Security checks not only in the SDLC alone but

also in the project life cycle by using a globally recognized standard from the NIST SP 800 37 Risk Management Framework and NIST SP 800 160 Systems Security Engineering. Therefore, the related studies only scrutinized the basic concepts and analysis on what project management methodology to apply; the most appropriate SDLC model for risk management; and the standard which can be used as reference to ensure that IT Security controls during system development and project management are in place. Consolidating these scholar publications eventually aided the proponent of this study to apply Agile project management methodology, Spiral SDLC model and NIST risk management and systems engineering guidelines. Standard risk management tasks and security engineering steps can be completed and tracked in a single web application tool to fortify not only the IT Security piece but also the system development and project management of an application project. A new web application which could identify and address vulnerabilities and risks during system development and support project management deliverables referenced from the NIST guidelines would suffice in helping the web application developers of TI (Philippines) to leverage in terms of IT Security compliance and deliver quality manufacturing web applications.

4. DEVELOPMENT OF SECURETI

To develop a prototype of SecureTI which will enforce IT security checks in every phase of SDLC by including risk management and secure system engineering methods referenced from NIST (SP) 800 37 and NIST (SP) 800 160 Volume 1, the Prototyping Model was followed. Information protection and security was practiced in all of the phases of the prototyping model in order to protect both the system and the information which would be processed by it. Prototyping starts with the requirements gathering, followed by quick design of the system, actual building/development of the prototype, user evaluation after development and updating the system for necessary revisions based from the user evaluation results.

After the completion of SecureTI prototype, simulations were performed starting with the secured system accessibility via TI Virtual Private Network (VPN) and TI authentication. The system features exhibited by SecureTI involved real-time project management phase status update and a single page project management tool (PMT) interface with the use of agile boards. Task definition of each system development life cycle (SLDC) phases were also evident. In addition, creating and editing new projects with the NIST referenced security checkpoints for each SDLC phase was also highlighted during the testing.

Project deletion was also part of the simulated scenarios. Another important feature was the secure coding initiative by means of static code analysis check that was triggered by SecureTI PMT and SDLC tool. The scanning was activated with the use of a third-party static analysis tool that was invoked in SecureTI to ensure that the static code analysis step will not be missed in the SDLC phases. User role restriction and management functions were also validated.

A TI local development database was used for SecureTI. The schema of the database, including its tables and views were omitted in this paper to avoid compliance issues with the data privacy laws. TI proprietary libraries and configurations were also excluded in this paper to avoid violation of the data privacy act.

4.1. SecureTI Framework

This section presents the detailed design and development of SecureTI: Advanced SDLC and PM Tool for TI (Philippines).

