PERGAMON

International Journal of Project Management 20 (2002) 461–468

INTERNATIONAL JOURNAL OF PROJECT MANAGEMENT

www.elsevier.com/locate/ijproman

Integrating system analysis and project management tools

Roy Gelbard[a], Nava Pliskin[b,*], Israel Spiegler[c]

[a]Department of Industrial Engineering, Tel-Aviv University, Israel
[b]Department of Industrial Engineering and Management, Ben-Gurion University, Israel
[c]Department of Information Systems, the Recanati Graduate School of Business Administration, Tel-Aviv University, Israel

Received 23 August 2000; received in revised form 28 November 2000; accepted 1 May 2001

Abstract

Currently, computer-aided tools for system analysis are distinct from project management tools. This study proposes and prototypes a model that integrates these two aspects of the Information System Life Cycle (ISLC) by automatically mapping system analysis objects into project management objects. To validate the feasibility of our model and without loss of generality, the conversion of Data Flow Diagrams (DFD) objects into Gantt and Pert diagrams is demonstrated in this study. Experiments with the prototype confirm that integrating common tools for system analysis and standard tools for project management, during system development, helps improve system building tasks and their management. In addition, project managers using the proposed mapping approach can better assess project duration and system performance parameters such as response time and data traffic. We address implications of our work to both academics and practitioners, discussing directions future research might take as well as opportunities and prospects for commercialization of the proposed approach.

Keywords: Software engineering; System analysis and design; Project management; Data Flow Diagram (DFD); Gantt diagram; Pert diagram; Information System Life Cycle (ISLC)

1. Introduction

Computer-Aided Software Engineering (CASE) tools support the analysis, design, construction, and implementation stages of the Information System Life Cycle (ISLC) [1,2,26]. Although Project Management (PM) tools support management and control of ISLC development tasks [4,5], there is hardly any integration to date between CASE and PM tools. Thus, ISLC modeling approaches, such as Data Flow Diagrams (DFD) or Unified Modeling Language (UML) [6], even when automated, are used 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 [7].

Given the current object orientation of both system analysis and project management methodologies, this paper contends that a model integrating these two aspects by automatically mapping system analysis objects into project management objects is not only desirable but also feasible. The need for such mapping emerges from the next section that contains a theoretical and practical review of DFDs, Functional Hierarchies (FH), Pert and Gantt diagrams, and the corresponding computer-aided tools. The focus of this study on dataflow, Gantt, and Pert diagrams are done, without loss of generality, for demonstration purposes of our model to validate the model and its feasibility.

Based on Section 2, it becomes clear that a major problem in system development is the gap between the analysis phase, which deals with the required *functionality* (i.e. the Requirements Engineering phase [8]), and the

* Corresponding author. Tel.: +972-8-647-2203; fax: +972-8-647-2958.

E-mail addresses: gelbard@post.tau.ac.il (R. Gelbard), pliskinn@bgumail.bgu.ac.il (N. Pliskin), spiegler@post.tau.ac.il (I. Spiegler).

0263-7863/02/$22.00
PII: S0263-7863(01)00044-8

construction phase, which deals with the actual program *coding*. Sections 3 and 4, respectively, describe and prototype a model that maps system analysis objects, based on DFDs or FHs, into project management and control objects, in the form of Gantt and Pert diagrams. On the basis of this mapping, it is also possible to run a Critical Path Method (CPM) analysis with respect to "DFD component chains" paths. Section 4 shows that bridging analysis and construction based on our approach is more objective than without our approach because raw DFD components are stored in the specifications database. Because of its automation, the proposed mapping can shorten project duration, contribute to efficient use of resources, and help estimate the required development effort and cost as well as the expected system performance, complexity, and quality. In the concluding section, we sum up the study and address the implications of our work to academics, in terms of directions for future research, and to practitioners, in terms of opportunities and prospects for commercialization.

2. Background

CASE tools are trying to automate the entire ISLC and commercial tools, such as Rational-Suites [www.rational.com/products], are covering main stages of ISLC. The "Rational Requisite-Pro" tool, for instance, is designated to the stage of requirement definition and analysis, "Rational Rose" to the analysis and design stage, "Rational Test-Studio" to the testing stage, and "Rational Clear-Case" to configuration management usage throughout the entire ISLC (including the production and maintenance stages). Since a complete review of ISLC tools is beyond the scope of this paper, the reader is referred to such studies as Barker [9], Pressman [2], Roberson [8], and Sommerville [10]. Similarly, this section does not attempt a complete review of the project management area, other than for the purpose of comparing the area of software engineering, and the reader is referred to Agnus [11], and Kerzner [12].

One conclusion that emerges from a thorough review of both areas is that tools for systems modeling and software engineering are much more heterogeneous than project management tools. According to Fox and Spence [13] and Pollack-Johnnson and Liberator [14], Gantt and Pert diagrams have become dominant project management modeling tools [15] and are currently included in standard PM software such as MS Project. A survey of 1000 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 [13]. The average satisfaction from PM tools in this survey was 3.7 on a scale of 1–5. Another survey reveals that only 10% of 240 project managers do not use PM tools at all, down from 33% in 1996 [14]. Moreover, more than 50% use Gantt/Pert-based project management software to manage every project, independent of its application domain and characteristics.

In contrast, the following three commercial CASE software packages demonstrate the heterogeneity of tools in the area of software engineering. PowerSoft (by Sybase) offers *Power-Designer*, that supports DFDs, for procedural system modeling, and Entity Relationship Diagrams (ERD), for database modeling. Oracle's *Designer 2000*, supports Functional Hierarchy Analysis and ERD [9]. Finally, Rational offers *Rose*, a UML modeling tool. Project management 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 [16].

Although the leading CASE tools mentioned earlier support teamwork, none contain elements for time planning and control that take into consideration teamwork, dependencies, and resources. Moreover, none include Gantt or Pert models or offer built-in interfaces to PM tools such as MS Project.

According to Reel [3], 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. In addition to the lack of PM tools for software development, Reel also observes that managers in charge of software projects usually refrain from basing managerial judgement on data about requirements and functional characteristics of the specific development project.

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 (Reel, [3]). Sawyer et al. [17] and Roberson and Roberson [8], among others, acknowledge that assembling a set of clear requirements, which is critical to success, is both expensive and time consuming. In a review of 18 non-automated tools for managing corporate IT, Ahituv et al. [18] classify the tools by their functionality: periodic training (7 tools), project feasibility analysis (12 tools), and project progress control (1 tool). They note, however, that in all but one of the approaches, utilization of CASE and PM tools is recommended. Boegh et al. [19] and Jiang and Klein [20] join many other authors in describing the need to introduce *effective* measures for better control of software projects. They mention, for example, that lines of code (LOC) or cost estimators are rather general and related to the specific