Performance of Techno Management View and Software Product SecurityFramework

Padala Kavitha,

Research Scholar, Computer Science and Engineering, OPJS University, Churu, Rajasthan.
Dr.Vijay Pal Singh ,
Assistant Professor, Computer Science and Engineering, OPJS University, Churu, Rajasthan.

Abstract

A framework focuses on the security consideration since the inception of the project. It is evident from the literature that there exist many frameworks that focus on the overall security of the system. We discuss the related work as the background. Various technical and management aspects of security play a vital role in developing secured systems. We propose techno- management view of Secured Software Development Process (SSDP) that links technical and management aspects of security and describe it. To help systematic considerations of various security aspects, we present generalized *Software Security Product (SPS)* framework as another proposal. The three layers of the framework are discussed followed by a case study . To evaluate the security considerations during the development process through *SPS* framework, we propose Security Factor, F_S Results.

1. Introduction

Since the revolution of Internet, businesses and individuals rely more on computers. At the same time, it has spawned many threats to the data and information leading to security and privacy concerns. The threats mainly owe to the vulnerabilities present in the software system. These vulnerabilities occur due to insecure design, coding and configuration. Security is an important non-functional attribute of software that needs to be included since the inception of the project. Organizations that consider security as one of the essential aspects can better apply the security policies and support in development of secured software product. The security processes provide the guidance regarding the types of controls and procedures that can be applied for developing secured software product.

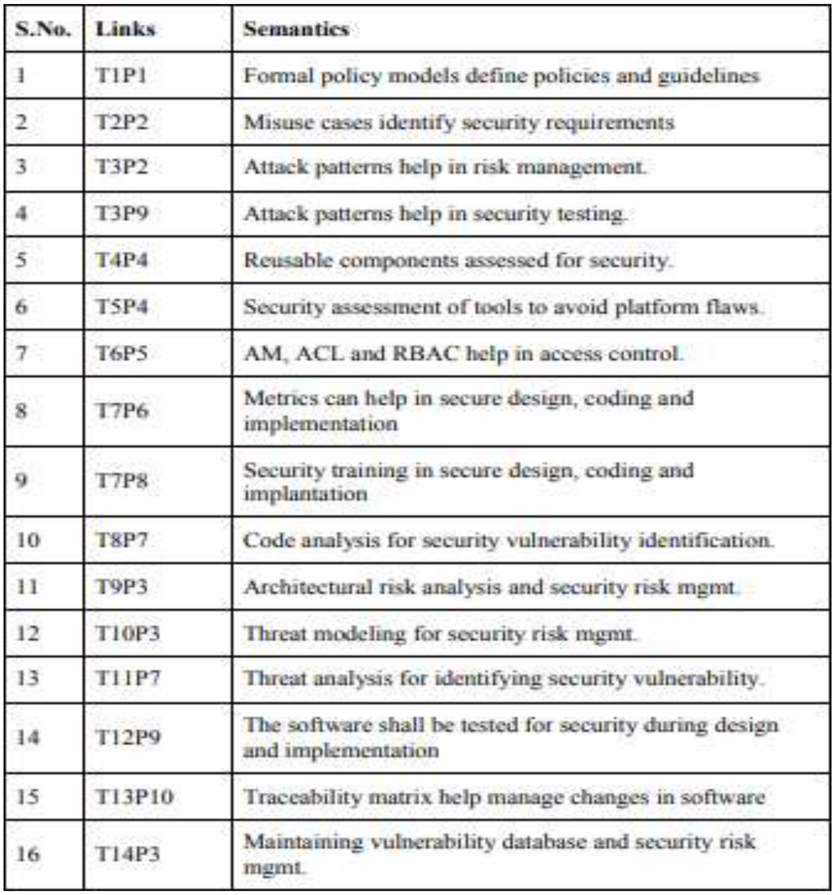
In the current scenario, many security tools and technologies are being used during secured development process. These include attack trees, misuse cases, threat modeling etc. To improve the overall security posture, organizations also focus on the management aspects that involve use of guidelines, compliances, metrics, etc. Thus, security has become an apprehension for technical and management view point. It has been observed that the technical and management aspects have been concerned with the secured development and information security respectively. It creates a gap between these two aspects during the development of software. Moreover, the organizations need a framework that can help to incorporate the various technical and management aspects of security. The existing frameworks address varied security issues related to training, governance, risk management, physical infrastructure etc.

