Volume 6, Issue 5, May 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Security of Software Development Life Cycle Processes

Tamanna Tayal
PhD Scholar, JJT University,
India

Dr. Yogesh Kumar Sharma
Associate Professor, JJT University,
India

Abstract: This articles present the information about existing processes, standards, life cycle models, frameworks and methodologies that support or could support secure software development. This includes software engineering process group (SEPG) members, software developers, and managers seeing information about existing software development life cycle (SDLC) processes that address security.

Keywords: SDLC, processes, security Risk Identification, security engineering activities.

I. INTRODUCTION

As the use of the Internet and networked systems become more pervasive, the importance of developing secure software increases. The purpose of this technical note is to present overview information about existing processes, standards, life cycle models, frameworks, and methodologies that support or could support secure software development. Where applicable and possible, some evaluation or judgment is provide.

The target of this document includes software engineering process group (SEPG) members who want to integrate security into their standard software development processes. It is also relevant for developers and managers looking for information on existing software development life cycle (SDLC) processes that address security. Technology or content areas described include existing frameworks and standards such as the Capability Maturity Model® Integration (CMMI®) frame work, the FAA-iCMM, the Trusted CMM/Trusted Software Methodology (T-CMM/TSM), the Systems Security Engineering Capability Maturity Model (SSE-CMM), in addition to existing processes such as the Microsoft Trustworthy Computing Software Development Lifecycle, the Team Software Process SM for Secure Software Development (TSPSM-Secure), Correctness by Construction, Agile Methods, and the Common Criteria.

II. CAPABILITY MATURITY MODELS

Capability Maturity Models provide a reference model of mature practices for a specified engineering discipline. An organization can compare their practices to the model to identify potential areas for improvement. The CMMs provide goal-level definitions for and key attributes of specific processes (software engineering, systems engineering, security engineering), but do not generally provide operational guidance for performing the work. In other words, they don't define processes, they define process characteristics; they define the *what*, but not the *how*: "CMM- based evaluation is not meant to replace product evaluation or system certification. Rather, organizational evaluation are meant to focus process improvement efforts on weaknesses identified in particular process areas".

III. CAPABILITY MATURITY MODEL INTEGRATION (CMMI)

Capability Maturity Model Integration (CMMI) framework helps organizations increase maturity of their processes to improve long-term business performance. The CMMI provides the latest best practices for product and service development, maintenance, and acquisition, including mechanisms to help organizations improve their processes and provides criteria for evaluating process capability and process maturity. Improvement areas covered by this model include systems engineering, software engineering integrated product and process development, supplier sourcing and acquisition. The CMMI has been in use for more than three years and will eventually replace its predecessor, the Capability Maturity.Model for Software (SW-CMM), which has been in use since the mid-1980s. As of June 2005, the Software Engineering Institute (SEI) reports that 782 organizations and 3250 projects have reported results from CMMI-based appraisals [SEI 05a]. Beginning in 1987 through June 2005, 2,859 organizations and 15,634 projects have reported results from SW-CMM-based appraisals and assessments [SEI 05b]. The CMMI addresses four categories for process improvement and evaluation. Each category includes several Process Areas. CMMI addresses project management, supplier management, organization-level process improvement as well as training, quality assurance, measurement, and engineering practices. However, it does not specifically address the four areas mentioned earlier (security risk management, security engineering practices, security assurance, and project/organizational processes for security), although it is not unreasonable to assume that each of these are special cases of practices already addressed by the CMMI