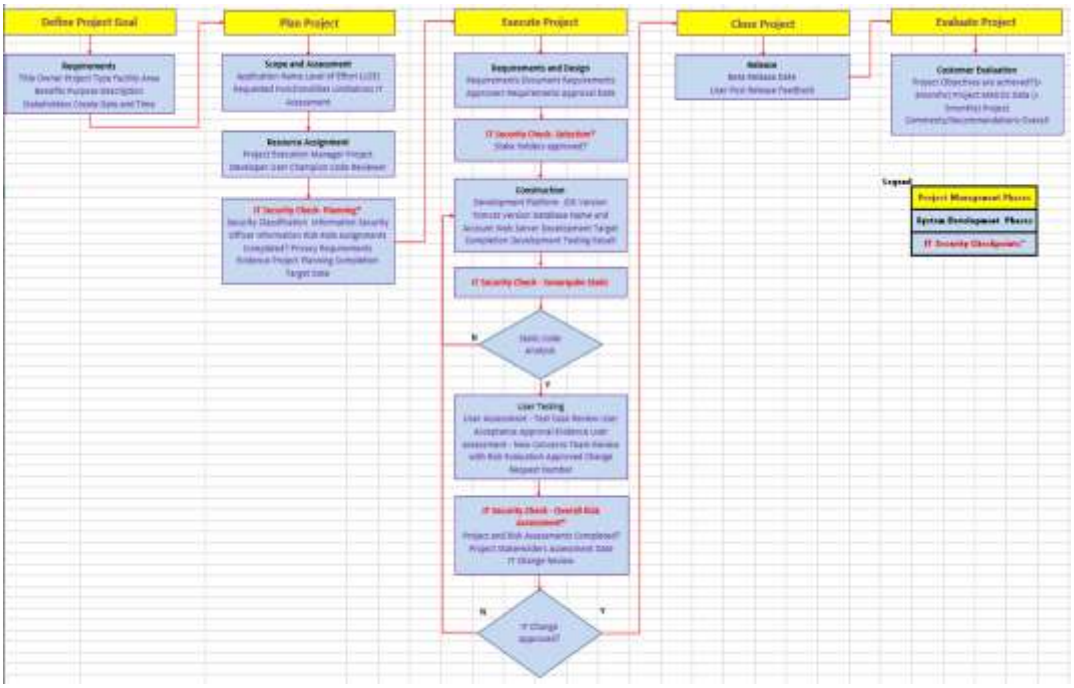

Figure 6. SecureTI System Flow

4.1.1. Project Management Phases

Project Management life cycle starts with initiation. In this phase, the project's value and feasibility are determined. Upon approval of the project initiation, a well written project plan gives guidance for the project manager to obtain resources and acquire financial support and laying out talents who can help accomplish the project. Project planning gives direction to a project, assess the risks, communicate benefits to stakeholders and prepares an action plan for roadblocks that the project may encounter. Next phase is execution which is all about deliverables which can satisfy the system user. Execution relies heavily on planning. Lastly, project closure is a vital step for a project to be evaluated and completing all the requirements to move on and close the project. Project evaluation and monitoring is relevant in measuring the progress of a project. Monitoring is done to avoid scope creep and calculate estimated performance of the system proposed.

4.1.2. System Development Life Cycle

Requirement analysis is the initial and most fundamental stage in SDLC. This phase mostly involves planning for the quality assurance requirements and identifying the associated risks with the proposed project. Defining requirements is usually done by providing SRS (Software Requirement Specification) document which consists of the system design and overall requirements. Based on the requirements specified in SRS, DDS - Design Document Specification is established which defines all the architectural modules and process flow of the proposed system. During the development state, coding happens and guidelines are also being followed. Testing activities happen after development. This is the stage where defects in the code are identified, fixed and retested until the system reaches quality standards. Lastly, deployment is the last part of SDLC where system is formally introduced and used by the target

users. Based on feedback, the system may be permanently released or generated suggested enhancements together with the system maintenance.

4.1.3. NIST Special Publication 800-37 Revision 2

Risk Management Framework implements security and privacy requirements. One of the most important goals of the Risk Management Framework For Information Systems And Organizations, A System Life Cycle Approach for Security and Privacy is to incorporate security and privacy into the SDLC. Security check items incorporated in the SDLC were taken from this reference: Objective/Purpose which can help organizations can simplify RMF execution, employ innovative approaches for managing risk, and increase the level of automation when carrying out specific tasks; Task P-9 involves identifying the key stakeholders who provides the goals and objectives of the project and prepare the organization to execute the RMF. Communication among stakeholders is important during every step in the RMF and throughout the SDLC to ensure that security and privacy requirements are satisfied. Next is asset identification which involve business impact analysis and identification of different resources with interaction to the system. Task P-11 determines the documented authorization limits of the system. Task P-15 focuses on requirements definition which require documented security and privacy requirements. Concept 3.2 specifically focuses on categorizing the summary of tasks for RMF. Task C-1 requires documentation of the characteristics of the system. Task C-3 talks about the approval of the project scope and requirements provided. Task S-4 summarizes the functionalities of the system which include the planned inputs and the expected outputs and behaviour of the system. Task A-3 are the activities related to developmental testing and evaluation, regression testing and application scanning. This is the specific SDLC step where static code analysis can be useful in detecting deficiencies earlier in the code. Task A-6 highlights the milestones based on the system assessments and target completion schedules. Task M-1 monitors the system and its environment for any changes which can impact the security and performance of the system. Task M-5 discusses the security, privacy and monitoring activities headed by the system privacy officer or the system security officer. OMB A-130 is the adequate security resulting from unauthorized access use and modification of system information. As for the key participants in the risk management process, the authorizing officials are also determined and usually consist of system owners, chief information officers, and other stakeholders. The assignment of the chief information officer is also part of the SDLC and security check item who develop and maintain security policies, procedures and other system actions. Another concept is the system user that is authorized to access system information and perform system acceptance testing. In developmental testing, users usually validate the functionalities of a system and the effectiveness of the implemented controls. Lastly, the information risk assessment is being examined; the level of trust and confidence to accept lesser risks and assurance that overall performance will not contribute to greater risk.

