December 2003

Decision Making in Software Projects: A Control Theory Perspective

Arpan Jani
University of Minnesota

DECISION MAKING IN SOFTWARE PROJECTS: A CONTROL THEORY PERSPECTIVE

Arpan Jani
University of Minnesota
ajani@csom.umn.edu

Abstract

Software projects are difficult to control given their complexity, intangibility and dynamic nature. Control theory has been applied in several domains in management. Control theory can provide a useful framework for understanding the learning and adaptive processes involved in managing software projects. Software project management can be understood as a goal driven activity with a hierarchy of goal structure. This research draws upon control theory to propose a hierarchy of control in software project management. Specifically, within this hierarchy of control, the level of project decision making is elaborated and propositions on managerial attention allocation, the role of mental models, feedback and decision strategies are put forth. Further research should test these propositions empirically.

Keywords: Control theory, software project management, decision making

Introduction

Building large scale software has been considered one of the most complex enterprises. It is complex not only because of its size but also due to its interactive complexity such that problem in one area can create rippling effects elsewhere (Kraut and Streeter, 1995). Software development projects have been notorious for the failure to meet the cost, quality and schedule requirements (Lyytinen and Hirchheim, 1987). Even though billions of dollars are spent on IT projects, many projects go out of control and end up with huge cost over-runs and schedule delays (Keil, 1995). Many projects have to be abandoned later on. A 1995 study by Standish group showed that only 16% of the software projects among the 8000 surveyed were finished on time and within budget.

There are many organizational and environmental reasons, some quite outside the control of managers, that lead to delays in the completion of projects. In case of software projects, estimating the project costs and duration has been proved to be difficult (Thayer and Fairley, 1994). Managing software projects is a complex, dynamic task. The intangible nature and complexity of the software being built and factors like staff turnover makes it difficult to judge the percentage of work that is complete. This may give a false perception that the completion is nearer (Montealegrea and Keil, 2000). Even though better estimating techniques have been developed over the years, until the project reaches towards the end, it is difficult to assess the actual work accomplished. After various modules of the software are integrated, the system testing may discover new bugs, needing further rework. Often customer requirements change leading to changes in the initial estimates of the project. Thus past estimates about the project scope and completion may have to be revised at a later stage. By anchoring to their initial estimates, managers may show conservatism in adjusting their estimates despite gaps between initial estimates and actual progress (Sengupta and Abdel-Hamid, 1996). Sometimes managers may continue to allocate resources to failing software projects even though it may be advisable to cut back resource allocation or terminate a failing project (Keil, 1995).

Software project management is defined as "the application of formal and informal techniques, tools, methods, and heuristics (collectively called 'project management practices') which are used to motivate and guide the project team members to carry out the software project within the set of constraints" (Kirsch, 2000). Kirsch (2000) argues that much of research in software project management has focused on hard skills pertaining to application of tools and techniques to manage a project and has paid less attention to the soft skills of managing people. There is also the need to look at the learning and adaptive processes within project management as it provides another dimension to research on software project management. There is an implicit notion in the project management literature that project management in general and software project management in particular is a goal oriented organization of activities involving a series of decisions spaced over time to achieve the goals set out in the beginning. Thus software project management could be conceptualized as an adaptive process of gaining control over an unfolding situation.

We argue that control theory provides a useful theoretical framework of understanding decision making in software projects.[1] We first discuss control theory and the hierarchical conceptualization of feedback processes and goals of perceptual control theory (Powers, 1973) and how it relates with software project management. One could develop models of feedback processes at each of the levels in Powers' hierarchy as it is applied in the context of software project management. We focus upon decision making as identified within the hierarchy of control processes of software project management. Then we discuss decision making from a control theory viewpoint and relate it with managing software projects to develop propositions on attention allocation, effect of mental models and feedback and decision strategies of project managers.

