Optimized Project Management of Systems and Software Projects

Denton Tarbet, Galorath Incorporated

Abstract. Successful software projects demonstrate good project management methods incorporating modern processes and effective utilization of metrics. This paper extends the application of control theory methods to project management decisions to include optimization of systems trade-offs to improve project performance. With the application of feedback control theory to the management decision process, future project control decisions result in changes that improve performance of the project process.

Issue

Systems and software projects traditionally experienced some degree of project failure. In this context, failure is defined as a project that demonstrated a failure to match (within a reasonable tolerance) the expected outcome. Studies such as the project data collected by the Standish Group in its annual Chaos reports [1] on software projects indicate improvement in software project success but the failure to meet project objectives is still high. Depending on the data referenced, there is a 50% to 80% probability that a systems and software project will:

1. **Require significantly more time than planned.**
2. **Cost significantly more than budgeted.**
3. **Deliver significantly less functionality than originally expected.**

In addition, project problems seem to demonstrate a direct correlation between the size of the project and the degree of project failure. That is, the larger the project, the longer the planned duration, and the greater the likelihood that the project will encounter at least one of the conditions. If we define success as achieving or exceeding expectations; i.e., success occurs when the actual outcome matches (within a reasonable tolerance) the expected outcome [2]. It follows that project success implies the need for a roadmap or plan that ensures meeting objectives within a reasonable confidence level. Software project performance can be improved with the application of effective management control during execution [3]. This paper proposes the application of classical control theory overlaid with optimum process control to establish methods to ensure the project maximizes potential of success.

System and Software Project Management and Control

Project management has been defined as, "A discipline that employs skills and knowledge to achieve project goals through various project activities." It involves planning, organizing, leading, and controlling costs, time, risks, project scope, and quality [4]. Systems and software project management in the context of this paper fulfills that definition. It is critical for success that we not only have a plan to implement, but also a way to lead and control to ensure a "best" project result.

Software projects can be considered as a system process with project management which:

1. **Follows a detailed plan.**
2. **Can be evaluated in process with process metrics.**
3. **Includes feedback to project management.**
4. **Incorporates feedback response into control decision process.**
5. **Applies process control "during the process" to reduce the risk of not meeting project objectives. That is to "optimize the project performance" and maximize probability of project success.**

The methods suggested to provide project management are aligned with classical control systems theory. Control systems theory provides a rigorous framework for analyzing complex feedback systems. We are all familiar with real-time control theory such as when driving a car. The driver constantly monitors the car's position in the lane. The process of monitoring the actual position against desired position and making steering adjustments is similar to tracking and controlling a software project. Similar mathematics apply, so that optimal control processes and modeling of classical control theory can be applied to systems and software project management [5].

Control Theory Applied to Systems and Software Project Management

Control systems in the broadest sense are critical in a wide range of applications. In manufacturing processes as simple as machining parts, the machines are set up, the manufacturing process is started, and output parts are regularly measured to ensure meeting specifications. Any changes of the manufactured parts away from specifications will be noted by the metrics being collected. As changes are noted, corrective action is taken to ensure the manufactured parts remain within specifications. Control theory is normally based on the principle of feedback:

1. **A desired objective is defined for the system.**
2. **Resources are applied, raw material is processed.**
3. **Sensors are used to collect metrics.**
4. **Metrics are analyzed to determine if the system is meeting its objectives.**
5. **Math models of the "real world" system are typically used to choose corrective actions from a set of possible actions.**
6. **Corrections are made to the input in order to improve the performance of the system with respect to the "desired objective."**

When control theory is applied to a dynamic system, it is critical to include a method to evaluate alternatives in the process of defining the "best" corrective actions. System dynamics provides the framework for modeling the systems and software process [5]. Within the overall system model, a critical decision tool is the method utilized to evaluate alternative actions. The application of control theory requires:

1. A defined system objective, for optimization the objective is referred to as a "cost function."
2. A control law—i.e. the initial software development plan.
3. Measurement of effectiveness accomplished provided by the set of software metrics collected.
4. A method to evaluate alternative corrective actions.

For the control process to be an "optimal control" process there must be a well defined "cost function." The cost function provides the basis for decisions reflected into the application of the "control corrections." The cost function or project goal presents a critical initial decision for the project. The goal can be to optimize, or minimize the difference between project cost and budgeted cost while monitoring schedule. The goal could be to minimize the difference between the actual project schedule and the planned schedule while monitoring cost. Applying classical control theory to the project management of systems and software projects can:

1. Optimize performance for cost within an acceptable schedule.
2. Optimize performance for schedule within an acceptable cost.
3. Optimize performance for functional capability with acceptable cost and schedule.