2. Software Product Security (*Sps*) Framework

Now, we propose the generalized framework (*SPS* framework) for the development of secured software. It is a three layered structure viz. control, security aspects and development layers as depicted. These layers are explained in detail as follows:

Table 1: Semantic Links between Process Management

ISSN:0975-3583,096-2833 VOL12,ISSUE01,2021



S.No.	Links	Semantics
1	T1P1	Formal policy models define policies and guidelines
2	T2P2	Misuse cases identify security requirements
3	T3P2	Attack patterns help in risk management.
4	T3P9	Attack patterns help in security testing.
5	T4P4	Reusable components assessed for security.
6	T5P4	Security assessment of tools to avoid platform flaws.
7	T6P5	AM, ACL and RBAC help in access control.
8	T7P6	Metrics can help in secure design, coding and implementation
9	T7P8	Security training in secure design, coding and implantation
10	T8P7	Code analysis for security vulnerability identification.
11	T9P3	Architectural risk analysis and security risk mgmt.
12	T10P3	Threat modeling for security risk mgmt.
13	T11P7	Threat analysis for identifying security vulnerability.
14	T12P9	The software shall be tested for security during design and implementation
15	T13P10	Traceability matrix help manage changes in software
16	T14P3	Maintaining vulnerability database and security risk mgmt.

2.1 Control Layer

The control layer is the top layer of *SPS* framework. It illustrates the managerial control of the entire SDP with the help of governance. Controlling keeps the unwanted users out of the system by providing right kind of access to authorized users. Security governance defines the roadmap for developing secured software, to regulate the software development process and to meet the security objectives of business. Such practices regulate SDP to organize, manage and measure the software security plans and objectives. Security governance also ensures that all the responsible people support the implementation of the security controls. The governance should provide guidance on how the users and the access controls of the roles should be managed.

Effective governance should be able to incorporate risk assessment and the security considerations in all phases of SDP. The governance provides guidance for achieving security attributes. On this account, the key role of the governance is to decide policies and guidelines forimplementing security in the software during development.

Policy

Policies outline specific security requirements of an organization that must be met by the software system. One of the key features of governance is to define security roles and responsibilities for the different types of users. Once the application is deployed, the deployer can map the roles to security identities in the operational environment. The governance needs to develop and implement security policies, such as PoLP, separation of duties etc. PoLP should be applied to define minimum permission associated with a role. It can be important in meeting the integrity objectives. The governance should also define the policy for separation of duties, in which the task is disseminated among multiple people for internal control. It may also help in prevention of fraud and errors. The policies can also be developed for password, inactive sessions, account lockouts, hardware devices, communication, risk assessment, information security according to sensitivity etc.

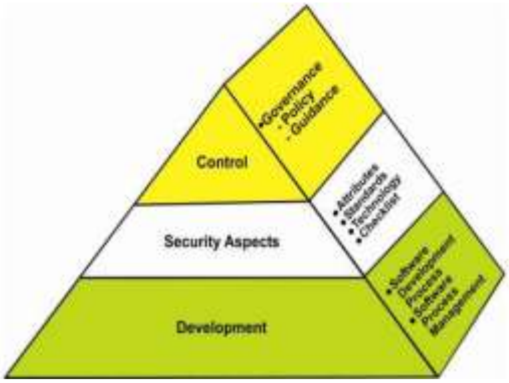


Fig. 1: Software Product Security Framework

Guidance

The governance provides guidance on the way to implement specific security policies, roles and responsibilities, and security controls effectively. The guidance can also be provided for documenting security requirements, documenting and validating security capabilities, and promoting international cooperation in the area of IT security. The governance needs to guide the development team for implementing the industry compliance, if applicable, such as HIPAA, Sarbanes-Oxley etc. The guidance also includes educating the developers regarding safe security techniques to minimize successful attacks as mentioned in security standards. The process of guidance also includes training the managers, developers and the end-users to help understand the security issues and the need for the implementation.