Control Theory

Cybernetic or control theory has proved quite useful in understanding a variety of phenomenon pertaining to self-regulating systems be it physiological, biological or social systems. In psychology and management literature, control theory has found applications in understanding motivation (Lord and Hanges, 1987), negotiation (Brett, Northcraft and Pinkley, 1999), self-regulation (Carver and Scheier, 1998) managerial effectiveness (Tsui and Ashford, 1994) and decision making (Brehmer,1992) to cite a few examples. Feedback processes are fundamental aspect of control within a cybernetic system (see figure 1).

The input function detects the current state of the variable to be controlled. The reference value sets the standard or the desired value for the response of the controlled variable. The comparator detects the "error" or the discrepancy between the current state of the variable and the reference value, which generates a response to effect changes in the controlled variable to reduce this discrepancy. The reference value can be considered as the goal to be attained. The response of the controlled variable will also be independently influenced by the disturbances from the environment which may cause it to deviate or come closer to the reference or goal state. The sensing of deviation triggers a response to bring the system to the reference or goal state. The system as a whole is self-regulatory since it attempts to keep the variable under control despite the disturbances from the environment.

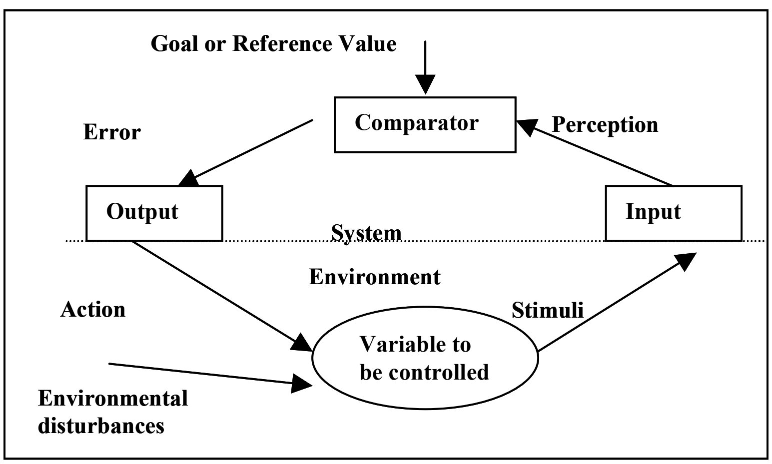

Figure 1. Cybernetic Model of a General Control Process

Powers (1973) conceptualizes control system as having a hierarchy of feedback processes with nesting of superordinate and subordinate systems (Powers, 1973, Carver and Scheier, 1998). The goals in a control system can be decomposed into a hierarchy (Miller, Galanter and Pribram, 1960).The reference value or goal for a given subordinate system will be derived from the

[1]Application of feedback principles in software engineering is not new. The work by Abdel-Hamid and Madnick (1991) to model software development using system dynamics approach is quite comprehensive. Their emphasis however is on modeling the software development process in an integrated manner and to understand the implications of alternate managerial policies on project outcomes. Just as one can apply the theoretical lens for multiple levels of analysis, here our emphasis is on understanding the decision making behavior of the project manager as he/she goes about the task of managing the project.

corresponding superordinate system. What is a reference value of the subordinate system will be the controlled value for the superordinate system. The superordinate goal can be decomposed in a hierarchical manner such that the accomplishment of a given goal necessitates accomplishment of lower level goals (Powers, 1973). The reference value, which may be quite abstract at the highest level, will become more concrete as one moves across the levels of hierarchy. For example, in the context of project management, achieving project profitability may be a superordinate goal which can be decomposed into managing the project on time, within budget, which in turn would necessitate that the sub-stages of the project (the activities related to analysis, design etc.) are completed within the planned time frame and budget so forth. Errors or discrepancies in the higher level systems can be resolved by control at the lower level (Lord and Levy, 1994 p. 339). As Lord and Levy (1994) describe "moving up one level explains *why* an action is done (to reduce discrepancies in higher-level systems) and moving down a level explains *how* discrepancies are reduced (by the operation of lower-level systems)" (emphasis original). There is the notion of both top-down and bottom up control (Lord and Levy, 1994). When the reference values are determined by the super-ordinate control system, it is a top down control. However, operations at lower level could also influence change in the reference values. Further it is also not necessary to assume the standard to be fixed as the process of gaining control could also alter the standard or goal. For example, when the goal is too difficult to achieve, one may alter the goal such that it is more achievable or one might focus on other goals.

