

## Chapter 2

# Assimilating and Optimizing Software Assurance in the SDLC: A Framework and Step-Wise Approach

**Aderemi O. Adeniji**

*University of North Carolina at Charlotte, USA*

**Seok-Won Lee**

*University of North Carolina at Charlotte, USA*

### **ABSTRACT**

*Software Assurance is the planned and systematic set of activities that ensures software processes and products conform to requirements while standards and procedures in a manner that builds trusted systems and secure software. While absolute security may not yet be possible, procedures and practices exist to promote assurance in the software lifecycle. In this paper, the authors present a framework and step-wise approach towards achieving and optimizing assurance by infusing security knowledge, techniques, and methodologies into each phase of the Software Development Lifecycle (SDLC).*

### **INTRODUCTION**

Software Assurance is steadily gaining ground in the Information Technology industry. The notion of proving secure software while supporting organization and system priorities is appealing to developers and customers alike. Software assurance aims to provide *justifiable confidence* that software is trusted to behave as intended

even amidst intentional and unintentional attacks (Goertzel et al., 2007; Sinclair, 2005).

Based on experiences and lessons learned from designing a graduate level software assurance curriculum, assurance optimization is aided by implementing techniques in each phase of the SDLC. The intent of this paper is to share a strategy for integrating software assurance throughout the lifecycle in a methodical manner, proving a secure

DOI: 10.4018/978-1-4666-1580-9.ch002

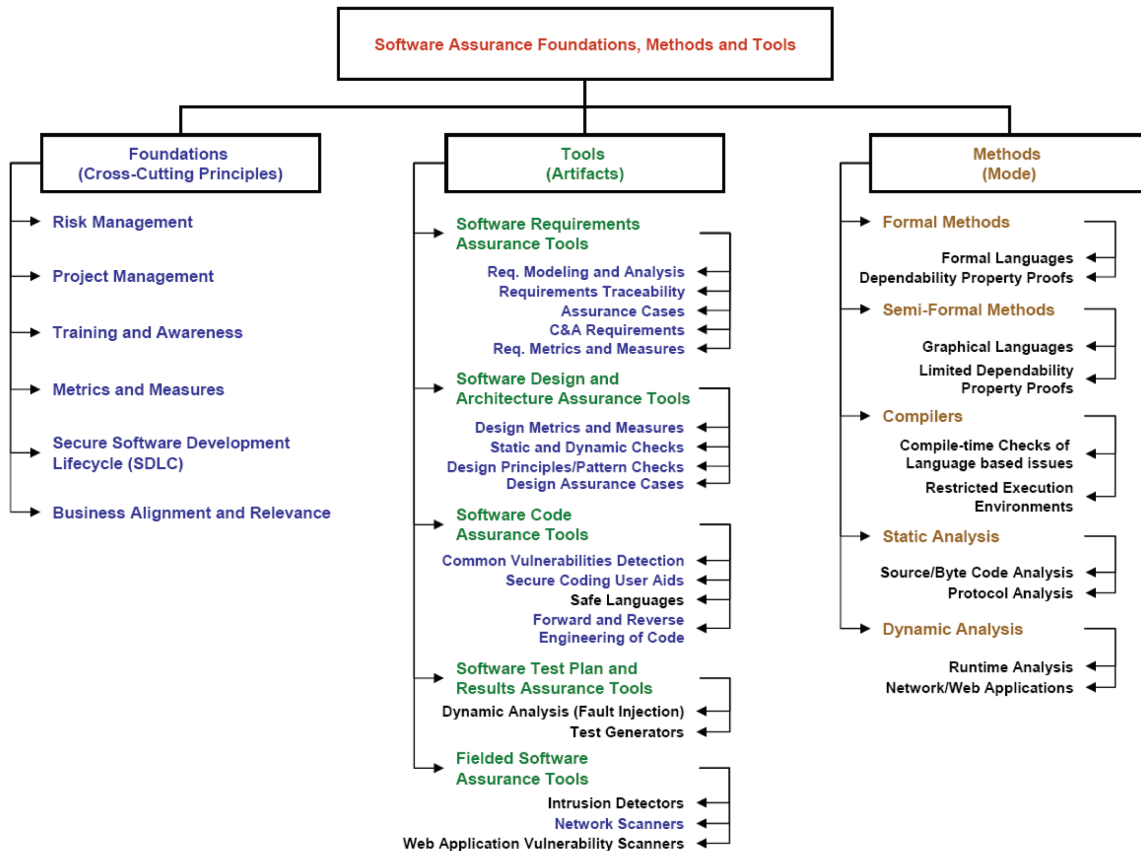
and trusted system. Several of the foundations, tools and methods used for optimization, shown on Figure 1, will be highlighted throughout the context.