2.2 Security Aspects Layer

It is the middle layer of *SPS* framework that aids in identification of security features the software should possess. It helps in recognizing the security mechanisms that can be practiced while developing software. The security aspects such as security attributes, standards, technology and security checklist can be considered while designing the software as shown in Fig. 1. The security aspects can promote gathering security requirements and focus on security objectives of the organization as well as software. The security aspects considered are discussed below:

Security Attributes

Security attributes signify the security properties that help to achieve security goals of software and data related to the organization. The main security attributes includes confidentiality, integrity and availability (CIA). Confidentiality and integrity are non-functional attributes. Security properties such as authentication, authorization, non-repudiation, and privacy relate to the people who use the information provided by the system. Authentication, authorization and audit are also external system functional attributes. Other attributes related to security are reliability, availability, perform ability and safety.

Security Standards

The security standards facilitate the development of more secured product by defining policies for managing data and information, criteria for evaluating the security measures applied, and assessing and monitoring the security practices applied throughout SDP. The security standards can relate to process security or product security. The commonly used security standards have been specified in Common Criteria (CC), National Institute of Standards and Technology (NIST), Open Web Application Security Project (OWASP), etc. CC helps in establishing security requirements for products and systems and has emerged as collaboration for national security standards organization of USA, Canada, France, Germany, India, and so on. NIST publishes many standards that may be useful to security managers, system designers and development team. OWASP focuses on the secure design, development and testing of Web applications. Another commonly used standard is ISO 17799 that comprises of comprehensive set of controls for information security. The areas covered by ISO 17799 are security policy, organization and information security, asset management, access control, physical and environmental security, compliance etc.

Security Technology

A number of tools have been used for security requirements gathering, designing, coding and testing such as Security Requirements Quality Engineering (*SQUARE*), threat modeling, threat

analysis, attack trees etc. During coding, the vulnerabilities can be identified using static and dynamic code analyzers. The resultant software can be black-boxed and white-boxed for testing functional and non-functional security requirements. Penetration testing, fuzz testing etc. can be used for evaluating the robustness of software to handle malicious attacks. The various security technologies are discussed.

Security Checklist

Brainstorming, experience, and knowledge have been the normally used methods to gather security requirements. A formal approach such as Software Security Checklist can also be used that cover the areas regarding requirements and specification, design and code issues, maintenance and decommissioning of software and system. Some of the issues of checklist include determining stakeholders and their security requirements, sources for software security risks, security of middleware, check input points of the code and so on. Checklist developed by CERN illustrates the hints and tips for all the phases of secured software development such as architecture, design, implementation, coding, and post implementation. Checklist for code review comprises of structure, documentation, variables, arithmetic operators, loops and branches etc. Thus, a checklist for security can be developed based on the organization's own security requirements.

2.3 **Development Layer**

Development layer is the lowest layer of *SPS* framework that facilitates the implementation of the security aspects of middle layer by help of control layer. Development requires two features for secured development; Firstly, software development process and another one, software process management. The development layer focuses on the security aspects identified during the middle layer of our framework.

Software Development Process

Some of the secured development processes include *SDL*, *CLASP* and *Touchpoints* etc. These emphasize on achieving security attributes through the use of fundamental practices. Similarly, SafeCode discusses the fundamental secured software development practices that can be adopted across varied development environments. The security aspects discussed in Section 5.3 can be applied during most of the software development processes followed. The development methodologies adopted by the development team can be either linear or iterative in nature. The linear development methodologies followed have been waterfall model, iterative development frameworks such as prototyping and Rapid Application Development (RAD) models or combination of two such as incremental and spiral models. The iterative and incremental are essential part of Unified development Process (RUP), the Dynamic Systems Development Method, Extreme Programming and generally the agile software development frameworks. The iterative methods possess feedback property that helps to improve quality by providing feedback. The feedback facilitates in uncovering security issues early that may lead to disaster.

Software Process Management

The software development process can be improved by the software management models, the software metrics and the process improvement models. Capability Maturity Model (CMM) and ISO 9001 do not provide any methodology to develop secured software, but focus on how well the organizations follow the well defined SDP. The other software management processes to regulate the software development process include 15504, ISO 9000, ISO 15504 or Software Process Improvement Capability Determination (SPICE), V-model etc. An extension to CMM is System Security Engineering Capability Maturity Model (SSE-CMM). It describes the characteristics of an organization's security engineering process that must exist to ensure good security engineering. SSE-CMM addresses security engineering activities that span throughout SDLC. It also includes tools to evaluate security engineering practices.