Powers' formulation of hierarchical control system goes upto 11 levels including the physical execution of actions (Robertson and Powers, 1990). However in our context of software project management the first four levels are relevant viz. *system concepts*, *principal*, *program* and *sequence*. In Table I the first four levels of Powers' hierarchy have been mapped with corresponding layers in project management related tasks and the goals at each level. In Powers' hierarchy, the *system concepts* form the highest level of goal abstraction which deals with the issue of "what should be". This relates with the strategic level of organizational decision making. In the context of software projects, this will link to the question of how the IT project supports the strategic objectives. The *system concepts* level provides goals for the next level i.e. *principal control* which formulate the principles that would achieve the system level control.

Table 1. Levels of Control Hierarchy

Control Hierarchy Levels	Project Hierarchy	Hierarchy of Goals Relating to each level
System Concepts	IS Strategic planning	Strategic objectives
Principle	Guidelines, Policies, Norms, Best Practices,	Project profitability/ROI, providing platform for future development
Program	Project portfolio management, Project planning	Overall Project portfolio performance, Project Plan, Performance Metrics
	Project decision making	Cost, quality, schedule, functionality etc.
Sequence	Activities	Completion of tasks

In organizational and project management context this would relate to the development of policies, norms and guidelines and planning activities. The next level is called the *program level* which manifests in terms of the courses of action with explicit decision points that attains the goals set at the principal level. This level relates with the decision making in projects. We could further split the program level in the context of project management into i) activities of project portfolio management in terms of selection of projects to be carried out and resource allocation across projects and ii) decision making within projects. The goals at project level decision making are typically related to cost, quality, schedule, and functionality of the software delivered. Program control specifies the goals for the next level of *sequence control*. Sequence is a set of actions taken together. In organizational and project management context this would relate to the activities performed as a result of the decisions. Completion of the tasks (which may be developed according to the Work Breakdown Structure (WBS) becomes a goal at this level. This is not to suggest that this is the only goal hierarchy that can be derived. For a superordinate goal, there could be multiple subordinate goals that could be derived such that the attainment of a superordinate goal could be through multiple paths (equifinality of achievement of goals). However as we mentioned earlier this is not strictly a top down process in which the goals are handed over to the lower levels. Control processes at lower levels also affect the standard setting at the higher levels. Problems in the achievement of goals at lower levels will result in revision of goals at the higher levels. Identification of failures at lower level could lead to change in the course of action at the higher level. However when that fails, one might also see its effects in terms of continued commitment to failing projects (Keil, 1995) where managers continue to 'throw good money after bad'.

As we can see a control theory conceptualization provides a useful and comprehensive framework for understanding the activities at different levels whether it is strategic, tactical or operational. Though one could develop frameworks for understanding goal oriented processes at different levels of software project management hierarchy, the focus in this paper is at the program level or the level of decision making in project management.

Control Theory and Decision Making

Control theory has been applied to the study of dynamic decision making (see Brehmer (1990, 1992) for a review). Brehmer (1992) has argued that decision making in many situations can be characterized more as gaining control over a situation rather than as a discrete choice based on considerations of alternatives and consequences. The latter takes a more static approach of decision making whereas the former emphasizes the continuous process of judgment-action-outcome feedback loops. Thus judgment is viewed as an ongoing process interrupted by choices rather than a discrete event (Hogarth 1981). Decision making is seen within a broader context of purposive behavior where decisions are made to achieve some ends. Shapira (1995) argues that managerial decision making is a continuous process in which each decision is dependent upon previous decisions: "when managers take upon risky projects they do not think about it as a one-point-in time decision". Shapira (1995) quotes a manager in his research study as saying "decision theory puts all the emphasis on the analysis leading to the moment of choice. While it is definitely important, my experience has taught me that my ability to influence whatever goes on *after* the moment of choice is perhaps even more important" (emphasis original).