4.1.4. NIST Special Publication 800-160 Volume 1

The Systems Security Engineering Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems present engineering-based solutions, perspectives and actions necessary to develop more stable and effective systems. SN-2.2 is similar to TaskP- 9 of NIST 800 37 which involve identification of project stakeholders. AQ-1.2 achieves the outcome of stakeholder needs and requirements definition process. In item 3.4.2 defines stakeholder needs and requirements together with the capabilities needed by users. PM-1.3 defines the security, authorization and accountabilities of a project. AR-5.1 involves IT Assessment that is oriented to provide technical suitability relative to the system requirements and associated security-related constraints. Trade-off is another concept from ISO/IEC/IEEE 15288 which drive decision-making actions on the basis of net benefit to the stakeholders. DE-3.2 enumerates the functionalities required to be satisfied all throughout the project. It also contains the limitations outside the capabilities of the system. Concept F.1.7 Reduced Complexity highlights the role of system developers to understand the correctness and completeness of

security functions. F.1.12 Hierarchical Protection talks about a trustworthy user who utilizes the system. AQ3.2 involves identification and evaluation of necessary changes for a project. The most important concept in NIST 800-160 is the purpose of the systems security engineering. Security verification identifies and provides evidence of cod anomalies, errors and flaws that are assessed through system analysis process. Static analysis is a crucial step in this SDLC step. IF-2.2 provides the changes to the infrastructure resources. Concept 3.2.3 Portfolio Management Process talks about closed projects which satisfy all security criteria and requirements definition. IF-2.1 discusses the method of evaluation and success criteria based on the defined requirements. PA-2.9 analyses security measurement results and provide recommendations to the moving forward steps for the system.

4.2. SecureTI Requirements

Since SecureTI uses a TI local database, credentials are no longer provided in this paper, but the conceptual entity relationship diagram is shown in Figure 7