Team Software Process (*TSP*) facilitates the development of self directed teams that can plan and track work of the team, establish goals and schedules, assess risk, and team plans thereby helping in mature SDP. Personal Software Process (*PSP*) aims on improving individual’s tasks by using a "disciplined, data-driven procedure" during software development activities [HUM99, DM03]. To increase the overall efficiency of development process, *TSP* in coordination with *PSP* can be used to improve the quality level and productivity of the software development team. It may result in improving the quality of software.

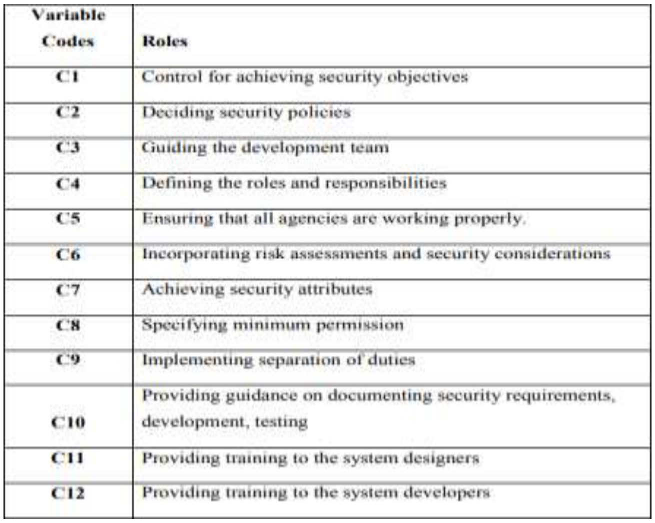
A variety of metrics are available which can help to improve and control the software development process such as program size, cyclomatic complexity, function point analysis, bugs per line of code etc. These metrics help in measuring size/ complexity, schedule, quality and cost. Such metrics are not applicable for measuring the security attribute of the software. Some of the quality metrics such as defect and reliability metrics may measure the security attribute with some limitations. The metrics related to security have been discussed.

2.4 **A Case Study**

We have attempted to develop software product security framework that may support systematic consideration of security during the development process. The framework shall help provide security through the governance and the security aspects to be considered during the development process, we have identified twelve key roles of control layer; five important mechanisms of security aspects layer; and four major mechanisms of development layer. Some of the key roles of control layer considered are Control for achieving security objectives, deciding security policies… Providing training to the system developers as shown in Table 2. The mechanisms describing security aspects layer are Contribution of security attributes, Consideration of security attributes in requirements gathering,…, Security checklist for requirements gathering, code, and testing issues as stated in Table 3. The mechanisms describing development layer are Types of software processes followed, Use of *TSP* and *PSP*, Use of CMM/ ISO and Use of metrics as mentioned in Table 4.

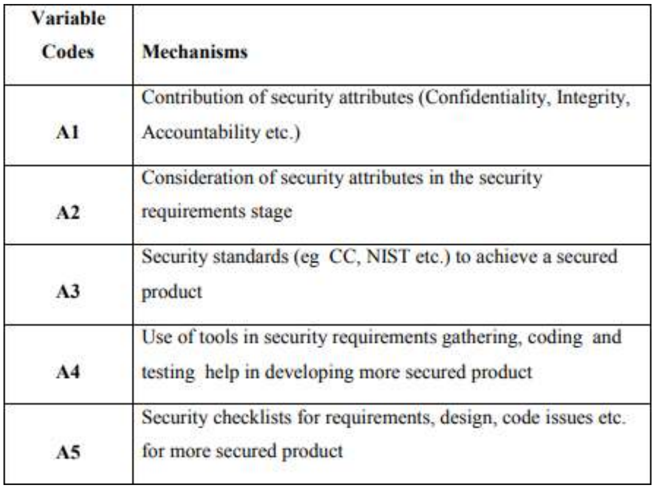
To examine the *SPS* framework, an empirical survey has been conducted using a self-designed questionnaire. The questionnaire consists of roles of control layer, mechanisms describing security aspects and development. The samples are comprised of the responses influenced by the security practices followed at various organizations by IT professionals ranging from moderate to high experience. Based on the responses, we tried to judge the importance of the three layersas well as roles and mechanisms describing the layers of the *SPS* framework.

