DECISION SUPPORT FOR THE ENTIRE PROJECT LIFECYCLE

James R. Burns, Texas Tech University
Balaji Janamanchi, Texas A&M International University
Pinyarat Sirisomboonsuk, University of Massachusetts Dartmouth

Abstract

The purpose of this paper is to provide architecture for a project management tool that subsumes existing project management tools and provides for integrated project initiation, planning, execution, monitoring and control, and closure. The paper is conceptual and addresses many issues with existing software support systems for project management. The paper substantiates a need for a support system for the entire project lifecycle. A number of computerized support and advisory tools are suggested as supplemental and integrative to existing software project management tools. Further studies and prototypes need to be developed that will enable support for the entire project lifecycle. This paper points to a lack of support from existing project management software systems for many of the major project management phases and provides insights into potential integrated use of project management tools for holistic, synergistic support of the entire project lifecycle.

Keywords: Project Management Software, Project Lifecycle, Project Management Body of Knowledge, Simulation, Expert Systems

INTRODUCTION

According to the Project Management Institute (2015), one-fifth of the global gross domestic product (GDP) or about US$ 12 trillion is derived from projects. Projects are the way professional work gets done, or so Tom Peters and Robert Waterman (1982) would suggest. Most firms today consider project management to be a core competence regardless of their size, industry sector, product line or distribution channel. This is especially so in the Information Technology (IT) community. Projects are the lifeblood of any IT consulting firm and certainly the implementations of enterprise systems and software are accomplished within projects. Enterprise systems and software initiatives are implemented as projects and this is irrespective of initiative type: IT development, installation, upgrade/enhancements, or some form of modification or maintenance. Project management is also critical both as a discipline and as a software module to the success of any enterprise resource planning (ERP) initiative. Further, project management modules are commonplace among the array of module types that comprise enterprise software. Many enterprise systems include a project management module that interoperates with the other modules that comprise the ERP system, utilizing human resource data, as well as other sources of data in its computations and outputs. Project management is, at once, one of the most important and most poorly understood areas of management (Sterman, 1992).

In what follows in this paper, we discuss the lifecycle of project management and what part of the lifecycle is supported with software. We describe what software assistance constructs can provide to the discipline of project management as well as to existing project management software and systems. Included in this discussion are wizards, knowledge bases and inferencing, simulation, flight-simulators, dashboards, quality gates, and expert systems to aid project managers in their judgment and decision making. Robust meta-project management software tools should be tied into the enterprise database so that corporate data are accessible. Specifically, human resource data (who is available when and at what cost) as well as project portfolio data and data on ongoing or past projects would be useful for effectively managing current projects. Such data would greatly assist with the art of estimating project task durations and costs, for example. The current state of the estimation art is greatly in need of probabilistic software support and this is described as well. While the various tools for project support may seem disjoint, there is a great need for integrated use of these tools for holistic, synergistic support of the entire project lifecycle.

RELATED REFERENCES AND WEBSITES

There is extensive literature in the form of books, articles, and websites on project management. We provide reference to the important literature here. First and foremost, The Project Management Institute[1] is a relevant source for understanding the project lifecycle, as well as the structure of knowledge about projects. Several hundreds of websites provide relevant information on projects. Many articles by Herroelen, De Reych, and Demeulemeester (1998), Herroelen and Leus (2001), Herroelen, Leus, and Demeulemeester (2002) describe the merits of critical chain scheduling as well as project management per se. Vanhoucke, Vereecke, and Gemmel (2005) discuss project scheduling through use of simulation, which is a corroboration of our simulation discussion as a useful planning/analysis tool that appears later in the paper. Trietsch (2005) provides an update on the continuing critical chain dialogue in which a holistic approach to use of Program Evaluation and Review Technique (PERT) and Critical Path Method (CPM) is provided. Leach (2005) has written extensively as an advocate on critical chain project scheduling. None of these works address the larger context in which all projects find themselves embedded and how projects get launched without any formal budget or resources when so much effort, initiative, time and cost is required to define and plan a project ahead of its launch.

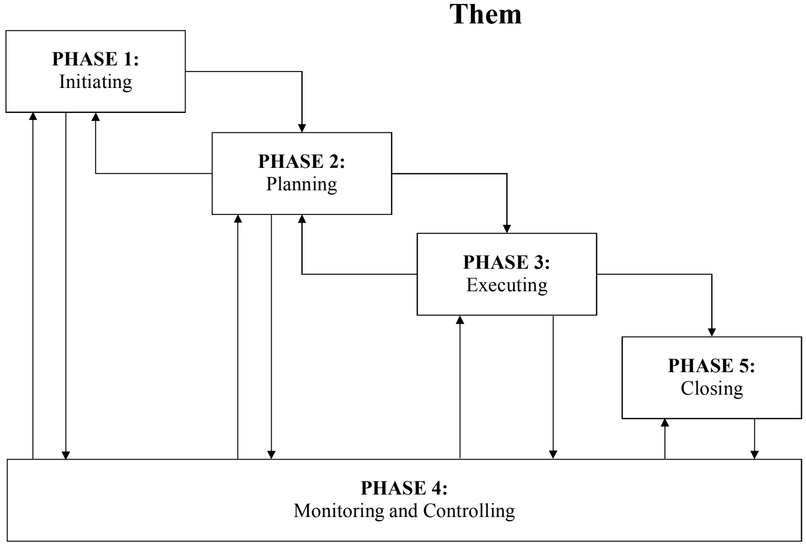
Moss and Atre (2003) discuss the entire project lifecycle for building decision support applications. This paper, in contrast, proposes innovations, improvements in the technology for initiating, planning, executing, controlling and closing projects. Dey (2006) discusses the use of a multiple-attribute decision making technique for project evaluation and selection. This is among the kinds of tools we have in mind for use here.

PROJECT LIFECYCLE AND ITS RELATIONSHIPS

The lifecycle of project management (as referred in *PMBOK Guide*[2], published by Project Management Institute, 2013) is perhaps the most important template invented in the last thirty years for completion of projects on-time and within-budget. Yet computerized project management tools do not support the lifecycle in its entirety. Specifically, traditional tools like Microsoft Project® do not support the first or last phases of the lifecycle. Microsoft Project® provides limited support for the second, and third phases of the lifecycle only. The absence of support for the first phase is particularly critical since the completion of the first phase of the lifecycle is absolutely necessary to successful completion of the project itself. The Initiating phase is one of the most critical in all of project management and is often considered the one most difficult to complete. According to William V. Leban (1996), program manager at Keller Graduate School of Management, lack of proper project definition and scope is a main reason why projects fail. Depicted in Figure 1 is an adaptation of the Project Management Institute's Lifecycle.