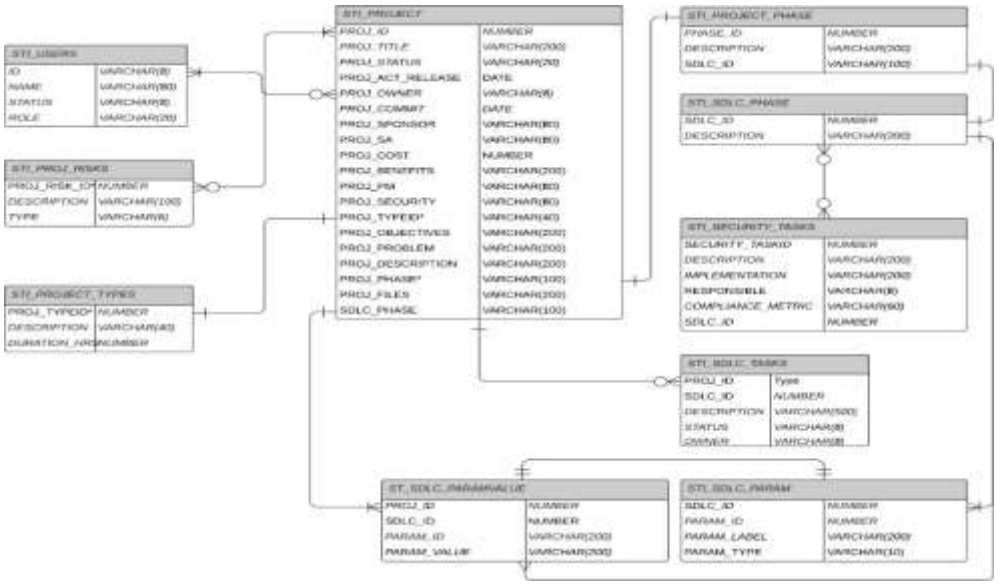

Figure 7. Conceptual Entity Relationship Diagram (Redacted)

4.2.1. Server Requirements

Apache Tomcat 6 above is required to run the system

4.2.2. Java Version Requirement

Java Runtime Environment is JDK 1.6 above

4.2.3. Browser Compatibility

Local server SecureTI and Sonarqube Windows CLI scanner is compatible with Internet Explorer.

4.3. SecureTI System

Figure 8. Virtual Private Network Login(Redacted)

Figure 9. Authentication (Redacted)