The task situations that can be framed under dynamic decision making perspective have the following characteristics (Edwards 1962, Brehmer 1992): i)the task requires a series of decisions, each decision interdependent upon others. ii) The decision environment changes autonomously and as a result of the decisions. iii) Moreover the decisions have to be made in real time or in other words timing of the decision matters (Brehmer 1992). As we can see, a number of decision situations can be characterized by dynamic decision making e.g. medical decision making, fire fighting, managerial decision making. However as Doherty (1990) pointed out, we acknowledge that not all decisions are dynamic and require a series of decisions. Tasks like the decision to get married, taking up or quitting a job, or hiring someone are one-shot kind (Doherty, 1990).

Decision Making in Software Project Management

Managing software projects is a dynamic task (Sengupta and Abdel-Hamid, 1993, Sengupta and Abdel-Hamid, 1996). Many of the problems of software project management have been linked to its dynamic nature where the business requirements and technologies change during the project life cycle. The decision to allocate resources affects future decisions (inter-dependence of decisions), the project environment changes autonomously (e.g. changing customer requirements leading to expansion of project scope, staff turnover) and as a result of the past decisions. Moreover, the correct timing of decisions matters a lot. It has been recognized that software project management requires managing multiple goals, some of which may be conflicting (Boehm, 1981, p.205) like maintaining costs within budget, meeting the schedule and achieving the desired level of quality within a single project. In case of managing multiple development projects concurrently there is also the constraint of allocating limited resources among projects competing for the same resources.

We present a cybernetic model of software project management at the decision making level (program level) in the control hierarchy identified earlier (see figure 2) which incorporates the standard, discrepancy sensing and discrepancy reduction processes. The project plan comprises of the cost and schedules estimates for the project, the resource requirements, the amount of work to be accomplished within the scheduled time and budget, scheduling of the resources and activities and so forth. The project goals are derived from the software project plan, which are in terms of the functionality of the software to be delivered, the budget, schedule and quality. The project plan also determines the sub-goals or milestones to be achieved as the project progresses further. The project manager possesses a mental model of the task, follows a decision process and applies the decision rules considering the current and future state of the project, the current gap in the final goals and sub-goals to be achieved. 'Mental model' is a concept often invoked to explain the way people understand and interact with the complex world. Mental models are internal symbolic representations of the world. Mental models can be considered mental maps that enable people to take actions based upon their limited conceptions about the world (Weick, 1990). Single loop learning refers to the actions taken and the revisions in the course of action taken based on the feedback received to enable the project manager to reach the goals. Double loop learning refers to the recognition of errors in ones plans, policies and goals or ones mental models (Argyris and Schon, 1978). Learning to revise ones action plan based on the feedback received to reach the given goals is an indication of single loop learning. However when managers revise their own mental models, plans and goals on the basis of feedback, it is an indication of double loop learning.

