

Integrating Software Engineering and Costing Aspects within Project Management Tools

Roy Gelbard

Bar-Ilan University, Israel

Jeffrey Kantor

*Bar-Ilan University, Israel
University of Windsor, Canada*

Liran Edelist

Bar-Ilan University, Israel

INTRODUCTION

Currently, there is no integration among CASE tools (computer aided software engineering, also named AMD tools, analysis modeling and design), costing tools, and project management (PM) tools. Not only are there no integrated tools, but there is also no conceptual integration among software engineering (SE) aspects and accounting-costing aspects of software projects within PM tools. PM tools, as well as costing tools are used not only for tracking and controlling an ongoing software project, but also at the very beginning stages of the project, in which critical estimations concerning budget and time frame are made. In order to have a firm, robust, and accurate planning, project planning should be based directly upon raw SE components-objects, that is, upon analysis and design components-objects.

According to the Standish Group CHAOS Report 2003, each year in the USA there are approximately 175,000 projects in IT Application Development which spends \$250 billion. Among these, 31.1% of projects will be cancelled, 52.7% of projects will cost 189% of their original estimates, only 52% of required features and functions make it to the released product, and Time overruns 82%. In financial terms \$55 Billion dollars is wasted in these projects (Madpat, 2005).

Budget overrun indicates cost management problems, although this area is defined by the project management integration (PMI), as one of the nine core activities of projects management. Costing difficulties result from both implementation limitations of costing solutions in complex and changing requirements as well as the technological environment. Risk management is

also defined by the PMI as one of the nine core areas of project management; but there is also no integration between PM tools and SE tools in light of the need for risk management.

According to Maciaszek and Liong (2005), success of a software project depends on five software engineering areas that are related to each other: the development of the life cycle of the software, processes management, the model's configuration and language, and SE tools and project planning. The combining between formal tools of SE and PM processes in the different stages has been proved by research as holding a positive contribution to the efficacy of the project and as an improver of the adherence to costs, technical requirements, and the schedules that were allocated to the project (Barker & Verma, 2003).

This study proposes and prototypes a model that integrates these three aspects of software projects by automatically mapping SE objects and accounting-costing objects into PM objects. To validate the feasibility of the model and without loss of generality, it is demonstrated using former research platform focused on conversion of data flow diagrams (DFD), which are actually full enterprise set of use cases diagrams reflecting entire system-software project into Gantt charts.

BACKGROUND

CASE and PM Tools

CASE/AMD tools support the analysis, design, construction, and implementation stages of the information

system life cycle (ISLC) (Barker & Longman, 1992; Pendharkar, Subramanian, & Rodger, 2005; Somerville, 2004). Commercial tools, such as IBM-Rational XDE, are covering main stages of ISLC; the “Requisite-Pro” module, for instance, is designated to the stage of requirement definition, “Rose” module to the analysis and design stage, and “Test-Studio” module to the testing stage.

Although PM tools support management and control along the ISLC, there is hardly any integration between CASE tools and PM tools. Thus, ISLC modeling approaches, such as the functional approach (e.g., DFD, ERD, STD), as well the object-oriented approach (e.g., use cases, activity diagrams, STD), even when automated, are used mainly in the early analysis stage primarily for visual documentation. The “database of specifications,” laboriously elicited and gathered during the creation of modeling diagrams, is hardly ever applied again for project management purposes, even though this information is valuable for project managers who are involved in the construction and implementation stages. In fact, due to lack of integration along the ISLC, the specifications database is often either overlooked altogether or collected again as if their creation earlier never took place. Moreover, standard methods for system analysis and development usually make no reference to methods for project management. Accounting and costing parameters, which are reviewed at the next chapter, are not represented not at SE tools or at PM tools, and handled in totally separated systems.

One conclusion that emerges from a thorough review of software engineering and project management areas is that SE tools are much more heterogeneous than PM tools. Gantt and Pert charts have become dominant project management modeling tools (Fox & Spence, 1998; Hughes & Cotterell, 2002) and are currently included in standard PM software such as Microsoft Project, PS-Next, and others. A survey of 1,000 project managers has found that 48.4% use MS Project, 8.5% use MS Excel, and the rest use Gantt/Pert-based tools from other vendors. The average satisfaction from PM tools in this survey was 3.7 on a scale of 1 to 5. Another survey reveals that only 10% of 240 project managers do not use PM tools at all, down from 33% in 1996. Moreover, more than 50% use Gantt/Pert-based project management software to manage every project, independently of its application domain and characteristics.

In contrast, the following two commercial CASE software packages demonstrate the heterogeneity of tools in the area of software engineering. Oracle’s *Designer* supports functional hierarchy analysis based on Barker and Longman (1992) methodology, and IBM-Rational offers XDE-Rose, a modeling tool based on the unified modeling language (UML) only. PM tools thus seem more standardized and mature than CASE tools. This could be the reason why 71% of 397 software engineers surveyed in 20 European countries employ PM tools while only about 26% utilize CASE tools, despite similar levels of training (Domges & Pohl, 1998).

Although CASE tools, including those mentioned above, support teamwork, none contain elements that take into consideration teamwork planning, time planning, dependencies, resources allocation, cost estimation, or risk management. Moreover, none include Gantt or Pert models or offer built-in interfaces to PM tools. Methodologies and models for managing software projects have yet to make it from the idea to the product phase, despite persistent improvements in automated tools for requirement definition, systems modeling, and software engineering. The failure to transform project management theory to practice in the context of software development is especially troubling since more than 50% of such projects do not succeed (Madpat, 2005; Reel, 1999). In addition to the lack of integration between SE tools and PM tools managers in charge of software projects usually refrain from basing managerial judgement on data about requirements and functional characteristics of the specific development project (Reel, 1999).

With decades of systems development behind us, there is quite a consensus today with respect to the critical success factors (CSF) of system development projects and agile methodologies, there is still a need to introduce effective concepts, methods, measures, and tools for better control of software projects. All these observations lead one to conclude that assembling a repository of system requirements and system components, complete as it might be, does not guarantee effective planning of teamwork, scheduling of tasks, and controlling deviations between planned milestones and actual progress.

Against this background, the questions to consider are:

12 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/integrating-software-engineering-costing-aspects/13391

Related Content

Better Executive Information with the Dashboard Approach

Frédéric Adamand Jean-Charles Pomerol (2009). *Encyclopedia of Information Science and Technology, Second Edition* (pp. 335-340).

www.irma-international.org/chapter/better-executive-information-dashboard-approach/13595

Online Perception Model of College Students' Emotion Under the Fusion of Social Media Such as Weibo

Yinan Gu (2025). *Journal of Cases on Information Technology* (pp. 1-20).

www.irma-international.org/article/online-perception-model-of-college-students-emotion-under-the-fusion-of-social-media-such-as-weibo/372210

Improving Context Aware Recommendation Performance by Using Social Networks

Golshan Assadat Afzali Boroujeniand Seyed Alireza Hashemi Golpayegani (2015). *Journal of Information Technology Research* (pp. 73-87).

www.irma-international.org/article/improving-context-aware-recommendation-performance-by-using-social-networks/135920

Information Systems Research in China: An Empirical Study

Shaobo Ji, Qingfei Minand Weihe Han (2009). *Handbook of Research on Information Management and the Global Landscape* (pp. 258-272).

www.irma-international.org/chapter/information-systems-research-china/20624

Modeling the Development and Use of Strategic Information Systems

Francis D. Tuggleand H. Albert Napier (1994). *Information Resources Management Journal* (pp. 5-19).

www.irma-international.org/article/modeling-development-use-strategic-information/50998