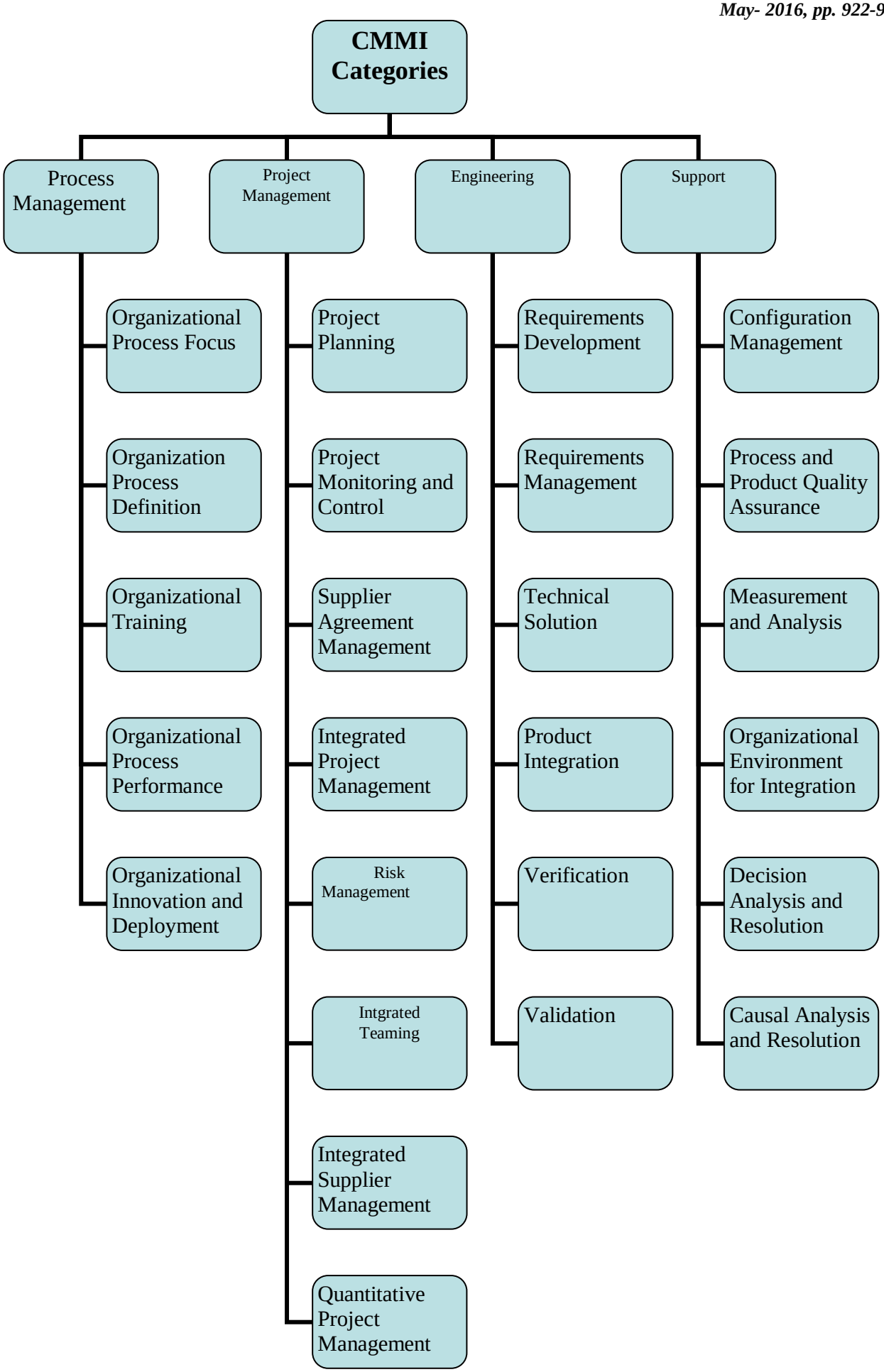

Figure 1: Process Areas of the CMMI Framework

IV. FEDERAL AVIATION ADMINISTRATION INTEGRATED CAPABILITY MATURITY MODEL (FAA-ICMM)

The FAA-iCMM was developed and is widely used by the Federal Aviation Administration. It provides a single model of best practices for enterprise-wide improvement, including outsourcing and supplier management. The latest version includes process areas to address integrated enterprise management, information management, deployment/ transition/disposal, and operation/support. The FAA-iCMM integrates the following standards and models:

- ISO 9001:2000
- EIA/IS 731
- Malcolm Baldrige National Quality Award
- President's Quality Award
- CMMI-SE/SW/IPPD and CMMI-A
- ISO/IEC TR 15504, ISO/IEC 12207, and ISO/IEC CD 15288

FAA-iCMM is organized into three main categories and 23 Process Areas [FAA 01]. The FAA-iCMM addresses project management, risk management, supplier management, information management, configuration management, design, and testing, all of which are integral to a secure SDLC. However, the FAA-iCMM does not address security specifically in any of these areas. Just as with the CMMI, FAA-iCMM includes a generic set of best practices that do not specifically address security concerns