Table 2: Roles of Control Layer



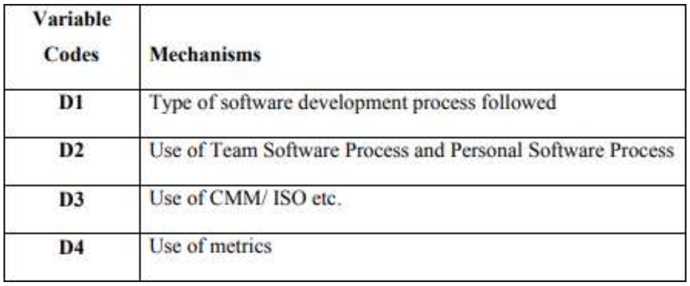
Variable Codes	Roles
C1	Control for achieving security objectives
C2	Deciding security policies
C3	Guiding the development team
C4	Defining the roles and responsibilities
C5	Ensuring that all agencies are working properly.
C6	Incorporating risk assessments and security considerations
C7	Achieving security attributes
C8	Specifying minimum permission
C9	Implementing separation of duties
C10	Providing guidance on documenting security requirements, development, testing
C11	Providing training to the system designers
C12	Providing training to the system developers

Table 3: Mechanisms of Security Aspects Layer



Variable Codes	Mechanisms
A1	Contribution of security attributes (Confidentiality, Integrity, Accountability etc.)
A2	Consideration of security attributes in the security requirements stage
A3	Security standards (eg CC, NIST etc.) to achieve a secured product
A4	Use of tools in security requirements gathering, coding and testing help in developing more secured product
A5	Security checklists for requirements, design, code issues etc. for more secured product

Table 4: Mechanisms of Development Layer



Variable Codes	Mechanisms
D1	Type of software development process followed
D2	Use of Team Software Process and Personal Software Process
D3	Use of CMM/ ISO etc.
D4	Use of metrics

3. Security Factor Of *Sps* Framework

In this section, we attempt to develop a mathematical model that can help to estimate the security concerned within some system using *SPS* framework through Security Factor (*Fs*). We assume that the aforesaid task executes under some assumptions. Let m, n and p be the total measures of each layer. Let W1, W2 and W3 be the weights assigned to the measures of control, security aspects and development layer respectively. The weights can assume binary values (i.e. 0 and 1) for presence and absence of the respective feature. Then, security factor of *SPS* framework can be computed using following computation technique:

Let us define roles and mechanisms of the three layers of *SPS* framework as the measure of the function W(i, j) for control, aspect and development layer.W(i, j) = 1, if j th measurable security feature at i th layer exists= 0, otherwise

Then, Security through Control Layer, $S_{CL} = \sum_{j=1}^{m} W(1, j)/m$ Security through Aspects Layer, $S_{AL} = \sum_{j=1}^{n} W(2, j)/n$

And, Security through Development Layer, $S_{DL} = \sum_{j=1}^{p} W(3, j)/p$ where and m, n, p > 0.

Then, Security Factor (Fs) can be given as: $F_s = \frac{1}{3}\left[S_{CL} + S_{AL} + S_{DL}\right]$ or

$F_s = \frac{1}{3}\left[\sum_{j=1}^{m} W(1, j)/m + \sum_{j=1}^{n} W(2, j)/n + \sum_{j=1}^{p} W(3, j)/p\right]$ The value of Fs can lie between 0 and 1 i.e. $0 \leq Fs \leq 1$

3.1 A Case Study

Consider a web portal intended to deliver Content Management System (CMS) to the journalists developed using *SPS* framework. The various types of users include editors, authors, journalists, photographers, designers, and community managers. The portal allows content planning of section, categories and blogs; header and template editor; maintenance to perform cache cleaning; security panel for log tracking and periodic password changes; etc. The portal also facilitates job editor, canvas editor, SEO, related content linker, poll and survey editor, and allows only authorized citizens to send reports from their mobiles. It also provides marketing tools that allows displaying banners and ads. It allows keeping track of individual and collective efforts over sections, categories, etc., sends reports and alerts over e-mail to group members, visitor information, and mobile/ pad interaction. The editor is allowed to approve/ modify/ publish the news in certain time frames.