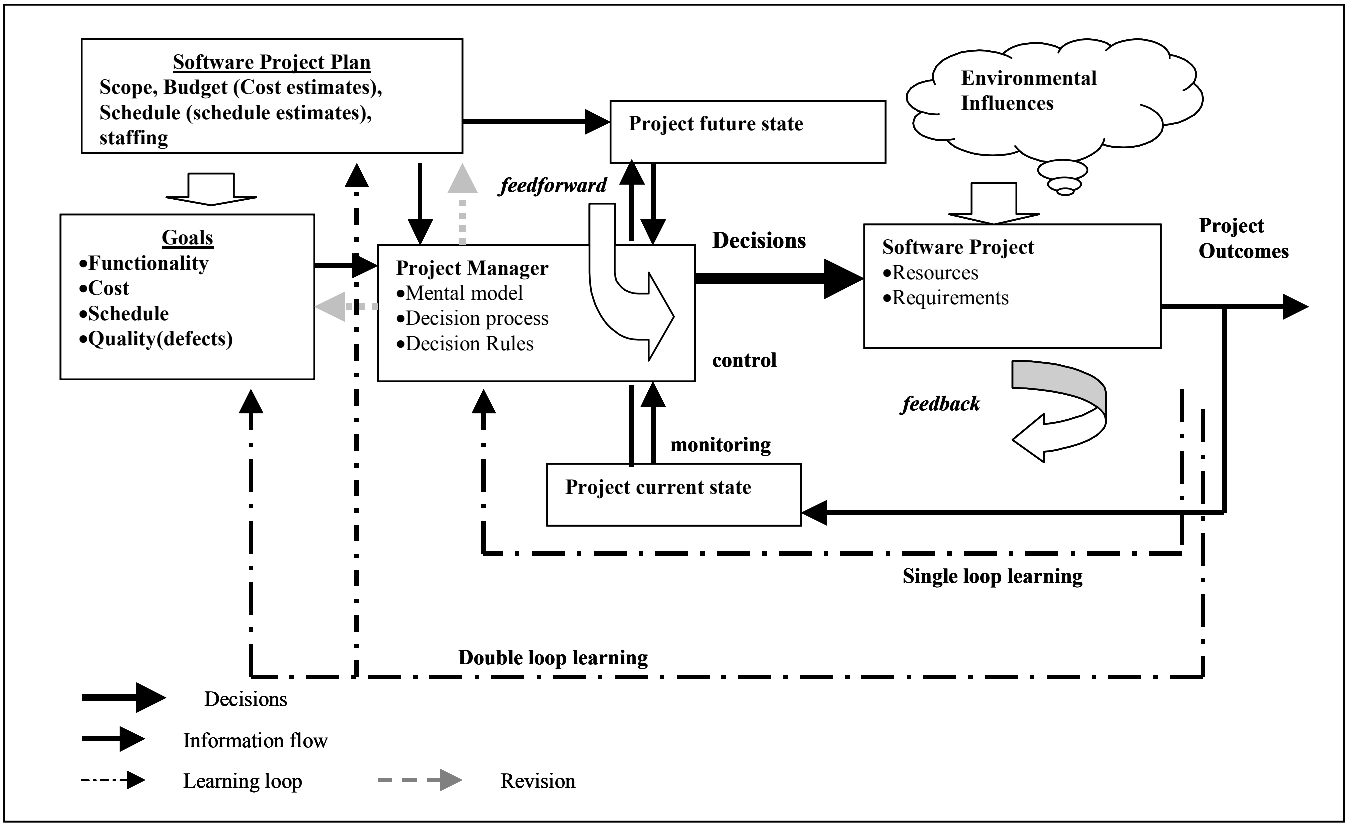

Figure 2. Cybernetic Model of Software Project Decision Making

Attention Allocation

Managerial attention is a scarce resource. Differentiating between substantive and procedural rationality, Simon (1978) had argued for the need to explain decision maker's behavior in complex, dynamic circumstances that involve a great deal of uncertainty and that make severe demands on managerial attention. As Simon (1978, p. 13) had pointed out "In a world where information is relatively scarce, and where problems for decisions are few and simple, information is almost always a positive good. In a world where attention is a major scarce resource, information may be an expensive luxury, for it may turn our attention from what is important to what is unimportant." Decision makers will be selective in what they pay attention to and what decision makers do depends on what issues and answers they focus their attention upon (Ocasio, 1997). Managers may focus their attention among the sub-goals at the same hierarchical level or shift their attention across the hierarchy. Borrowing from Brett, Northcraft and Pinkley's (1999) conceptualization of two types of discrepancy reduction processes, we could consider the shift in the focus of attention at the subordinate goals as horizontal shift and the shift in the attention across hierarchical levels (from sub-ordinate to superordinate and vice versa) as horizontal shift. Goal discrepancy would be the trigger for attention to the achievement of a particular goal. Greater the discrepancy for achievement for a goal, greater will be the efforts When there are multiple goals to be achieved, decision makers would need to prioritize what they pay attention too since attentional resources are limited (Kernan and Lord, 1990). However if the discrepancy remains despite the repeated efforts to reduce the discrepancy, it may lead to refocusing of the attention such that different goals are pursued. Thus we put forth the following propositions

> ***P1: Failure to achieve goals at the lower level would lead managers to focus attention at the higher level goals (vertical shift)***

> ***P2: While pursuing multiple goals in a software project, managers would tend to prioritize what goals they would pay attention to at a given time.***

> ***P3: The prioritization of goals would be related with the discrepancy for each goal.***

> ***P4: There will be a curvilinear relationship between goal discrepancy and attention allocation for the achievement of a goal***

Mental Models