## BACKGROUND

Software is the core component of modern products and services, supporting business operations for all sectors of life. With each software use, there are factors which contribute to increased mission risk including: project size and complexity, attack sophistication, and use of third-party vendors (Ellison, 2006; McGraw, 2005). Dependence on this software makes security a primary concern (Allen et al., 2010). Software Assurance is achieved by

understanding the mechanics of software built and/or acquired and incorporating validation tools and strategies into each phase of its lifecycle to build a trusted and secure product. Figure 2 diagrams this process, showing a step-wise approach for infusing assurance techniques into the SDLC by outlining approaches and artifacts produced. Knowledge gained from performing each step in a methodical and well-defined manner is carried forward, resulting in progressive learning. This is an iterative process, as education acquired from one phase will allow for more intelligent review in another. Assurance optimization can be achieved by mitigating common weaknesses in software throughout the aforementioned process. Peter G. Neumann identified nine sources of problems in computer systems (1994). A framework for

Figure 1. Software assurance foundations, methods and tools



17 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

[www.igi-global.com/chapter/assimilating-optimizing-software-assurance-sdlc/65840](http://www.igi-global.com/chapter/assimilating-optimizing-software-assurance-sdlc/65840)

## Related Content

---

### Building Sustainable Smart Cities: Integrating Cloud Technology and Intelligent Parking Systems

Monika Sharma, Manju Sharma and Neerav Sharma (2024). *The Convergence of Self-Sustaining Systems With AI and IoT* (pp. 104-129).

[www.irma-international.org/chapter/building-sustainable-smart-cities/345508](http://www.irma-international.org/chapter/building-sustainable-smart-cities/345508)

### Towards Construction of Business Components: An Approach to Development of Web-Based Application Systems

Dentcho N. Batanov and Somjit Arch-int (2003). *Practicing Software Engineering in the 21st Century* (pp. 178-194).

[www.irma-international.org/chapter/towards-construction-business-components/28118](http://www.irma-international.org/chapter/towards-construction-business-components/28118)

### Improved Fall Detection Model on GRU Using PoseNet

Hee-Yong Kang, Yoon-Kyu Kang and Jongbae Kim (2022). *International Journal of Software Innovation* (pp. 1-11).

[www.irma-international.org/article/improved-fall-detection-model-on-gru-using-posenet/289600](http://www.irma-international.org/article/improved-fall-detection-model-on-gru-using-posenet/289600)

### Object-Aware Business Processes: Fundamental Requirements and their Support in Existing Approaches

Vera Künzle, Barbara Weber and Manfred Reichert (2011). *International Journal of Information System Modeling and Design* (pp. 19-46).

[www.irma-international.org/article/object-aware-business-processes/53204](http://www.irma-international.org/article/object-aware-business-processes/53204)

### Blockchain Implications and Utility for Higher Education

Neeta Baporikar (2024). *Frameworks for Blockchain Standards, Tools, Testbeds, and Platforms* (pp. 73-95).

[www.irma-international.org/chapter/blockchain-implications-and-utility-for-higher-education/337207](http://www.irma-international.org/chapter/blockchain-implications-and-utility-for-higher-education/337207)