The project actually begins, in the sense that its tasks get executed, in the third phase (executing). The larger context, the whole lifecycle, is referred to here as the *meta-project*. Clearly, there are many tasks, processes, and steps that fall outside of the project itself—in the first, second, fourth, and fifth phases that have to be accounted for, estimated, managed, monitored, controlled, and supported with software. This paper addresses the larger project context which we have called the meta-project. By providing integrated software support for the entire lifecycle, it is possible to pass important information on from a former phase to the next phase. For example, if the requirements are specified by code and those codes are passed on to the planning phase then the software system may be able to pull up related plans and budgets (from former projects) to accomplish and fulfill that requirement. A greater degree of integration and automation leads to faster/cheaper completion of the first two phases, leading to faster/cheaper completion of the entire meta-project. Indeed, much of the manual work required to complete the last phase can also benefit from automation and assistance. In phase five, the following should take place: make sure all deliverables and obligations are fulfilled, hold a postmortem session, populate a history database of task durations and costs, and document lessons learned. It would be easy for software to produce a checklist of 'things' to be completed before final project

FIGURE 1
The Five Phases of the Project Lifecycle and the Relationships between Them



closure. The ability to automate these tasks results in a lower lifecycle cost, a faster lifecycle completion and better quality deliverables.

The focus of the project support system discussed herein is on reducing time-to-project (and product) completion, reducing project and product cost while maintaining or increasing the contribution to customer-perceived value, all with fewer defects.

THE NEED OF SUPPORT FOR THE VARIOUS PHASES

Starting a new project is an arduous task. It has been likened to the launch of a rocket. Expertise is needed to ensure a successful launch of a project. For example, only an expert knows how to assess requirements, needs, and from that build consensus among the stakeholders. Only an expert knows how detailed requirements should be taken in relation to the scope of the project. Requirements that are too detailed create additional costs, as well as difficulties during execution of the project's tasks. Requirements that are insufficiently detailed will not enable accurate estimates of project cost and duration to be determined in the planning and budgeting phase. Expertise is required to adequately perform a feasibility study, another important step in the first phase of the lifecycle of the project. Typically, an economic feasibility, a technical feasibility, and a general feasibility are performed. Out of these studies, a GO/NO-GO decision is made. Clearly, substantial expertise is again required.

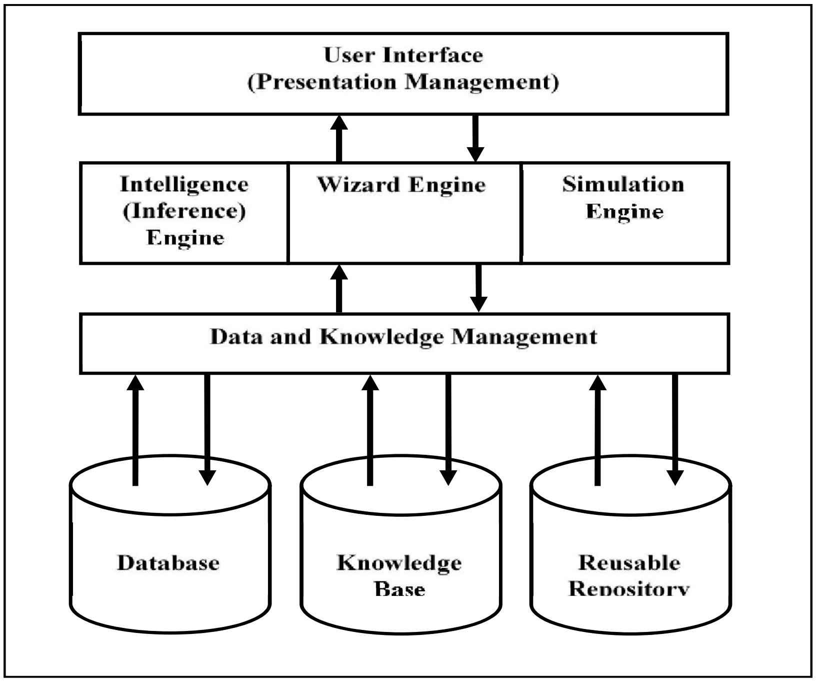
Expert systems capture and codify the knowledge of experts and regurgitate that knowledge back on demand to an inexperienced user. Two basic components of any expert system are its knowledge base and its intelligence (inference) engine that performs inferences on the knowledge contained in the knowledge base. Evidently, an expert system would be desirable in terms of assistance with the various phases of the lifecycle. A meta-requirement for a project decision support system would therefore entail a knowledge base and an intelligence engine that could perform inferencing. The expert system might be able to advise its users when code, designs, requirements, are already available and can be reused. Reuse is one of the most powerful concepts in software engineering for leveraging the investment in previously developed assets. Reusable components, modules are deposited in a repository that includes a 'dictionary' which defines the functionality, inputs, and outputs of each component/module in the repository. Developers are encouraged to reuse components placed in the repository rather than develop the same or similar components from scratch.

Thus, the expert system architecture needs to include a repository of reusable requirements. Portions of the expert system should be capable to discern which among hundreds of reusable requirements to bring to the attention of the users. Users could then avail themselves of this knowledge. Reusable requirements will stimulate users to think more creatively and systematically about what other teams have already created. Tied to the reusable requirements might be reusable work breakdown structures, reusable project plans, reusable designs, even reusable code. Reusability can greatly leverage past projects and is a great way to get projects and products out quickly and at minimal cost. In the software arena, reusable code has been shown, in some contexts, to reduce the time and cost of producing a product by an order of magnitude (Swanson, McComb, Smith, and McCubbrey, 1991). More will be said about reuse in a subsequent section.