For effective control of a system, the controller must have an adequate representation of the controlled system (Conant and Ashby, 1970). For effective project management, the project manager must have an adequate mental model of the complex task of project management. Research in decision making in dynamic environments and process control have identified a number of factors that make it difficult for decision makers to gain effective control of a dynamically complex system (e.g. Hall, 1984, Sterman, 1989a; 1989b). Deihl and Sterman (1995, p. 198) point out three deficiencies of decision makers in dynamic, complex settings, what is termed as 'misperceptions of feedback': "people generally adopt an event-based, open loop view of causality, ignore feedback processes, and fail to appreciate time delays between action and response and in the reporting of information, do not understand stocks and flows, and are insensitive to nonlinearities which may alter the strengths of different feedback loops as system evolves." As Sterman (1989a, p. 325) writes "It is as if the subject purchases a car, but has to wait for the delivery. The next day, the garage is still empty, the subject goes to the dealer and orders another one." Thus we propose that

> ***P5: Software project managers with appropriate mental model of the project management task will perform better at project control than project managers with inadequate mental model.***

Feedback

The control theory perspective emphasizes the adaptive or leaning aspects of decision making. As one attempts to gain control of a situation, one gets an opportunity to correct ones' errors from the feedback of the choice outcomes (Kleinmuntz and Thomas, 1987). Some of the biases of decision making that are evident in static task environments may be adaptive when seen from continuous perspective. For example, Hogarth (1981) suggest that the anchoring and adjustment heuristic is an indication that the decision maker is attempting to revise ones judgment as one receives more feedback from the environment about the effectiveness of the initial judgment. Studies of decision making in software projects have found that managers are conservative in revising their initial estimates when they receive information that suggests that the initial estimates were inaccurate (Sengupta and Abdel-Hamid, 1996, Abdel-Hamid, Sengupta and Ronan, 1993). This is especially more likely when the feedback information itself is fallible. Inaccurate project estimates have been identified as one of the major causes of problems in software project control. Feedback is not automatically available to the decision makers that would indicate the deviations from the standards. Managers would need to actively search for feedback information (Tsui and Ashford, 1994) and their effectiveness may be related to this (Ashford and Cummings, 1983). Since feedback provides also an opportunity to correct past errors, the frequency of feedback should be related to the effectiveness of the course of action. Thus we propose

> ***P6: Software project managers that actively seek feedback information should perform better at controlling a software project than those who do not seek feedback actively.***