It will be a program management issue to define the objectives and resolve any conflicting goals for the project.

Systems and Software Projects

Considering the systems and software project, for a typical project with a defined beginning and end to develop a product a detailed project plan is developed. Project management assigns resources and initiates the project. Past projects have often demonstrated a process reflective of Figure 1.

The process runs according to the plan but when disturbance forces impact the process, there is no well-defined method to react to the impacts and correct the controller functions to improve performance. The goal of applying control theory is to implement the process as in Figure 2.

The process of Figure 2 includes the action to determine corrective actions to insert into the controller. For software process, a parametric effort and risk estimation tool, such as SEER-SEM [6], provides an effective method to identify and evaluate alternative actions that can be applied to improve project performance against the objective.

Project management implements methods to regularly review the process metrics and identify the requirement to implement corrective actions. Experienced project managers identify alternative actions to improve project performance. However, without

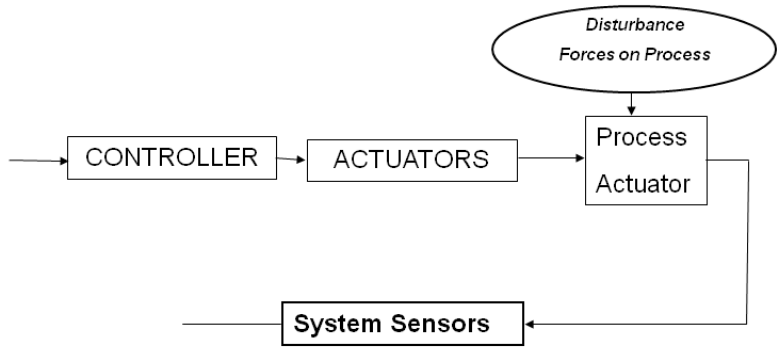

Figure 1

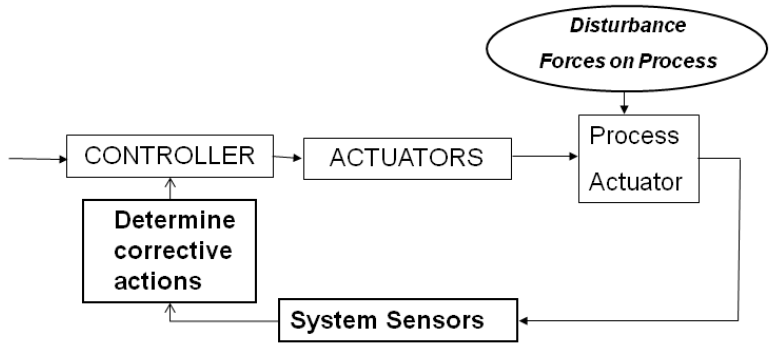

Figure 2

an effective tool such as a parametric software process model, it is difficult to identify reasonable project corrective actions that yield the best ROI. Occasionally, program management will suggest adding people to a project that is behind in schedule, or proposing that the team "work harder" which normally means extended overtime. However, it is easy to demonstrate the realities of Brook's law (i.e. adding people to a late project normally makes it later) with a parametric model that is set to minimum time estimation. The goal for project management should be to identify potential corrective actions that can be implemented by the project management and not as a function of asking the team to "work harder." Actions that can be implemented by project management that will affect productivity on the project include:

1. Reducing the requirements changes that impact the program.
2. Reducing the level of documentation required for the delivered product.
3. Updating software development systems to reduce the rehost from development system to target system.
4. Providing effective total software development tool sets (assuming there is sufficient time to incorporate the new tools without impacting the project).

A critical requirement for project management is the collection of a regular basis of effective project metrics. The measurement on the process is not of value if it is implemented as a "check-the-

box" process rolled out to satisfy a scheduled review or process improvement assessment. The metrics should be collected to support the "feedback control" of the project. Measurement must provide real information to support critical project and organizational business and technical decisions.

Applying classical control theory implies a feedback control loop that necessitates continual collection of metrics reflecting performance of the project development process. In the situation of optimally controlling a software development process the feedback control system can effectively incorporate the parametric model that should provide a best possible representation of the real world. The model will provide a basis to evaluate alternative control applications.

Applying the control theory construct to a software project would result in a process that can be represented by Figure 3.

