

A CASE Tool Evaluation and Selection Methodology

Aniruddha Guha Biswas, Indian Institute of Information Technology, Allahabad, Uttar Pradesh, India

Raveesh Tandon, Indian Institute of Information Technology, Allahabad, Uttar Pradesh, India

Anurika Vaish, Indian Institute of Information Technology, Allahabad, Uttar Pradesh, India

ABSTRACT

CASE Tools have been considered as the most advanced technique to derive quality products. The cost associated with CASE tools is generally high due to their steep learning curves and is a matter of concern for software engineering houses. This makes it imperative that the right set of CASE tools are evaluated and selected for a particular set of tasks. This research paper proposes a methodology for the purpose of comparison and selection of various CASE tools available in the market. The proposed methodology allows its users to easily identify and select the best CASE tool suited to their purpose. The paper proposes a matrix based model, mapped across parameters and their attributes that produces an optimised score card that supports the decision making process.

Keywords: CASE Tool Evaluation, CASE Tools, Computer Aided Software Engineering (CASE), Matrix Based Model, Software Engineering Process Management

INTRODUCTION

The Computer Aided Software Engineering is a paradigm that incorporates integrated tools, technology and a methodology that supports and supplements the software development life cycle. The basic objective of CASE is to provide a set of tools to the Software Development team to automate their efforts, bringing about enhanced efficiency and reduced time and efforts. Object oriented approaches have been continuously drawing importance in the area of CASE Tool evaluation research as discussed by

(Church & Matthews, 1995; Massacci, Mylopoulos, & Zannone, 2007; Post & Kagan, 2000, 2001). This is highly beneficial for developing appropriate structured design and carrying out knowledge enhancement.

Irrespective of whether it is the development of the demand model during the system analysis phase or the conceptual data model and the physical data model during the system design stage the ultimate goal of the CASE approach is to bring about extreme automation in software development. Some of the popular advantages of The CASE approach can be listed as:

DOI: 10.4018/jsita.2013040104

- High Speed software development
- Reducing software development cycle
- Lowering the development cost
- Improving the system operational efficiency, the system sustainability and maintainability.

As quoted by Siltanen Aila, Sorensen Carsten, and Tahvanainen Veli-Pekka (1992) Large organisations understand the importance of standards and benchmarks in selecting a right set of CASE Tool as it is correlated to their product performance and quality, however small and midsize organisations do not perform such tests due to time constrain. With the help of this research paper we can identify a methodology for selecting a CASE Tool which will benefit us with its inherent capabilities.

IMPLICATIONS OF THE CASE APPROACH

In their paper the authors (Li & Ye, 2011) categorized the traditional software development life cycle into 7 distinct phases:

- Question Definition (T1)
- Demand Analysis (T2)
- System Design (Preliminary Design and detailed design) (T3 + T4)
- Coding (T5)
- System Test (T6)
- System Maintenance (T7)

The variables T1-T7 denote the time the respective phase requires.

Then the total time T, required to successfully accomplish the Systems Development Life Cycle is: $T = T1 + T2 + T3 + T4 + T5 + T6 + T7$

We can build a scenario, where the use of CASE tools can directly transform or translate the concept data model during the outline design stage to the physical data model in the detailed design stage. The time it takes to undertake this transformation is negligible compared to the time it would take to follow the traditional design methodology. Thereby if we label this

transformation time M4, we can say that M4 is far less than T4.

Similar reductions in time can be observed when a CASE tool is used in the testing and the maintenance phase (Li & Ye, 2011). During the test phase, CASE tools can generate test data automatically without requiring any manual interventions and during the maintenance phase Reverse Engineering reduces the time of system maintenance significantly. The new times for the Testing and the Maintenance phases can be denoted by M6 and M7. Thus, due to the use of CASE tools, the new software development cycle time T_{New} is:

$$T_{New} = T1 + T2 + T3 + M4 + T5 + M6 + M7$$

$$T_{New} < T$$

$$T1 + T2 + T3 + M4 + T5 + M6 + M7 < T1 + T2 + T3 + T4 + T5 + T6 + T7$$

Remove the same on both sides:

$$M4 + M6 + M7 < T4 + T6 + T7$$

Thus, we can see that the use of CASE tools reduces the software development cycle significantly making the design and coding stages easy to implement.

LITERATURE REVIEW

In their paper the authors (Li & Ye, 2011) through the analysis of CASE Tools to the efficiency of software development have proven effectively that a high degree of automation is achieved in software development by using CASE Tools. They reduce the software development cycle, reduce costs of development and mitigate risk attached to the project, improve the system operational efficiency and the system maintainability and so forth.

Similarly, in their paper the authors (Radmila & Jasna, 1999) have discussed the problem of OO CASE tools as methodology companions that encourage or enforce methodology support. The paper presents a basis for establishing an evaluation approach that can successfully

11 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/article/a-case-tool-evaluation-and-selection-methodology/89357

Related Content

ACEnet: Facilitating Economic Development Through Small Business Electronic Commerce

Craig Van Slyke, France Belanger and Marcy Kittner (2006). *Cases on Strategic Information Systems* (pp. 336-357).

www.irma-international.org/chapter/acenet-facilitating-economic-development-through/6449/

IS Strategy at NZMilk

P. Cragg and B. McQueen (2006). *Cases on Strategic Information Systems* (pp. 272-284).

www.irma-international.org/chapter/strategy-nzmilk/6445/

Process and Structural Implications for IT-Enabled Outsourcing

Paul Drnevich, Thomas H. Brush and Gregory T. Luckock (2011). *International Journal of Strategic Information Technology and Applications* (pp. 30-43).

www.irma-international.org/article/process-structural-implications-enabled-outsourcing/60143/

Citizen Journalism: How Technology Transforms Journalism Business through Citizen-Reporters in Nigeria

Olubunmi P. Aborisade, Caroline Howard, Debra Beasley and Richard Livingood (2011). *International Journal of Strategic Information Technology and Applications* (pp. 1-11).

www.irma-international.org/article/citizen-journalism-technology-transforms-journalism/54706/

Dynamic Data Replication Based on Tasks scheduling for Cloud Computing Environment

Siham Koudri and Belabbas Yagoubi (2017). *International Journal of Strategic Information Technology and Applications* (pp. 40-51).

www.irma-international.org/article/dynamic-data-replication-based-on-tasks-scheduling-for-cloud-computing-environment/210602/