PROJECT MANAGEMENT SUPPORT SYSTEM ARCHITECTURE

Figure 2 shows a suggested architecture for a project management support system. The project management support system draws upon all three resources, namely, available data from databases, current existing knowledge from the knowledge base, as well as all types of reusable components from the repository of reusable components. With the help of an inference engine that can draw inferences from these resources and a wizard engine that can lead users (project personnel) through arduous processes, intelligent support can be provided. Expert advice about best practices, options for reuse, as well as the various aspects needing careful consideration during the first and second phases of the project lifecycle are all possibilities. The simulation engine would provide additional insights into possible outcomes by performing policy-testing, sensitivity and optimization runs, results whereof would be stored in the knowledge base and databases, and therefore accessible to the wizard engine as well as the inference engine for consultation and advice. This architecture will

FIGURE 2
Project Management Support System Architecture



lead to the accumulation of a very rich, resourceful, and integrated amalgamation of data, knowledge and reusable components over a period of time. The architecture is consistent and compatible with the modern distributed architectures that comprise all commercial applications today. In what follows each of the components of the architecture are discussed, beginning with knowledge representation and inferencing.

Knowledge Representation and Inferencing

The architecture in Figure 2 provides for knowledge representation and inferencing. Burns, Winstead, and Haworth (1989) show how Boolean multiplication of relational Tables (in the form of binary matrices) leads to reachability connections and inferencing that is superior to conventional inferencing schemes. In this construct, knowledge is represented in the form of binary matrices. Some examples follow:

TABLE 1
A List of Project Management Processes and Where These Fit Within the Project Management Lifecycle

	PHASE 1: Initiating	PHASE 2: Planning	PHASE 3: Executing	PHASE 4: Monitoring and Controlling	PHASE 5: Closing
INTEGRATION MANAGEMENT	1	1	1	1	1
Develop Project Charter	1				
Develop Project Management Plan		1			
Direct and Manage Project Execution			1		
Monitor and Control Project Work				1	
Perform Integrated Change Control				1	
Close Project or Phase					1
SCOPE MANAGEMENT		1		1	
Plan Scope Management		1			
Collect Requirements		1			
Define Scope		1			
Create WBS		1			
Validate Scope				1	
Control Scope				1	
TIME MANAGEMENT		1		1	
Plan Schedule Management		1			
Define Activities		1			
Sequence Activities		1			
Estimate Activity Resources		1			
Estimate Activity Durations		1			
Develop Schedule		1			
Control Schedule				1	
COST MANAGEMENT		1		1	
Plan Cost Management		1			
Estimate Costs		1			
Determine Budget		1			
Control Costs				1	
QUALITY MANAGEMENT		1	1	1	
Plan Quality Management		1			
Perform Quality Assurance			1		
Perform Quality Control				1	
HUMAN RESOURCE MANAGEMENT		1	1		
Plan Human Resource Management		1			
Acquire Project Team			1		
Develop Project Team			1		
Manage Project Team			1		
COMMUNICATIONS MANAGEMENT		1	1	1	

	PHASE 1: Initiating	PHASE 2: Planning	PHASE 3: Executing	PHASE 4: Monitoring and Controlling	PHASE 5: Closing
Plan Communications Management		1			
Manage Communications			1		
Control Communications				1	
RISK MANAGEMENT		1		1	
Plan Risk Management		1			
Identify Risks		1			
Perform Qualitative Risk Analysis		1			
Perform Quantitative Risk Analysis		1			
Plan Risk Responses		1			
Control Risks				1	
PROCUREMENT MANAGEMENT		1	1	1	1
Plan Procurement Management		1			
Conduct Procurements			1		
Control Procurements				1	
Close Procurements					1
STAKEHOLDER MANAGEMENT	1	1	1	1	
Identify Stakeholders	1				
Plan Stakeholder Management		1			
Manage Stakeholder Engagement			1		
Control Stakeholder Engagement				1	

The content of the Table 1 is abstracted from the Project Management Institute's *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*, 5th Edition (2013). In the far left column (in all caps) is listed the ten knowledge areas that make up the *PMBOK Guide*. Below each knowledge area is listed the processes that comprise that knowledge area. For example, there are six processes identified within the INTEGRATION knowledge area—develop project charter, develop project management plan, direct and manage project execution, monitor and control project work, perform integrated change control, and close project or phase. If a process is used within a particular phase of the project such as *Initiating*, a "1" is placed in the respective row and column to indicate that the process is used in that phase. Thus, we see that the 'develop project charter' process is used within the *Initiating* phase. The 'develop project management plan' process is used within the *Planning* phase and so forth.

The Table 1 is a binary matrix, one that can be combined with other binary matrices through matrix multiplication to produce new information, inductions about relationships between processes. As an example, the process called 'develop project charter' will be fundamental in getting the project off on the right foot, since it impacts the first phase of the project lifecycle.

Table 2 shows a condensation of Table 1 in which all of the processes acknowledged by *PMBOK Guide* are removed.

TABLE 2
A Condensation of Material Presented in Table 1

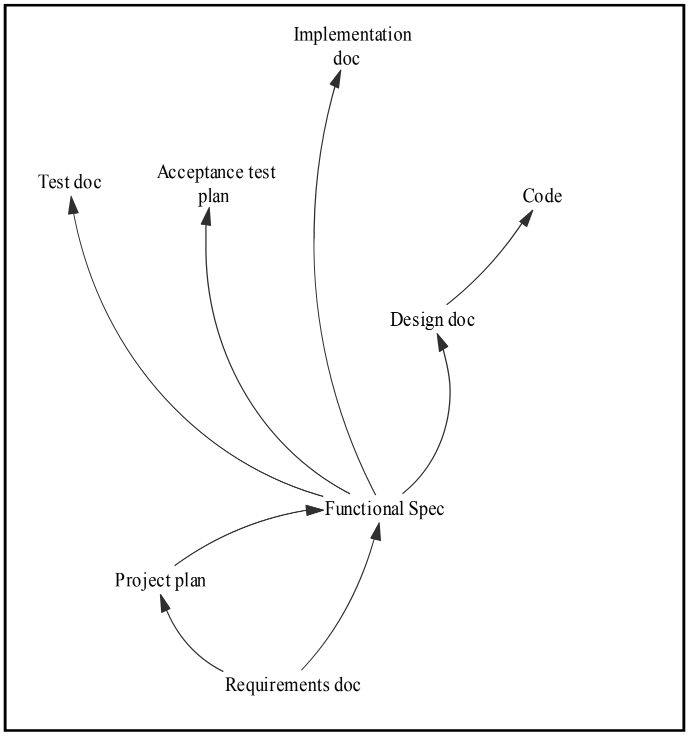
	PHASE 1: Initiating	PHASE 2: Planning	PHASE 3: Executing	PHASE 4: Monitoring and Controlling	PHASE 5: Closing
INTEGRATION	1	1	1	1	1
SCOPE		1		1	
TIME		1		1	
COST		1		1	
QUALITY		1	1	1	
HUMAN RESOURCE		1	1		
COMMUNICATIONS		1	1	1	
RISK		1		1	
PROCUREMENT		1	1	1	1
STAKEHOLDERS	1	1	1	1	