In this process the system requirements that provide the "desire" are utilized initially to develop an estimate of effort, schedule, and risk. The project plan (schedule with allocated resources) is developed and implemented. Project direction is the responsibility of the project manager and the software development process is initiated. At regular intervals, process metrics are collected and provided as performance feedback into the estimation process. An effective parametric model which is being maintained, i.e. updated with the metrics data, can be used to evaluate the cost to complete (both in effort and schedule) and provide that input to the project manager for decisions with respect to implementing corrective action. With a decision to implement corrective action, the estimating activity takes on an expanded role of reviewing suggested process actions to provide an estimate of the ROI for each alternative action considered. With the selection of corrective action(s), a parametric model software project estimation model is updated to match as closely as possible project status and utilized to provide a revised estimate of effort, schedule, and risk between alternative proposed actions in order to establish a consistent ROI between alternative actions. The use of a parametric estimation tool provides a consistent method to evaluate between alternate actions that are being considered.

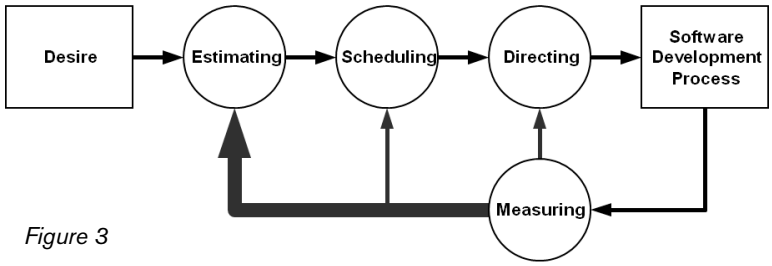

Figure 3

Without a parametric model, many software organizations incorporate one or more of the following to evaluate software project alternatives: 1) ballpark/rough order of magnitude engineering estimate, 2) top-down/constraint-driven estimating, and/or 3) bottom up/design-driven estimating. Despite the difference in planning methodologies, what all of these approaches have in common is that they tend to be executed as labor intensive practices dependent on the availability of scarce, over-committed personnel and manual or minimally-automated processes such as spreadsheets and other homegrown tools. Such practices are, by their nature, inconsistent, unpredictable, and highly susceptible to human error. Individual planners possess diverse capabilities and project histories; they may be overly optimistic or pessimistic; they may be influenced by internal politics or other factors unrelated to the project; or they may simply overlook some of the less obvious project elements.

Applying an established, proven estimation process that is well integrated with development processes helps ensure that project plans are credible and achievable, meet, or exceed customer expectations [7].

Parametric estimation models have been shown to provide a tool to develop the initial estimate and project plan as well as an effective means to evaluate project alternatives to optimize performance against goals [8]. With the establishment of a "high fidelity" parametric model during project conception, it only becomes necessary to maintain the model during project execution by updating the model effectively from the regularly scheduled metrics collected on the project. With parametric project modeling, it is expected that the standard metrics collected to support an Earned Value Management (EVM) process will provide sufficient update and calibration metrics for the model.

Performance-based Project Management

Central to Performance-based Project Management (PBM) is the goal of reducing risk in the forecasted effort and schedule of the final project outcome at each stage of the project. Application of PBM requires the establishment of a project baseline and estimate. That baseline will be considered the starting point for PBM. At project stages (regularly planned process points within the project) the size estimates are updated and the productivity driver assumptions are revisited. For example, the initial parameter settings with respect to tools, the level of testing, and the level of documentation will be revisited to see if the parametric model can be better calibrated to actual performance. Scheduling assumptions should be updated to reflect the actual performance.

The calibrated model may indicate a final cost and schedule that is not acceptable within the defined project goals. If that occurs, the application of the control process can be applied to evaluate the ROI for any changes that can be implemented into the process in order to improve performance against project goals.

Optimal Project Control

As defined previously, optimization of any activity requires the identification of a "cost function." With a parametric model in place to support analysis of project alternatives, a series of stages of the project will be defined. For example, stages might be defined on regular calendar intervals such as bi-annually, quarterly, or monthly depending on the size and planned duration of the project. Alternative stages could be defined at specific planned reviews such as preliminary design review or detailed design review.

Metrics are collected at each stage of the process and utilized in the "baseline" project model to provide a performance based

EVM projection of the estimated cost to complete. The model is then used to evaluate defined alternative management actions against the project optimization cost function. The "best" corrective action is chosen and applied. With revised parameters defined the model is updated to correspond with the time at the stage and a revised project plan with the incorporated project control changes is developed [5].