Considering the roles and mechanisms being adopted during the development process of the portal as mentioned in Table 5.2 – 5.4 having m=12, n=5 and p=4. The weights associated with the variable codes at different layers for the new portal are depicted in Table 5.5 (a-c). Using Equation 5.1, the Security Factor F_S can be computer as

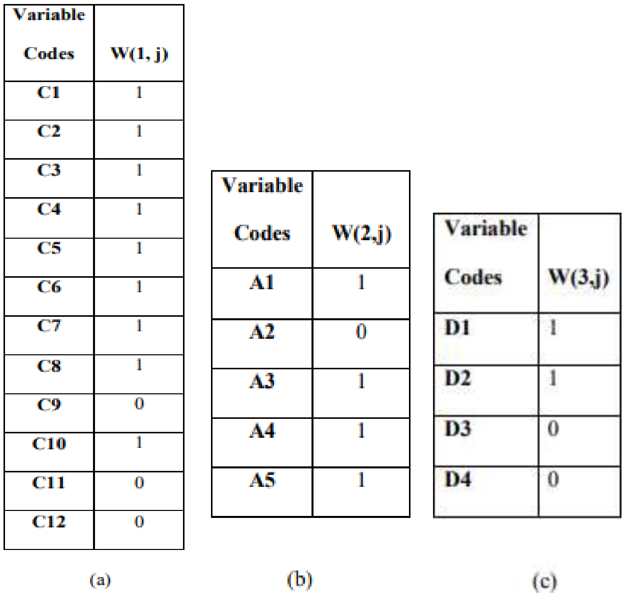
$$S_{CL} = \sum_{j=1}^{m} W(1,j)/m = 1+1+1+1+1+1+1+1+0+1+0+0 = 9/\ 12$$

$$S_{AL} = \sum_{j=1}^{n} W(2,j)/n = 1+0+1+1+1 = 4/\ 5$$

$$S_{DL} = \sum_{j=1}^{p} W(3,j)/p = 1+1+0+0 = 2/\ 4$$

Thus, Fs = [9/ 12 + 4/ 5 + 2/ 4]/ 3 = 0.68

Table 5: Sample Weights for (a) Control Layer, (b) Security Aspects Layer, and (c)Development Layer



Variable Codes	W(1, j)
C1	1
C2	1
C3	1
C4	1
C5	1
C6	1
C7	1
C8	1
C9	0
C10	1
C11	0
C12	0

(a)

Variable Codes	W(2,j)
A1	1
A2	0
A3	1
A4	1
A5	1

(b)

Variable Codes	W(3,j)
D1	1
D2	1
D3	0
D4	0

(c)

ISSN:0975-3583,096-2833 VOL12,ISSUE01,2021

4. Results

We have attempted to identify the various technical and process management aspects that may be considered for developing secured software systems. The semantic links have been established to identify the relationship between the technical and process management aspects leading to techno-management view of SSDP. We analyzed the data for evaluating the links on the basis of feasibility of actions and have been illustrated in the graph in Fig. 5.3. It has been established that formal policy models define policies and guidelines. Access Management, Access Control Lists, and Role Based Access Control help in access control. The attack patterns facilitate in risk management as well as security training while the architectural risk analysis and threat modeling supports security risk management. The above links accounts to 100% acceptance by the professionals. The other links are also accepted by more than 75% professionals such as security can be identified with the help of misuse cases and attack patterns. The reusable components and the development platform must be assessed for security. Secure designing and implementation can be supported by use of metrics and by providing training to development team. The security vulnerabilities in the design and code can be identified through threat and code analysis. The techno-management view demonstrates the relation between process management and technical aspects that may support the managers and the development team to bridge the gap between the aspects. Thus, the technical and process management aspects of SSDP may be considered helpfulin developing secured software.