In both of the above Tables, the "1" indicates that the entity in the row on the left is used within the phase in the column heading above. In what follows we explore the relationships between the various project documents that are forthcoming in an information technology development project. Let:

A = Requirements document
B = Project plan and proposal
C = Functional specification
D = Design document
E = Code
F = Test document
G = Acceptance test plan
H = Implementation document

The relationships between these documents might be diagrammed as exhibited in Figure 3 wherein the edge or arrow represents the relation "has an influence upon."

FIGURE 3
Relationships between the Various Documents that Are Employed in the Lifecycle of an Information Technology Project Lifecycle



The information contained in Figure 3 can be represented internally within a computer as a binary matrix, as follows.

	A	B	C	D	E	F	G	H
A		1	1					
B			1					
C				1		1	1	1
D					1			
E								
F								
G								
H								

From either of these constructs, we learn that the requirements document is fundamental to all of the documentation that is created within a project and drives the content of all the other documentation and deliverables that are created in support of an information technology project. If the project support system finds a set of reusable requirements, then all subsequent documents should be reusable as well.

Each of the edges (arrows) in Figure 3 has a meaning. For instance, "B" is derived from and driven by "A", where "A" is the antecedent to the edge, and "B" is the subsequent node toward which the edge is directed. One implication of Figure 3 is that a reusable requirement can be linked to specific "elements" in the project plan, the functional specifications, the design document, the code, etc., resulting in an entire collection of reusable elements emanating from an identified reusable requirement.

If we take the Boolean product of the Table above indexed by A, B, C, D, E, F, G, and H with itself, we get the following Table (or matrix):

	A	B	C	D	E	F	G	H
A			1	1		1	1	1
B				1		1	1	1
C					1			
D								
E								
F								
G								
H								

The product of a binary matrix with itself produces yet another binary matrix. The numeral "1" in any cell implies the index pair is connected by a path of length two, exactly. For example, the "1" in row A and column C means there is an edge from A to C of length two, exactly. This is easy to verify from Figure 3 because we can see there is a path of length two leading from "requirements doc" to "functional spec". Thus, the matrix above provides a "1" everywhere there is a path of length exactly two. These would be paths consisting of exactly two edges. From this we learn that there is a path of length two leading from the requirements doc to the design doc, from the requirements doc to the test doc, from the requirements doc to the acceptance test plan, and to the implementation doc, as well as to the functional spec.

This type of inferencing has been shown to produce very fast, very robust results. For example, algorithms implementing this approach (Burns et al., 1989) can find all the ways a particular task or work package can be completed, all the processes by which a phase of the lifecycle can be fulfilled. Conventional backward or forward chaining inference engines can find only one way of achieving the desired result, at best. *PMBOK Guide* (Project Management Institute, 2013), for example, avows that each of its ten knowledge areas is comprised of three or more processes. The "best practices" associated with those processes are not discussed in *PMBOK Guide*, however. The project management intelligent support system proposed here would enable managers to quickly examine several likely practices for a given process and select the one that is "best" for them. The practices could be stored in a data or knowledge base or inferred from binary knowledge Tables of the type illustrated above.

An example query that could be addressed to such an expert system is "Give me all of the processes and best practices associated with the phase *Planning*." The tool would first find the processes associated with the *Planning* phase and then the best practices associated with them. These are listed below.

Processes Associated with PLANNING PHASE:

Develop Project Management Plan	Determine Budget
Plan Scope Management	Plan Quality Management
Collect Requirements	Plan Human Resource Management
Define Scope	Acquire Project Team
Create WBS	Develop Human Resource Plan
Plan Schedule Management	Plan Communications Management
Define Activities	Plan Risk Management
Sequence Activities	Identify Risks
Estimate Activity Resources	Perform Qualitative Risk Analysis
Estimate Activity Durations	Perform Quantitative Risk Analysis
Develop Schedule	Plan Risk Responses
Plan Cost Management	Plan Procurement Management
Estimate Costs	Plan Stakeholder Management

Best Practices Associated with the *PLANNING PHASE* (McConnell, 1996, p. 390):

Goal Setting	Rapid Development Languages
Joint Application Development	Requirements Scrubbing
Lifecycle Model Selection	Reuse
Measurement	Signing Up
Miniature Milestones	Top 10 Risks List
Outsourcing	User-Interface Prototyping
Principled Negotiation	Voluntary Overtime

Having a list as well as a description of a collection of best practices that could be implemented for specific processes within a particular phase would be of considerable benefit to project managers.

Wizards

Wizards are software tools that lead users step-by-step through a difficult process. Since there are so many varied tasks required of the project manager during the first and second phases of the lifecycle, an array of intelligent wizards would be most helpful. These wizards would recommend "best practices" to the project manager that would enable him or her to obtain consensus, cohesion and a commonly-held shared vision regarding what the product will consist of and how the project required to produce the product will unfold.

During the first two phases, in which the project runs on "auto-pilot," wizards along with an expert system would serve to guide the project manager through the sequence of steps required to complete each of the first two phases of the lifecycle. Both seasoned and new project managers can benefit from advice as well as a list of tasks they must do to complete the first two phases. These electronic 'assistants' (wizards, expert systems) would enable the project manager to assess likely project cost and duration to complete the first two phases. Currently, no such assistance exists. Such constructs would enable the project manager to design the overall project deliverable or product that is well-conceived.