This process has been applied in multiple instances for clients in the field. A simple example software program for an unmanned missile command-and-control function has been used to provide a demonstration of the concepts. With that application, the project was approximately 48K SLOC of reuse and new code with a resultant 36K of Effort SLOC. The cost function was defined so as to minimize the difference between the actual project effort and the budgeted project effort. At the quarterly review the metrics to indicated completion of systems requirements and about 90% of the software requirements, which should have been completed. The Cost Performance Index (CPI) was 0.96. SEER-SEM with the four dimensional earned value capability of the project management control provided the EAC for both effort and schedule. Using the parametric model of the project the actual metrics were input as a "snapshot" of the project with an estimate of an increase in effort against budget at completion of approximately 3,000 effort hours and 1.2 schedule months. The model was utilized to evaluate the impact of alternate management actions. It was determined the best ROI would result from the change to a development system that minimized the rehost of software from the development to the target system. Assuming that change and updating the model to reflect that revised management input, the program was continued with the result that at the next quarterly review the CPI indicated a .98.

Dynamic Programming

In order to establish the effective best control changes, the theory of dynamic programming was utilized. Dynamic programming is a method of solving problems that exhibit the properties of overlapping sub-problems and optimal substructure [9]. By applying the dynamic programming paradigm it can be shown that by optimally selecting sub-projects at each stage of a project we can be assured of an optimal (or best) result for the total project. As applied to our management control of software projects, the theory validates that we will have a "best solution" by:

1. **Establishing and implementing a valid project plan.**
2. **Monitoring results on a regular basis (i.e. utilize metrics to evaluate performance against the project goal or cost function).**
3. **Define corrective management actions when indicated by the metrics analysis.**
4. **Evaluate the corrective actions to determine a "best" change in performance of the process.**
5. **Repeat the actions at the next stage.**

Following the process will result in a "best" project performance.

Summary and Conclusions

The use of parametric models is well accepted for software project estimation and planning. Maintaining the models at a high fidelity level for the duration of a project has been demonstrated to provide high confidence estimates of expected cost to complete and optimized selection of corrective actions. Updated models when calibrated to the actual performance provide a confidence level estimate of future performance based on results to date. The updated models generate stoplight charts and a set of information similar to EVM outputs but based on project estimates from a calibrated model.

Cost/effort parametric estimating models provide a powerful aid to project management, yielding "objective information" for systems tradeoffs, project management decisions, and controlling project performance throughout the program's lifecycle.

ABOUT THE AUTHOR

Dr. Denton Tarbet is a Senior Consultant with Galorath Incorporated. His research area at the University of Houston was Optimum Control Theory and he has more than 40 years experience in systems and software engineering, successfully managing projects with management responsibilities as Director of Engineering, Chief Technical Officer, and Vice President of Engineering. He has performed size and effort estimation projects for NASA, Air Force, Navy, Republic of South Korea, and multiple contractors.

E-mail: dtarbet@galorath.com
Phone: 310-414-3222

REFERENCES

1. Web site: <http://www.irise.com/blog/>
2. Conference Proceedings: Ross, M, Parametric Project Monitoring and Control – Performance-Based Progress Assessment and Prediction, SCEA, April 2005
3. Journal: Tarbet, D., On Time and On Budget, as Specified: An integrated approach to the Software Development Lifecycle, The Rational Edge, IBM Jan. 2008.
4. Journal: Smith, L. Overview of Project Management, STC Crosstalk, Jan 2003
5. Conference Proceedings: Tarbet, D, Software Project Management – Controlling the Process, PSM User's Conference, Vail, CO, July, 2007
6. Conference Proceedings: Madachy, R and Tarbet, D, Case Studies in Software Process Modeling with System Dynamics, Software Process Improvement and Practice, 5(2-3), 2000 (Initial version in Proceedings of ProSim Workshop 1999.)
7. Book: Galorath, D and Evans, M, Software Sizing, Estimation, and Risk Management, Auerbach Publications, Boca Raton, FL, 2006.
8. Conference Proceedings: Tarbet, D, Project Management Decisions Based on SW Metrics – Modeling Beyond the Estimate, SSCAG Software workshop, Santa Monica, CA Jan., 2006
9. Book: Bellman, Richard, Applied Dynamic Programming, Princeton University Press, Princeton, N.J., 1962