Decision Strategies

The strategies used by the project manager for control could be feedback or feedforward (Brehmer, 1990). Managers may use the combination of both feedback and feedforward strategies. Feedback strategy involves choosing actions based on the current state of the project whereas feedforward strategy would require projection of future states of how the project would unfold. Feedforward strategy is useful when the system remains relatively stable. In case of constantly changing condition, feedback strategy would be a better approach. However in case of feedback strategy there should not be delays in feedback. Feedback control is cognitively simpler and Brehmer (1990) argues that it would be preferred in dynamic decision situations. In case of software projects it is unlikely that the project scope and estimates may remain the same as the project unfolds. Thus we propose that

> ***P7: Software project managers will be more likely to use feedback strategy for control rather than a combination of feedforward and feedback.***

Conclusion

Software projects are difficult to manage. Several factors including the size, interactive complexity, and dynamic nature contribute to the problems of gaining control over projects. Project management practices influence the project outcomes. Project management is a goal-directed activity. Understanding what project managers do as they attempt to gain control over an unfolding, dynamic situation is vital for developing prescriptions to improve the practice of software project management. Unlike the rational tradition that assumes that managers have fixed preferences, perfect information and knowledge of consequences and alternatives, control theory emphasizes the learning and adaptive processes in decision making. Control systems conceptualization of a software project enables one to see a hierarchy of feedback processes and the multiple levels of goals that the control processes are seeking. We believe that application of control theoretic framework to software project management is an important step in understanding ideas about attention allocation, decision making and goal striving processes in project management. Future work needs to test these ideas empirically.

References

Abdel-Hamid, T. K., and Madnick, S. Software Project Dynamics: An Integrated Approach, Englewood Cliffs, NJ: Prentice-Hall, 1991.

Argyris, C., & Schon, D. Organisational Learning: A Theory of Action Perspective, Reading, MA: Addison Wesley, 1978.

Ashford, S. J., and Cummings, L. L. "Feedback As an Individual Source: Personal Strategies for Creating Information," Organizational Behavior and Human Performance, 32, 1983, pp. 370-398.

Boehm, B. Software Engineering Economics, Englewood Cliffs, NJ:Prentice Hall. 1981.

Brehmer, B. "Strategies in Real Time, Dynamic Decision Making," In R. Hogarth (Ed.) Insights in Decision Making. Chicago,IL: University of Chicago Press, 1990.

Brehmer, B. "Dynamic Decision Making: Human Control of Complex Systems," Acta Psychologica, 81, 1992, pp.211-241.

Brett, J. F., Northcraft, G. B., and Pinkley, R. L. "Stairways to Heaven: An Interlocking Self-Regulation Model of Negotiation," Academy of Management Review, 24 (3), 1999, pp. 435-451.

Carver, C. S. and Scheier, M. F. On the Self-Regulation of Behavior, Cambridge, UK: Cambridge University Press, 1998.

Conant, R. & Ashby, W. "Every Good Regulator of a System Must be a Model of the System," International Journal of System Sciences, 1, 1970, pp. 89-97.

Diehl, E. and Sterman, J. D. "Effects of Feedback Complexity on Dynamic Decision Making," Organizational Behavior and Human Decision Processes, 62 (2), 1995, pp. 198-215.

Doherty, M. E. "Discussion," In Robin M. Hogarth (ed.) Insights in Decision Making: A Tribute to Hillel J. Einhorn, Chicago: The University of Chicago Press, pp. 280-291,1990.

Edwards, W. "Dynamic Decision Theory and Probabilistic Information Processing," Human Factors, 4, 1962, pp. 59-73.

Hall, R. I. "The Natural Logic of Management Policy Making: Its Implications for the Survival of an Organization," Management Science, 30 (8), 1984, pp. 905-927.

Hogarth, R. M. "Beyond Discrete Biases: Functional and Dysfunctional Aspects of Judgmental Heuristics," Psychological Bulletin, 90, 1981, pp. 197-217.

Keil, M. "Pulling the Plug: Software Project Management and the Problem of Project Escalation," MIS Quarterly, 19 (4), 1995, pp.421-447.