Reuse

As has been explained already, reuse has proven its usefulness by aiding in project definition, planning, and execution in achieving immense cost and schedule reductions, as well as substantial gains in quality improvements, among software projects. An expert system that has access to a repository of reusable modules can assist project managers in several aspects of project initiation, planning, and execution. For example, to what extent is reuse of existing requirements, design elements, and code being made use of? Obviously, the first and foremost benefit of the reuse approach is the avoidance of, "wheel reinvention" in software development projects. Unless the past projects were executed with a view toward reusability, components from the past projects may not be readily reusable on an 'as-is' basis. Opportunities, however, may exist through inclusion of small changes. Commercial enterprise software works in such a fashion; generalized modules are tuned to the specific needs of a customer through the setting of switches. Expert systems, by directing the attention of project managers to such opportunities of reuse, albeit with some changes to the existing code or designs, if need be, accomplish optimal utilization of available resources. For example, reuse of 'triggers' and 'constraints' (in a relational database) from a past project may allow programmers to attach procedures to the database to ensure their execution in certain reusability situations (Swanson et al., 1991). In some 'traditional' application development contexts, Swanson et al. (1991) show that code reuse may cut costs and duration by 90% while increasing the overall quality of the software product. Rothenberger (2003) discusses software reuse factors in the context of software development.

Reuse need not be limited to just code reuse—it could also include the functional specification, the design documentation, the test documentation, the developers' code documentation, etc. Project plans can themselves be intelligently reused if properly selected and edited by an expert system.

Estimation

The discipline of estimation (determination of the time required and cost entailed) is in its infancy for information technology projects. A manual of process steps coupled with step durations and costs does not exist, yet such information is commonplace for construction projects. Because estimation is a knowledge- and expertise-driven process, technology can be brought to bear to innovate the process and improve the estimates. Information technology professionals have a notorious reputation for under-estimation of time requirements. At the same time, some project professionals tend to over-estimate the time it will take them to complete a task. A history database can be of substantial assistance. Project players are notoriously poor judges of the length of time it takes them to get things done and should be encouraged to regularly keep a log in which they estimate the length of time required to complete a task, and then measure and record the actual length of time required (Humphrey, 1997). The patterns of over- and under-estimation that appear should serve as a basis for learning and maturity, with the result that the estimator is able to improve the accuracy of his or her estimates.

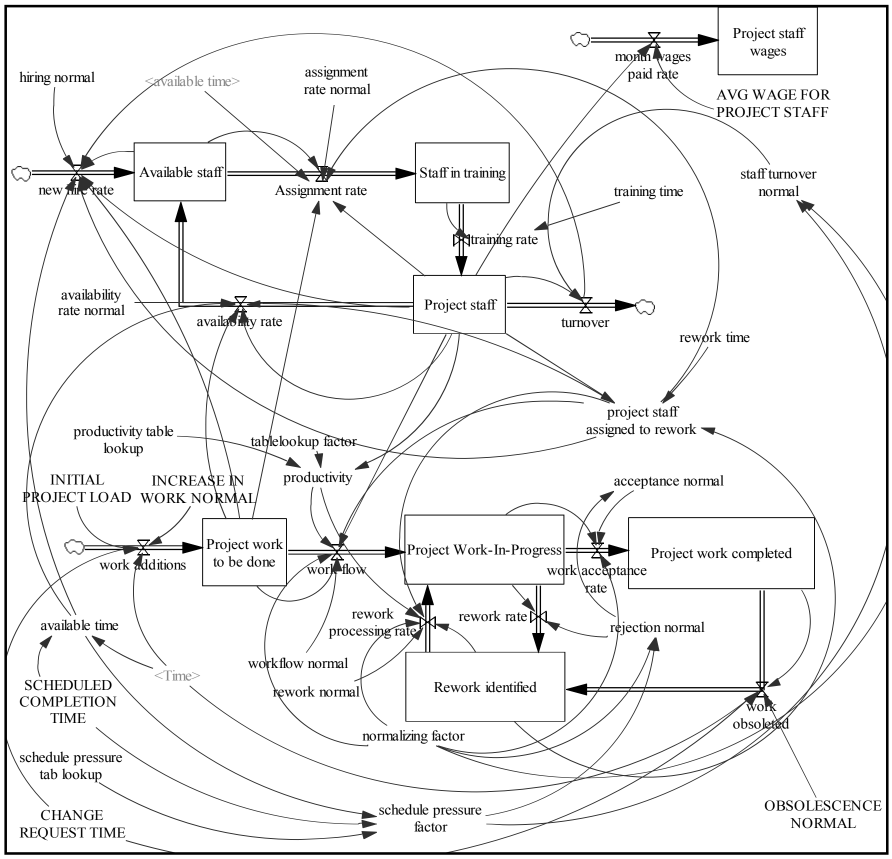
An expert system that interoperates with the history database could be of significant assistance to the discipline of estimation. Such a system might be able to discern when too much time has been allocated to an estimate, or conversely, not enough. The issue of estimation has been extensively addressed in the literature (see Herroelen and Leus, 2001). Currently, there is very little support for automated data collection of the actual length of time required to execute each step.

Simulation Tools

Simulation can be a very useful aide in planning a project because it will enable tradeoffs between cost and time to be studied. As more and more resources are added to a project, the simulation model will tell the user (the project manager) how much time can be saved. It is not a one-to-one relationship because, as Brooks (1975) and Fried (1994) point out, when more people are added to a project, the training, interaction, and overhead time goes up significantly cutting into the productivity of the existing knowledge workers.

An illustrative system dynamics model shown in Figure 4 considers the actual decline in productivity that occurs as a result of increases in project staff. This model consists of three sectors: a work sector, a people (or staff) sector, and a cost-accumulation sector. The work sector consists of the stocks Project-work-to-be-done, Project-work-in-progress, Project-work-completed and Rework identified. The staff sector consists of the stocks Available-staff, Staff-in-training, and Project-staff. The cost-accumulation sector contains the stock Project-staff-wages in which all wages paid to project staff are accumulated. The model also realistically represents situations in which the number of project staff is not static throughout the entire project, but varies, especially during ramp-up and scale down. Further discussion of this model can be found in Burns and Janamanchi (2006).