V. PROPOSED SAFETY AND SECURITY ADDITIONS TO THE CMMI AND FAA-ICMM

Because of the integration of process disciplines and coverage of enterprise issues, the CMMI and the FAA-iCMM are used by more organizations than the SSE-CMM; yet the two integrated models contain gaps in their coverage of safety and security. As a result, some organizations with in the FAA and the Department of Defense(DoD) have sponsored a joint effort to identify best safety and security practices for use in combination with the FAA-iCMM and the CMMI. The proposed Safety and Security additions to the FAA-iCMM and the CMMI identify standards-based practices expected to be used as criteria in guiding process improvement and in appraising an organization's capabilities for providing safe and secure products and services. The proposed safety and security additions include the following four goals and 16 practices:

1. Goal 1 – An infrastructure for safety and security is established and maintained.

a. Ensure safety and security awareness, guidance, and competency.

b. Establish and maintain a qualified work environment that meets safety and security needs.

c. Ensure integrity of information by providing for its storage and protection, and controlling access and distribution of information.

d. Monitor, report, and analyze safety and security incidents and identify potential corrective actions.

e. Plan and provide for continuity of activities with contingencies for threats and hazards to operations and the infrastructure.

2. Goal 2 – Safety and security risks are identified and managed.

a. Identify risks and sources of risk attributable to vulnerabilities, security threats, and safety hazards.

b. For each risk associated with safety or security, determine the causal factors, estimate the consequence and likelihood of an occurrence, and determine relative priority.

c. For each risk associated with safety or security, determine, implement and monitor the risk mitigation plan to achieve an acceptable level of risk.

3. Goal 3 – Safety and security requirements are satisfied.

a. Identify and document applicable regulatory requirements, laws, standards, policies, and acceptable levels of safety and security.

b. Establish and maintain safety and security requirements, including integrity levels, and design the product or service to meet them.

c. Objectively verify and validate work products and delivered products and services to assure safety and security requirements have been achieved and fulfill intended use.

d. Establish and maintain safety, security assurance arguments and supporting evidence throughout the life cycle.

4. Goal 4 – Activities and products are managed to achieve safety and security requirements and objectives.

a. Establish and maintain independent reporting of safety and security status and issues.

b. Establish and maintain a plan to achieve safety and security requirements and objectives.

c. Select and manage products and suppliers using safety and security criteria.

d. Measure, monitor and review safety and security activities against plans, control products, take corrective action, and improve processes.

Other key standards and methods that apply to developing secure software but have not been summarized in this technical note include

- ISO/IEC 15288 for System Life Cycle Processes, available from http://www.iso.org
- ISO/IEC 12207 for Software Life Cycle Processes, available from http://www.iso.org

- ISO/IEC 15026 for System and Software Integrity Levels, available from http://www.iso.org
- Cleanroom Software Engineering [Linger 94, Mills 87]

VI. CONCLUSION

This technical approach demonstrates that although there are several processes and methodologies that could support secure software development, very few are designed specifically to address software security from the ground up. The notable exceptions are Microsoft's Trustworthy Computing SDL and the SSE-CMM. As software security becomes a more important issue in an increasingly networked world, more processes that explicitly address the four focus areas identified in this paper (security engineering activities, security assurance activities, security organizational and project management activities, and security risk identification and management activities) should achieve visibility.

REFERENCES

[1] A**.** Alliance (2001)," Manifesto for Agile Software Development", http://agilemanifesto.org.
[2] "Common Vulnerabilities and Exposures", http://www.cve.mitre.org/ (2005).
[3] Federal Aviation Administration," The Federal Aviation Administration Integrated Capability Maturity Model® (FAA-iCMM®)", Version 2.0. Washington, DC: Federal Aviation Administration, September 2001. http://www.faa.gov/aio/common/documents/iCMM /FAA-iCMMv2.htm
[4] Goldenson, R. Dennis & Gibson, L. Diane (2003)," Demonstrating the Impact and Benefits of CMMI: An Update and Preliminary Results (CMU/SEI-2003-SR-009)", Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.
[5] Humphrey, Watts S.(2002)," *Winning with Software: An Executive Strategy*. Boston", MA: Addison Wesley, (ISBN 0201776391).
[6] Linger, R. C.(1994), "Cleanroom Process Model." IEEE Software 11, 2 ,page no: 50–58.
[7] IEEE.,"IEEE Standard Glossary of Software Engineering Terminology*",* ANSI/IEEE Std 610.12-1990. February 1991.
[8] Criteria. http://www.commoncriteriaportal.org/ (2005).