Further, we developed *SPS* framework with an aim to consider security as an overall organizational perspective during the development of secured software product. Let, control, security aspects and development layers are represented by C, A and M while, moderate and high experienced professionals are represented by M and H respectively. Thus, in the graph (Fig. 5.4), CM implies view of moderately experienced professional for the roles of control layer. Similarly, the layer consideration and the experience of professionals are denoted by CH, AM, AH, DM and DH. As illustrated in Fig. 5.4, following observations are made:

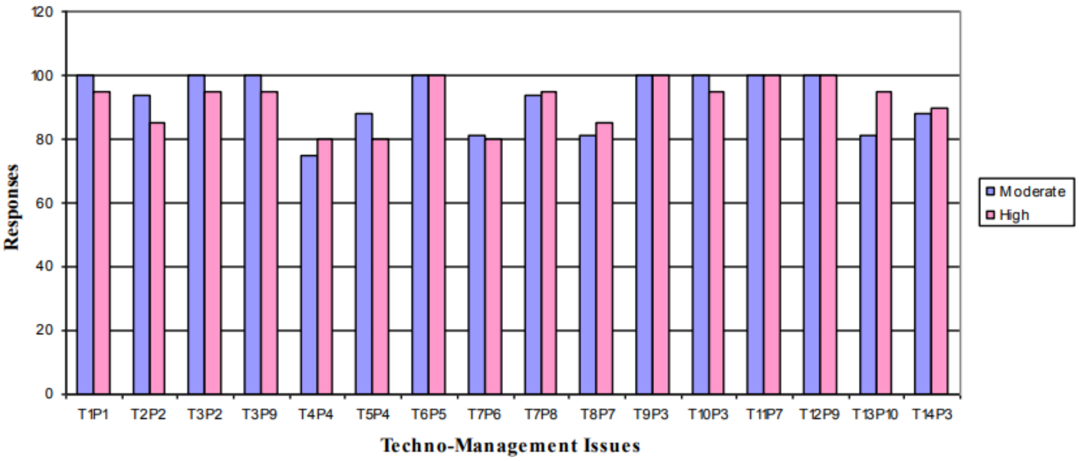


Fig. 5.3: Data Analysis of Techno-management View of Security

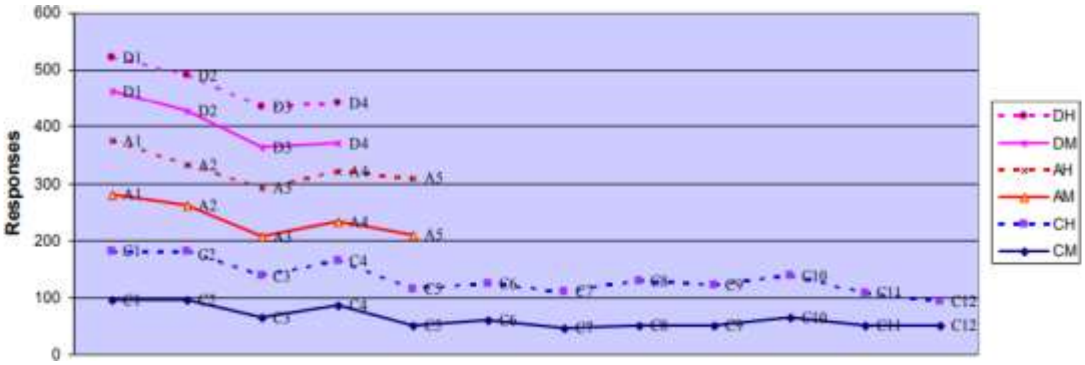


Fig 5.4: Perception of Various Layers of SPS Framework

- All the professionals indicated the importance of the three layers of the framework for secured software development.

- Most of the functions of control layer have been marked essential to achieve proper controlling of the software development process.

- Control layer may support in defining security policy and objectives of the business. It can assign roles and responsibilities, provide access control and identify security requirements thereby helping to incorporate security during the development.

- Some of the functions of control layer have been considered less important by moderately experienced professionals such as proper working of all the agencies responsible for secured development specifying minimum permission, separation of duties, and providing training to the system designers and developers

- Highly experienced professionals too considered training to be less important. This may be due to the fact that training the software developers regarding security is usually not possible due to tight deadlines of the business; but, it may help avoid certain known flaws.

- The security aspects layer can support in defining important security attributes that act as benchmark for attaining security attributes. Security aspects such as use of standards, security tools, security checklists etc. can help reduce vulnerability and have been considered important for secured software.