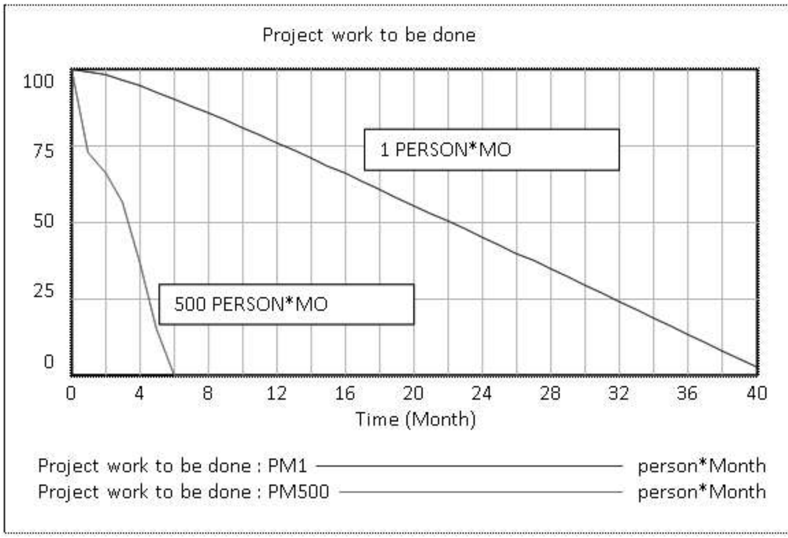
FIGURE 4
A System Dynamics Model of Project Dynamics, Implemented in Vensim[4]



The system dynamics model produces the following behavior for two distinct scenarios. First, a scenario in which there are very few staff and very slow hiring of staff is considered. Second, an opposite scenario is considered in which there is a huge abundance of staff and more on the way is considered.

From Figure 5, we see that when there were abundant resources (500 persons in each of the staff stocks, the project work completes in about six months. Whereas, if we have only one person in each of the staff stocks and a slow hiring rate, it takes 40 months to complete the same project.

FIGURE 5
Dynamics of Project Completion Time as Derived from the Structure Depicted in Figure 4

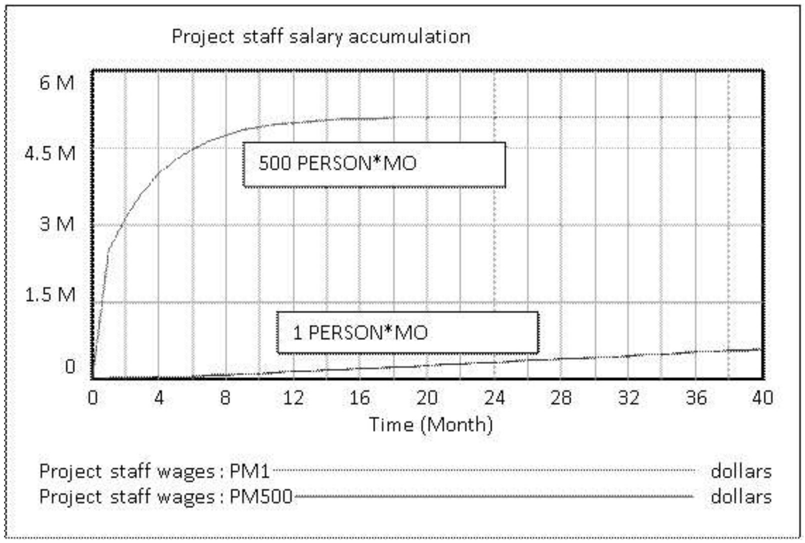


On the other hand, there is a stark contrast in terms of cost between the two scenarios, as shown in Figure 6. The first scenario is much less expensive, costing only a half million dollars while the second scenario costs roughly five million dollars, an order of magnitude more. The stock actually being plotted here is referred to as 'project staff wages' in the upper-right-hand corner of Figure 4. It is easy to see how such a simulation could serve as an excellent planning tool for studying various staffing levels and their impact on total project duration and total project cost. As previously mentioned, this model takes into consideration the decline in productivity per person resulting from increases in staffing levels. Also, this model allows for dynamic staffing in which the staffing level varies as a function of time.

Flight Simulators

Flight simulators are tools that project managers can use to exercise and improve their decision-making and judgment. Such tools would take an aspiring project manager through the entire project lifecycle in one sitting at a computer. The project manager would be able to see the effects of his decisions in faster-than real-time and, in the end, the simulation program would grade his performance and coach him as to how to improve. Such tools are practically nonexistent today, but very much needed to build the expertise required for managing projects in the 21st Century.

FIGURE 6
Cost Comparison Between the Two Scenarios: The Slow-as-a-Snail Scenario and The No-Holds-Barred Scenario



Improved judgment, especially in pressure situations, is a learned competency and like any competency, requires practice to become perfect. A flight simulator would enable project managers to try various strategies out on the model and observe the effects, thereby learning from the experience. Argyris and Schön (1978), Argyris, Putnam, and Smith (1985), Senge (1990) and many others are proponents of creating learning organizations that actually get better at what they do best as time moves on. An excellent treatise on managerial judgment is found in Bazerma (2001). According to Action Science advocate Argyris (1978, 1985), the best way to learn is by "playing" and that is exactly what flight simulators can do.

Dashboards

Dashboards are becoming increasingly popular as tools for quickly communicating how the project is proceeding and where there might be problems. Drill-down capabilities from the dashboard would be desirable. Easy-to-read and quick-to-grasp cost and schedule meters and dials as illustrated in Figure 7 can be used for executive decision support and for daily monitoring of the project progress. The variances are given with reference to the planned value (PV), earned value (EV), and actual costs (AC). Therefore, a negative value for schedule variance (SV) would indicate the project is running behind schedule. Similarly, the cost variance (CV) is computed as the difference between EV and AC incurred to date. Once again, a negative value of CV would indicate that budgeted costs are exceeded. The PV and EV indicators can also be shown as percentages, as indicated by the linear scales on the right in the dashboard above. The PV% indicator shows the planned value that should have been achieved by the current point of time in project execution. In comparison, the EV% indicator shows the percentage of total project value earned to date. Therefore, when the project is behind schedule, the EV% indicator will be lower than the PV% indicator, and vice versa.