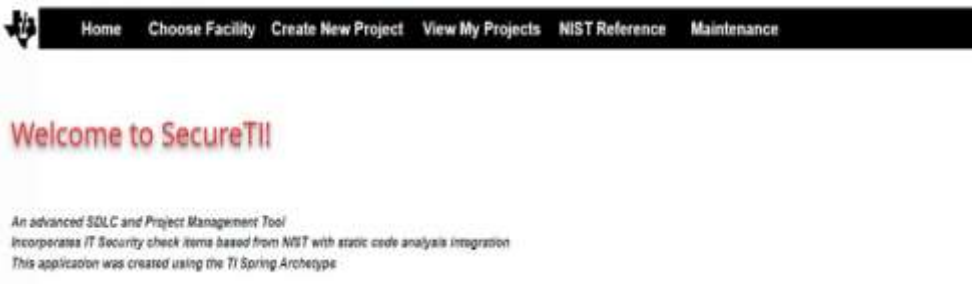

Figure 10. Home Page/ Welcome Page

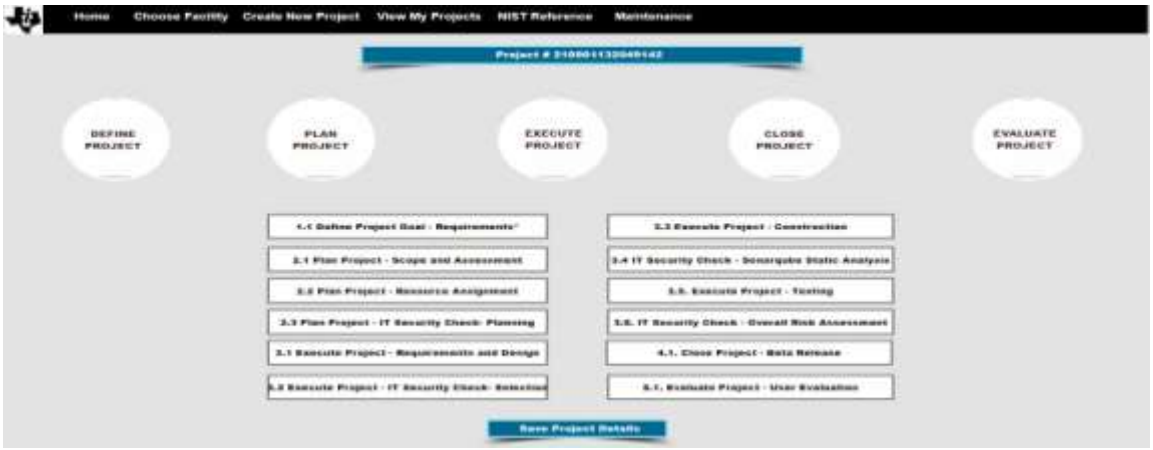

Figure 11. Create New Project Page

Figure 12. Sample SDLC steps with IT Security Checks

Figure 13. SecureTI Submitted Project Page

Figure 14. SecureTI Edit

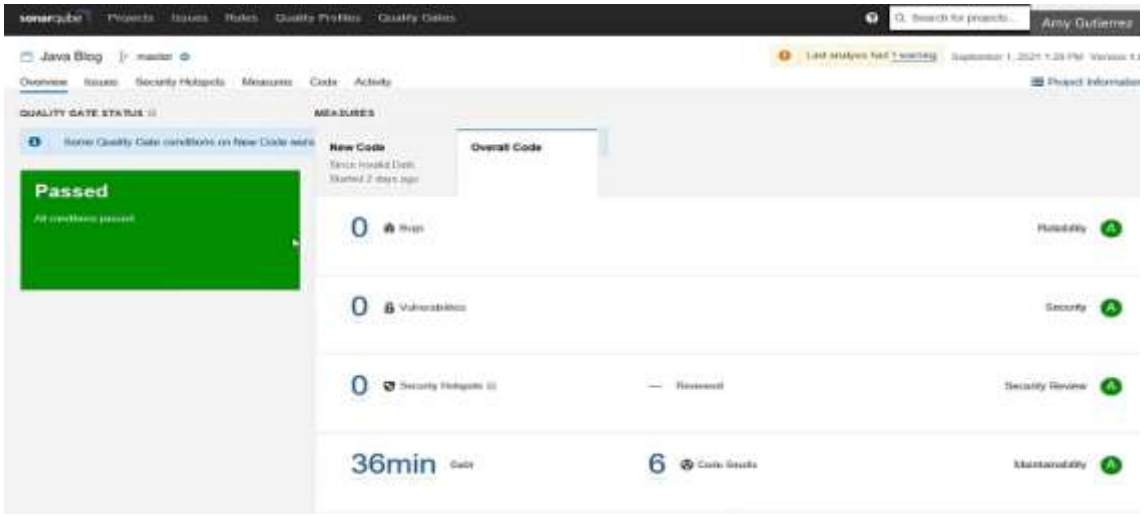

Figure 15. Sonarqube Static Analysis Result

Figure 16. Sonarqube Result Attachment in SecureTI PMT

Figure 17. Completed Project Management and SDLC Phases

Figure 18. NIST Reference in SecureTI

5. CONCLUSION

IT Project Management and system development are very crucial aspects in every organization. Web application security is a top priority especially in the currently automated environment that semiconductors like Texas Instruments have in its manufacturing areas.

The current issues which were discovered in the study is how to incorporate risk management in the system development process of TI; reassess the importance of secure coding in TI web application development; utilize a TI standard secure coding tool and enforce this in the actual project management flow; how to develop a TI web application that will implement IT Security, project management and SLDC initiatives; and what will be the impact to TI developers if they will be introduced to the proposed web application which will deliver IT Security, project management and SDLC goals.

This study aimed to help TI in practicing NIST strategies to include risk management checkpoints in the system development of TI developers; apply secure coding by means of static code analysis; enforce a standard secure code checking tool to TI web developers as it will be part of the actual project management workflow; introduce a prototype and the project management web application tool mapped with IT Security goals, project management initiatives and a risk focused SDLC model; after development of the prototype, it was evaluated by TI web developers in terms of IT Security alignment, performance and effectiveness.