- The development mechanisms such as type of software development process followed and use of *TSP*, *PSP*, CMM/ ISO, metrics etc. have been considered supportive for the development of secured software.

- Use of metrics is mostly the aspect of security management. Hence, moderately experienced developers do not consider it important. The use of metrics during development can help the governance to achieve the security attributes as described in security aspects layer.

- The high experience group do not emphasize on the type of software development process for secured software. It is in line with the fact that the developers mostly rely on secure deployment.

Conclusion

A mathematical model has been developed to identify the security achieved through the use of the *SPS* framework. Based on the case study, it can be observed that 0.75 or 75% security can be achieved through the roles provided by the control layer, 0.8 of 80% security owe to the security aspects layer while 0.5 or 50% security owe to the development layer individually. When the three layers of the framework are combined, then Security Factor, F_S = 0.683 emphasize that the software developed using the *SPS* framework is 68.3% secured. Based on the security considerations during the development process, security gap owe to lack of security training to the developers and designers. Training shall help in secured design and coding. The development team should also incorporate separation of duties to enhance security consideration. *TSP* and *PSP*, CMM/ ISO etc. shall improve the overall quality of the product including security. The use of metrics can improve the security consideration of the development process as well as softwareproduct.

Secured development is the result of incorporating security during the development process. We have started the process with enhancing the understanding of technical and process management aspects required to support the development process. By establishing the links between the aspects, we designed techno-management view of security. It can be established that the techno- management view of security bridges the gap between the process management and the technical aspects of security that have to be considered during software development. Further, a generalized *SPS* framework has been developed that considers security as an overall perspective in developing secured software. The *SPS* framework shall act as a blueprint for security concerns. It may support management and the developers in considering security as an overall organizational perspective based on all three layers of the framework. It may help consider security from the pre-development phase itself and provide guidelines to include security within SDP to produce a secured product. The overall security factor of the software can be computed using Equation 5.1 to evaluate security concerns throughout the development process.

References

1. Landwehr, C. E., Bull, A. R., McDermott, J. P. and Choi, W. S., ―A Taxonomy of Computer Program Security Flaws, with Examples‖, ACM Computing Surveys, Vol. 26, No. 3, Sep. 1994, pp. 211-254.

2. Lebanidze, E., ―Securing Enterprise Web Applications at the Source: An Application Security Perspective‖, OWASP, [Online] Available: https://www.owasp.org/images/8/83/Securing_ Enterprise_Web_Applications_at_the_Source.pdf.

3. Lehman, J.,―Use Offense to Inform Defense. Find Flaws Before the Bad Guys do‖, Penetration Testing, SANS Institute, 2011, [Online] Available: http://pen-testing.sans.org/resources/papers/ gwapt/robotstxt-108867.

4. ―India Drafts New Policy Regulations‖, Privacy and Information Security Law Blog, Hunton and Williams LLP, May 18, 2011, [Online] available: http://www.huntonprivacyblog.com/2011/05 /articles/india-drafts-new-privacy-regulations/.

5. IT Security and Crime Prevention Methods, [Online] Available: http://www.interpol.int/Public/TechnologyCrime/CrimePrev/ITSec urity.asp#1, 2003.

6. Bartoldus, M., ―Security in the SDLC: It Doesn't Have to be Painful‖, Gotham Digital Science,Ltd., 2010, [Online] Available: https://www.owasp.org/images/b/b7/SSDLC_painful_- _owasp.pdf.

7. McConnell, S., ―From the Editor-An Ounce of Prevention‖, IEEE Software, Vol.18, No. 3, May 2001, pp. 5-7.

8. Mehta, D.M., ―Effective Software Security Management‖, White paper, OWASP, [Online] Available: https://www.owasp.org/ images/2/28/Effective_Software_Security_Management.pdf.

9. Meland, P. H., and Jensen, J., ―Secure Software Design in Practice‖, In Third International Conference on Availability, Reliability and Security (ARS08), IEEE, Mar. 2008, pp. 1164-1171.

10. Payne, S.C., A Guide to Security Metrics, SANS Institute InfoSec Reading Room, Jun. 19, 2006, [Online] Available: http://www.sans.org/reading_room/whitepapers/auditing/ a_guide_to_security_metrics_55.