FIGURE 7
Schedule Variance, Cost Variance, Earned Value and Planned Value Meters for Executive Decision Support

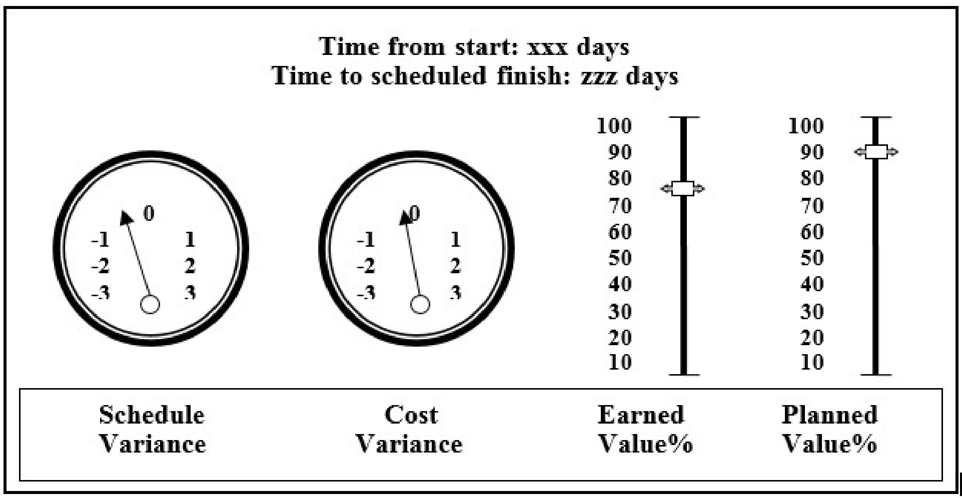
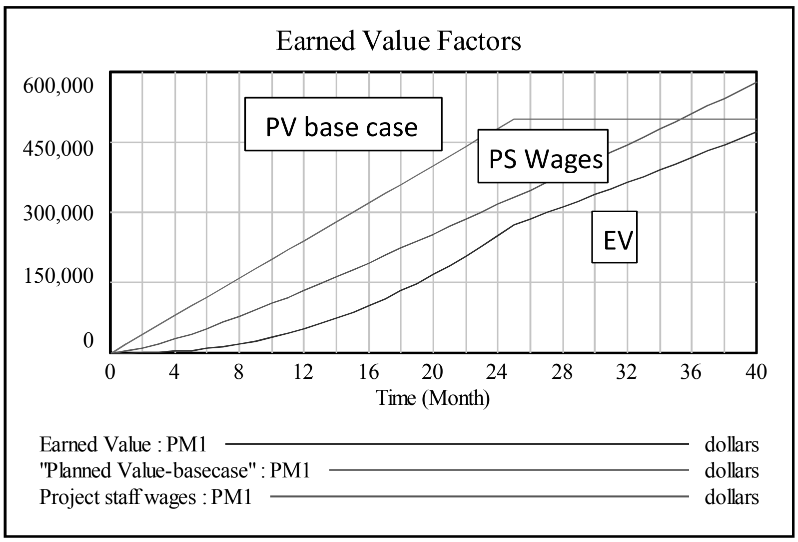


Figure 8 exhibits a different kind of dashboard in which earned value, planned value and costs are plotted against time on the horizontal axis. The project shown in Figure 8 below is behind schedule because the earned value lags the planned value and over budget because the actual costs (project staff wages) exceed the earned value. Clearly, the project is taking longer than 40 months when it should have been done in 25 months. Assume the project has been ongoing for 20 months, the plot predicts a 40 month completion time and a total expenditure of $600,000 as opposed to the original budget at completion of $500,000.

Quality Gates
A project management support system of the type we are envisioning can add structure and discipline to the entire project lifecycle. For example, the software may not allow its users to proceed to the next phase of the lifecycle until certain requirements are met. A checklist of items must be finished to its satisfaction before the project is allowed to proceed. For example, before the meta-project can proceed with *Planning*, the second phase, it must have endorsements (signoffs) from a variety of users that all planning processes are satisfactory, that

a risk assessment has been performed, that a feasibility study is complete, that a project charter is approved, that there is consensus among stakeholders regarding the nature of the ultimate product the project is to produce, etc. The initial delays arising from such a gate will result in significant time and cost savings accruing later in the lifecycle as a result. Quality gates (phase exits) may appear at the end of every phase and occasionally even more often. Occasionally, they can result in the cancellation of a project.

FIGURE 8
Plots of Earned Value, Planned Value and Actual Costs over Time for the Slow-as-a-Snail Scenario, as would be Exhibited on a Timeline Dashboard



Other Contributions of Expert Systems

The capacity to codify the expertise of the most experienced project management professionals into a knowledge base can be of great value in the management of projects by less proficient project managers. Expert systems would be able to suggest ways projects could be shortened in duration and reduced in cost. Examples include: when crashing makes sense, when overtime or outsourcing would be helpful, when to reduce intermediate deliverable batch sizes so that fast-tracking can be employed. These are just some of the time and cost-saving recommendations possible. The expert system, with its understanding of current resource usage, would be able to suggest ways to get better utilization of the resources, possibly through doing tasks in parallel, or by adding resources to critical tasks early-on in the project. The expert system might suggest fast-tracking when appropriate, crashing when appropriate, subcontracting when appropriate. With its knowledge of the design and architecture of the proposed product, it may be able to suggest improved designs that remove cost and time required to bring the product to completion. Time-batching, requirements scrubbing, quality-at-the-source, fast tracking, critical-chain scheduling are all ways the expert system might be able to prescribe time-reducing, cost-saving strategies to management of the project. Suggestions and advice with regard to leanness will further serve these interests of lower cost and less time. The expert system may be able to discern where safety and multitasking exist in the schedule and how these might be removed[5]. Expert systems could advise in regard to accommodation of changes, assisting with change management. In order to increase focus, an expert system would certainly be able to tell from the human resource data who is having their time divided between too many tasks and should, instead, become more focused.

SUMMARY AND CONCLUSION

This paper is about the science of decision-making in a complex and chaotic context-projects. As mentioned at the outset, roughly one fifth of the world's GDP is derived from projects. Project management, therefore, is a core competence for most firms. Yet it is something that project players do poorly in part because of a lack of computerized expertise and support available. This is particularly true for information technology projects; they are notoriously over-budget and behind-schedule. This paper addresses these issues with a support system for the entire lifecycle. A number of computerized support and advisory tools are suggested as a means for compensating for the lack of human expertise in project management. Moreover, computerized systems such as automated data collection would be helpful to populate a history database of the duration and cost of tasks and a lessons-learned knowledge base. These knowledge/databases can be used in the second phase of the lifecycle in the future projects.

Computerized assistance in the form of expert systems, wizards, simulation, and other similar tools and techniques can be of considerable value to the project manager when integrated with existing project management tools like Microsoft Project®. For example, a simulation tool will inform project managers what manpower/cost/duration tradeoffs are possible. Other simulation tools might enable trade studies to be performed as a part of requirements determination and definition of the project deliverable.

