

Chapter 13

ERP Software Inspections and Audits

Julius Murumba

Technical University of Kenya, Kenya

Jackson Kipchirchir Machii

Technical University of Kenya, Kenya

ABSTRACT

The role of software inspections, product reviews, walk-troughs, and audits in ERP software is analyzed in this chapter. Software inspections are a disciplined engineering practice for detecting and correcting defects in software artifacts with the aim of correcting them. Walkthroughs involve software peer review mechanism in which a programmer leads peers through a software product, in a process in which participants ask questions and make comments about possible errors, violation of development standards, and other problems. This chapter also discusses ERP systems audit and control risks and seeks to help understand key risks and control issues surrounding ERP systems.

INTRODUCTION

Enterprise resource planning software's standardize, streamline, and integrate business processes across all departments of an organization, for example finance, human resources, procurement and distribution. Enterprise Resource Planning software usually operates on an integrated software platform using common data definitions operating on a single database, and provides several separate but integrated modules that can be installed as a package for any organization (Bae &

DOI: 10.4018/978-1-5225-7678-5.ch013

Ashcroft, 2004; Wailgum & Perkins, 2018). ERP systems help organizations track information across all departments and business functions (Hawari & Heeks, 2010). ERP software reviews are done to find defects or errors and correct them before the software is released to customers. A defect is an instance of non-conformance with the initiating software requirements or standards. Software errors can be classified into a number of general categories. These include: (1) Faulty or erroneous definition of requirements and inclusion of unnecessary requirements or functions that may not be needed in the software's future. (2) Misunderstandings resulting from defective client and developer communication may lead to errors that undermine the development process (3) Deliberate deviations from software requirements that may occur when a developer reuses software module taken from previously done projects without sufficient analysis and understanding of the changes and adaptations needed to correctly fulfill all the new requirements. This may be due to time or financial limitations, that may lead to the developer deciding to omit some functions that are required in an attempt to cope with these pressures arising out of the limitations (4) Logical design errors (5) Coding errors (6) Non-compliance with documentation and coding instructions (7) Shortcomings of the testing process (8) Procedure errors (9) Documentation errors.

ERP SOFTWARE INSPECTIONS

Software inspection is a process carried out manually in the early stages of the software development cycle by a group of peers on code and software artifacts developed by other developers. Software inspection requires human effort and is deemed to be very cost-effective, since it leads to defects being removed before they enter the phases of implementation or maintenance. The regular application and success of inspections in the software industry have led to suggestions for its introduction to ERP software. Barhate (2012) states that software inspection was first invented by M. Fagan at IBM in 1976 and has since been improved and adapted by many major software companies. ERP software inspection is a technique that can be used to achieve quality control for the source code and for identifying associated process improvements. It is part of quality assurance in the software development process performed by people i.e. reviewers and complements automatic checking tools. ERP software inspections involve careful examination of its code and other software artefacts, checking them for characteristics that are known to be problematic from past experiences. It is a formal process involving labour intensive and manual analysis techniques that are applied to code documents in the early stages of the software development. Inspections are non-execution-based meaning that the inspector does not have to compile or execute the code during the examination; and

16 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/erp-software-inspections-and-audits/232361

Related Content

Developing Professional Competence in Project Management Using E-Simulation on Campus

Ian Searleand Hossein Zadeh (2013). *Enterprise Resource Planning: Concepts, Methodologies, Tools, and Applications* (pp. 541-556).

www.irma-international.org/chapter/developing-professional-competence-project-management/77238

Beyond ERP Systems as a Hype: Understanding ERP Systems as Distinct Technological, Organizational and Cognitive Phenomena

Eveline van Stijn (2002). *Enterprise Resource Planning: Solutions and Management* (pp. 243-254).

www.irma-international.org/chapter/beyond-erp-systems-hype/18458

Academics' ICT Capabilities in a New Educational Paradigm in Developing Countries: A Capability Approach

Agnes Chigonaand Rabelani Dagada (2013). *Enterprise Resource Planning Models for the Education Sector: Applications and Methodologies* (pp. 18-32).

www.irma-international.org/chapter/academics-ict-capabilities-new-educational/70258

Effective Use of External Expertise in Enterprise Systems: A New Zealand Case Experience

O. Maxie Burns (2002). *Enterprise Resource Planning: Global Opportunities and Challenges* (pp. 267-283).

www.irma-international.org/chapter/effective-use-external-expertise-enterprise/18475

A Case for Enterprise Interoperability in Healthcare IT: Personal Health Record Systems

Mustafa Yuksel, Asuman Dogac, Cebail Taskinand Anil Yalcinkaya (2014). *Revolutionizing Enterprise Interoperability through Scientific Foundations* (pp. 192-215).

www.irma-international.org/chapter/a-case-for-enterprise-interoperability-in-healthcare-it/101111