This study proved that the proposed solution IT Security checkpoints were aligned with the National Institute of Standards and Technology when it comes to risk management and secure systems engineering. The system required IT security fields and risk check items in every phase of the project management and system development steps. Secure code checking was emphasized by SecureTI by applying static code analysis of an open source tool called Sonarqube. During Sonarqube static code analysis, the users were able to correct the detected errors by following the recommended Java standard compliant code suggested by the Sonarqube default quality profile and evidently reduced the number of code errors after the re-scan. The inclusion of Sonarqube in the project management and system development added a layer of review for security and proper code implementations in the scanned application. TI web developers also agreed that SecureTI helped them profoundly reassess the importance of secure code checking via static code analysis tool which also enhanced their system development workflow.

In terms of SecureTI performance, the proposed system was successful in scanning web applications deployed in the local server. The interface was direct and simple and used agile boards for visual status progress transparency. Since SecureTI used an open source community static analysis tool, there was no cost needed for the project to be functional. Access to SecureTI was exclusive for TI (Philippines) web developers and can be deployed using TI virtual private network to ensure security and authorized access.

Lastly, the proponent of the study was able to map and deliver the project management steps together with the SDLC phases and successfully integrated the security check items based from NIST in a single web application only which most certainly helped achieve the goals of this research. With this project, the target organization will greatly benefit and have a baseline of the standard security check items that must be completed in each SDLC step with their corresponding project management phases.

The researched improvement in the project management and SDLC process using SecureTI provided an IT Security focused, useful project management and SDLC interface for Texas Instruments' selected web developers and can be further improved to deliver quality projects that will help the organization to continuously innovate and revolutionize new technologies which will help offer better convenience to the society.

6. RECOMMENDATION

Whilst the study invited encouraging evaluation results, unceasing research on the subject matter is advantageous for the modern automated society. The actual deployment of SecureTI to Texas Instruments using more stable project environment and architecture is highly recommended. The static analysis tool may be explored further on how it will be integrated with external repositories. Browser limitation can also be overcome if the proposed system will be tweaked to run tool scanner using shared development resources. The generic concept of SecureTI can also be optimized for use by other organizations involved in system development and IT project management.

While ensuring that IT Security goals are achieved, every organization should find ways on improving their current workflows may it be in system development or other fields of problem solving. Referring to well-known standards and applying these guidelines can help progress and create innovation relevant to mankind.

ACKNOWLEDGEMENTS

Before anything else, the researcher would like to express her gratitude to the Almighty God for providing her the essential perseverance, knowledge, skills, patience, visible and non-visible resources to accomplish this research. The Gutierrez family, who exhibited enormous patience and support to the researcher in completing the tasks for this project amidst the COVID-19 pandemic. The researcher would like to acknowledge Texas Instruments and the IT, HR, Legal and IT Security for participating and evaluating the proposed system in this study. The researcher would also like to offer her genuine appreciation and thankfulness to the following role models who greatly contributed to make this capstone project possible: Dr. Marlon Tayag, her adviser, for the unwavering support, efforts and guidance towards the completion of this study; Dr. Francisco Napalit, Dean of the School of Computing, for the encouragement and valuable inputs for this research; The cool and amazing panels, Dr. Alma Theresa Manaloto, Dr. Jehan Bulanadi, Dr. Joseph Esquivel, for giving time, energy and patience to evaluate and review this study and for the help in capturing the gaps and missing aspects of this research together with the proper documentation standards. The awesome HAU and PSM in Cybersecurity faculty members and admin who willingly shared their expertises and inspiring experiences all throughout the PSM Program and the diverse and talented PSM in Cybesecurity students who also shared their knowledge, skills and cool ideas during the entire PSM Program.

REFERENCES

[1] "What is a Semiconductor?", Semiconductor Industry Association, nd. Available: https://semiconductors.org/semiconductors 101/what is a semiconductor/. [Accessed 11 January. 2020].