ENDNOTES

1. The Project Management Institute's website can be found at www.pmi.org.
2. PMBOK or PMBOK Guide stands for A Guide to the Project Management Body of Knowledge.
3. More about Concerto Critical Chain Planning® can be learned at www.realization.com.
4. Vensim is a trademark of Ventana Systems, Inc. More information, including free downloads, are available at www.vensim.com

5. Goldratt (Critical Chain, 1997) suggests removing safety from all tasks on the critical path and placing it at the end.

REFERENCES

Argyris, C., and Schön, D.A. 1978. Organizational Learning: A Theory of Action Perspective, Addison Wesley, Reading, MA.

Argyris, C., Putnam, R., and Smith, D.M. 1985. Action Science: Concepts, Methods, and Skills for Research and Intervention, Jossey-Bass, San Francisco, CA.

Bazerman, M.H., and Moore, D.A. 2001. Judgment in Managerial Decision Making, 5th edition, John Wiley & Sons, New York, NY.

Brooks, F.P. Jr. 1975. The Mythical Man-Month: Essays on Software Engineering, Addison-Wesley Publishing Co, Reading, MA.

Burns, J.R., and Janamanchi, B. 2006. Dynamics of Change Management in a Technology Project Context. International Journal of Information Systems and Change Management (1:2), 115-137.

Burns, J.R., Winstead, W.H., and Haworth, D.A. 1989. Semantic Nets as Paradigms for Both Causal and Judgmental Knowledge Representation. IEEE Transactions on Systems, Man and Cybernetics (19:1), 58-67.

Dey, P.K. 2006. Integrated Project Evaluation and Selection Using Multiple-Attribute Decision-Making Technique. *International Journal of Production Economics* (103:1), 90-103.

Fried, L. 1994. Managing Information Technology in Turbulent Times, John Wiley & Sons, New York, NY.

Goldratt, E.M. 1997. Critical Chain: A Business Novel, North River Press, Boston, MA.

Herroelen, W., De Reyck, B., and Demeulemeester, E. 1998. Resource-Constrained Project Scheduling: A Survey of Recent Developments. Computers and Operations Research (25:4), 279-302.

Herroelen, W., and Leus, R. 2001. “On the Merits and Pitfalls of Critical Chain Scheduling. *Journal of Operations Management* (19:5), 559-577.

Herroelen, W., Leus, R., and Demeulemeester, E. 2002. Critical Chain Project Scheduling: Do Not Oversimplify. *Project Management Journal* (33:4), 48-60.

Humphrey, W.S. 1997. Introduction to the Personal Software Process, Addison Wesley, Reading, MA.

Leach, L.P. 2005. Critical Chain Project Management, 2nd edition, Artech House, Inc, Norwood, MA.

Leban, W.V. 1996. Keller Graduate School of Management.

McConnell, S. 1996. Rapid Development: Taming Wild Software Schedules, Microsoft Press, Redmond, WA.

Moss, L.T., and Atre, S. 2003. Business Intelligence Roadmap: The Complete Project Lifecycle for Decision Support Applications, Pearson Education, Boston, MA.

Peters, T.J., and Waterman, R.H. Jr. 1982. In Search of Excellence: Lessons from America's Best-Run Companies, Harper and Row, New York, NY.
Project Management Institute. 2013. A Guide to the Project Management Body of Knowledge (PMBOK® Guide), 5th edition, Project Management Institute, Upper Darby, PA.
Project Management Institute. 2015. What are PMI Certifications? Project Management Institute, Inc. (http://www.pmi.org/Certification/What-are-PMI-Certifications.aspx; accessed December 25, 2015).
Rothenberger, M.A. 2003. Project-Level Reuse Factors: Drivers for Variation within Software Development Environments. Decision Sciences (34:1), 83-106.
Senge, P.M. 1990. The Fifth Discipline: The Art and Practice of the Learning Organization, Doubleday, New York, NY.
Sterman, J.D. 1992. System Dynamics Modeling for Project Management, Massachusetts Institute of Technology, Cambridge, MA.
Swanson, K., McComb, D., Smith, J., and McCubbrey, D. 1991. The Application Software Factory: Applying Total Quality Techniques to Systems Development. MIS Quarterly (15:4), 567-579.
Trietsch, D. 2005. Why a Critical Path By Any Other Name Would Small Less Sweet? Towards a Holistic Approach to PERT/CPM. *Project Management Journal* (36:1), 27-36.
Vanhoucke, M., Vereecke, A., and Gemmel, P. 2005. The Project Scheduling Game (PSG): Simulating Time/Cost Trade-Offs in Projects. *Project Management Journal* (36:1), 51-59.

ABOUT THE AUTHORS

James R. Burns is Professor of Information Technology and Operations Management in the Rawls College of Business Administration at Texas Tech University. He received his MS and PhD from Purdue University. He regularly teaches courses in project management and operations management. His research is focused on new developments in project/process management using concepts borrowed from maturity and learning. He is the author/co-author of one hundred journal papers and publications on subjects as diverse as knowledge management to design of decision support systems. He is also the author/coauthor of four textbooks on subjects ranging from information technology to project management.

Balaji Janamanchi is currently an Associate Professor of Management, Division of International Business and Technology Studies of the A.R. Sanchez Jr. School of Business Administration at Texas A&M International University, Laredo, Texas where he is employed since fall 2006. He received Ph.D. and MS degrees from the Rawls College of Business Administration, Texas Tech University majoring in Production and Operations Management (POM) and Management Information Systems (MIS), respectively. He worked as an MIS instructor in the

Rawls COBA, TTU (2001–2004). His research interests include supply chain management, project management, online teaching, and e-commerce related issues.

Pinyarat Sirisomboonsuk is a full-time lecturer at the Charlton College of Business, University of Massachusetts Dartmouth. Her research interests are in the area of operations and project management capabilities. Her work has appeared in academic and professional journals and conference proceedings, such as Decision Line, Southwest Decision Sciences Conference Proceedings and elsewhere. She received her Ph.D. in Operations Management from the Rawls College of Business, Texas Tech University. She earned her MBA from Oregon State University and a bachelor's degree in economics (with first class honors) from Chulalongkorn University, Thailand. She earned her Project Management Professional (PMP) credential in 2014.