Kernan, M.C. and Lord, R.G. "Effects of Valence, Expectancies, and Goal-Performance Discrepancies in Single and Multiple Goal Environments," Journal of Applied Psychology, 75 (2), 1990, pp. 194-203.

Kirsch, L. "Software Project Management: An Integrated Perspective for an Emerging Paradigm," In R. W. Zmud (Eds.) Framing The Domains of IT Management: Projecting the future through the past, Cincinnati,Ohio: Pinnaflex, 2000.

Kleinmuntz, D.N. and Thomas, J.B. "The Value of Action and Inference in Dynamic Decision Making," Organizational Behavior and Human Decision Processes, 39, 1987, pp. 341-364.
Kleinmuntz, D.N. "Cognitive Heuristics and Feedback in a Dynamic Decision Environment," Management Science, 31 (6), 1985, pp. 680-702.
Kraut, R. E. and Streeter, L.A. "Coordination in Software Development," Communications of the ACM, 38 (3), 1995, pp. 69-81.
Lord, R. G. and Hanges, P. J. "A Control System Model of Organizational Motivation: Theoretical Development and Applied Implications," Behavioral Science, 32, 1987, pp. 161-178.
Lord, R. G. and Levy, P. E. "Moving from Cognition to Action: A Control Theory Perspective," Applied Psychology: An International Review, 43 (3), 1994, pp. 335-398.
Lyytinen, K., and Hirschheim, R. "Information Systems Failure – A Survey and Classification of Empirical Literature," Oxford Surveys in Information Technology, 4, 1987, pp. 257-309.
Miller, G. A., Galanter, E., and Pribram, K. H. Plans and the structure of behavior. New York, NY: Holt, Rinehard and Winston, 1960.
Montealegre R., and Keil, M. "De-escalating Information Technology Projects: Lessons from the Denver International Airport," MIS Quarterly, 24, 2000, pp. 417-447.
Ocasio, W. "Towards an Attention-Based View of the Firm," Strategic Management Journal, 18 (Summer Special Issue), 1997, pp. 187-206.
Powers, W. T. Behavior: The Control of Perception, Chicago,IL: Aldine Publishing Company, 1973.
Robertson, R. J. and Powers, W.T. An Introduction to Modern Psychology: A Control Theory View. Gravel Switch, KY: The Control Systems Group Inc., 1990.
Sengupta, K. and Abdel-Hamid, T.K. Alternative Conceptions of Feedback in Dynamic Environments: An Experimental Investigation. *Management Science,* 39, 1993, pp. 411-428.
Sengupta, K. and Abdel-Hamid, T.K. "The Impact of Unreliable Information on the Management of Software Projects: A Dynamic Decision Perspective," IEEE Transactions on Systems, Man, and Cybernetics, 26, 1996, pp. 177-189.
Shapira, Z. Risk Taking: A Managerial Perspective, New York: Russell Sage Foundation,1995.
Simon, H. A. "Rationality as Process and Product of Thought," The American Economic Review, 68 (2), 1978, pp. 1-6.
Sterman, J. D. "Misperceptions of Feedback in Dynamic Decision Making," Organizational Behavior and Human Decision Processes, 43 (3), 1989a, 301-335.
Sterman, J. D. "Modeling Managerial Behavior: Misperceptions of Feedback in a Dynamic Decision Making Experiment," Management Science, 35 (3), 1989b, 321-339.
Thayer, R. H. and Fairley, R. Project Management, In J. J. Marciniak (Ed.). Encyclopedia of Software Engineering. Volume II, pp. 900-923, NY: John Wiley, 1994.
Tsui, A. S. and Ashford, S. J. "Adaptive Self-Regulation: A Process View of Managerial Effectiveness," Journal of Management, 20 (1), 1994, pp. 93-121.
Weick, K.E. "Cartographic Myths in Organizations," In Huff, A.S., (Ed.) Mapping Strategic Thought, Chichester, UK: Wiley, 1990.