[2] "Top 50 Semiconductor Equipment & Products Companies 2019 Ranked by Sales or Revenue", Globenewswire, 18 Jun. 2019. Available: https://globenewswire.com/news release/2019/06/18/1870313/0/en/Top 50 Semiconductor Equipment Products Companies 2019 Ranked by Sales or Revenue.html. [Accessed 11 January. 2020].

[3] "At a glance | Fact sheet | About Texas Instruments | TI.com", Ti, nd. Available: https://ti.com/about-ti/company/ti-at-a-glance.html. [Accessed 11 January. 2020].

[4] Maskur, Achmad Fahrurrozi, "Static Code Analysis Tools with the Taint Analysis Method for Detecting Web Application Vulnerability | IEEE Conference Publication | IEEE Xplore", International Conference on Data and Software Engineering, pp. 1 2019. Available: https://doi.org/10.1109/icodse48700.2019.9092614. [Accessed 11 January. 2020].

[5] W. Stephen, "Application Developer Roles and Responsibilities", bmc blogs, 19 Oct. 2018. Available: https://bmc.com/blogs/application-developer-roles-responsibilities/. [Accessed 12 January. 2020].

[6] V. Varun, "Risk Management in System Development Life Cycle (SDLC)", International Journal of Advance Research in Computer Science and Management Studies, Mar. 2017. Available: https://ijarcsms.com. [Accessed 12 January. 2020].

[7] "What is Secure Coding and Why is it Important?", Vpnoverview, 18 Oct. 2019. Available: https://vpnoverview.com/internet safety/business/what is secure coding/. [Accessed 12 January. 2020].

[8] K. Gerberding, "Why Should I Use Security Features in Project Management Software?", Wrike, 16 Oct. 2018. Available: https://wrike.com/project-management-guide/faq/why-should-i-use- security-features-in-project-management-software/. [Accessed 12 January. 2020].

[9] "The Systems Development Life Cycle", Misprivate, 2015. Available: https://misprivate.boun.edu.tr/ozdinc/MIS433/ch02.ppt. [Accessed 13 January. 2020].

[10] D. Ozkan, Mishra, "Agile Project Management Tools: A Brief Comparative View", Cybernetics and Information Technologies, vol. 19, no. 4, pp. 17–25, 2019. Available: https://doi.org/https://doi.org/10.2478/cait 2019 0033. [Accessed 13 January. 2020].
[11] Gupta, A.,2017. Project Management Tool. Available: http://csus dspace.calstate.edu/bitstream/handle/10211.3/190305/Project_Management_Tool.pdf?sequence= 1
[12] R. Leal, "ISO 27001 – integrating A.14 controls with SDLC", Doi, 24 Jan. 2017. Available: https://advisera.com/27001academy/blog/2017/01/24/how to integrate iso 27001 a 14 controls into the system software development life cycle sdlc/. [Accessed 26 January. 2020].
[13] B. Evans, "The System Development Life Cycle: A Phased Approach to Application Security", Security Intelligence, 7 Jan. 2019. Available: https://securityintelligence.com/the system development life cycle a phased approach to application security/. [Accessed 2 February. 2020].
[14] I. Meschankina, "The Software Development Life Cycle: Phases And Methodologies", Producttribe, 19 Mar. 2018. Available: https://producttribe.com/project management/sdlc guide. [Accessed 2 February. 2020].
[15] S. Hasan, UA. Khan, "SDLC models", International Journal of Computer Applications, vol. 178, no. 53, pp. 1–1, 2019. Available: https://csus dspace.calstate.edu/bitstream/handle/10211.3/190305/Project_Management_Tool.pdf?sequence=1.[Accessed 2 February. 2020].

AUTHORS

Amelie Anne V. Gutierrez: She received her BS in Computer Science from Holy Angel University, Pampanga, Philippines in 2013. She is currently working as an application developer in one of the largest semiconductor companies in the world. She is also acting the role of an IT Security Ambassador.