

Is There an Underlying Theory of Software Project Management? (A critique of the transformational and normative views of project management)

Version 4.0, 1.30.04

Glen B. Alleman
Niwot, Colorado
glen.alleman@niwotridge.com

Abstract: Traditional project management methods are based on scientific principles that would be considered “normal science,” but lack any theoretical basis for this approach. [33, 34, 65] These principles make use of linear step-wise refinement of the project management processes based on a planning-as-management paradigm. Plans made in this paradigm and adjusted by linear feedback methods cannot cope with the multiple interacting and continuously changing technology and market forces. They behave as if they were is linear closed loop control system.

This paper suggests that adaptive control theory may be a better foundation for a model for project management. Using closed loop adaptive control system rules, parallels are drawn between control systems and agile project management. From these parallel, a comparison can be made between project management practices and adaptive control algorithms.

Finally a control systems view of the project management practices is provided with a discussion of how this view can be applied to agile project management practices. Constructing the connection between control systems, especially adaptive control systems, and project management is the goal of this paper. This project management process is then applied to the management of software development and the agile methodologies currently moving into the marketplace.

1 Introduction

Since the earliest days of the computer software industry managing of software development projects has been fraught with uncertainty and risk. While the technical content of software products and the technical methods used to build them have changed over time, the fundamental issues that determine the success or failure of software projects have remain constant. The fundamental management mistakes have remained the same.

The success rate of applying traditional methods to complex software development projects over the years has been underwhelming. [29] This linear step-wise approach has its roots in the waterfall methods of the 1970's. It was clear then [63, 62] and has become even cleared today that this approach to managing software projects is inappropriate in many domains. What is not answered in the project management literature is the question – is there an underlying theory of project management appropriate for software development projects? A secondary question is – can a theory be constructed that is consistent with adaptive system and agile processes currently in use in manufacturing, science, economics, biology, and ecology?

One approach is to look for theories in other domains that closely match the behavioral aspects of project management. Control systems theory is one such domain. Performance references, control loops, and stochastic processes all have similar paradigms in project management. In addition the theory of complex adaptive systems and adaptive controls for those systems has a similar paradigm in the “agile” domain.

The development of software systems has substantial elements of creativity and innovation. Predicting the outcome of the development effort, given a fixed set of resources and time is difficult. Add to this process external market forces, incomplete or ill-formed requirements and changing stakeholder's needs creates three questions for consideration: a) what methods are appropriate for the management of software development projects? b) what theoretical aspects of project management can be applied to the software environment? c) what gaps exist in current project management methods that should be closed to meet these new needs?

1.1 Project Management Theory

The current project management literature describes project management in terms of controlling, planning, and scheduling. This literature often assumes project management takes place within the paradigm of *management-as-planning*. This paradigm holds the view there is a causal connection between the actions of management and the outcomes of the project. Assuming the translation of plans into action as a simple matter of execution. This view of project management regards projects as instruments with which to achieve a goal rather than as individual organizations in their own right.¹ Feedback from this planning process is based on an *after the fact* variance detection. As a feedback control system, gaps in the feedback include: delays that can be used to correct the plans and execution before the deviation grows too large, adaptive

¹ The origins in industrial society can explain why much project management theory assumes that projects take place within a single organization. This basic assumption is out of step with post-industrial society's joint ventures, and strategic collaborations.

planning through adaptive feedback loops, and feed-forward controls to direct the execution based on inputs about future needs of the stakeholders.

In the literature, project management methods are reduced to stable, technical, and linear processes.² The impact on the project from external forces or from problems within the project is given little attention. It is assumed in this traditional model that “change” is an undesirable thing, when in fact change in the business systems world is not only natural it is desirable. The conflict between “managing in the presence of change” and “managing change” by attempting to control it is the source of many of the gaps between traditional project management and agile project management.

One approach to defining a theory of project management can be found in [45]. It is conjectured that a well-functioning bureaucracy aided by scientific planning tools can efficiently deal with a project through these “normal-science” methods. This approach assumes projects are carried out under conditions of complete rationality.³ It also assumes that projects are repetitive, with their requirements and stakeholder needs built existing knowledge.

The majority of software development projects are not conducted under conditions rationality. Software projects are not repetitive, stable, or linear. They are unique, driven by unstable requirements, technology, and market forces, and contain many non-linear activities. Software development is complex, the exact business and tech-

² Linear project management models are sometimes referred to as *waterfall* models. In these models it is assumed that each phase of the project is completed in a fixed sequence, followed by the next logical phase.

³ All rational action embodies some sort of precautionary principle. What kind of harm can be averted? What kinds of cost are willing to be incurred by the stakeholders? In the rational context, risks can be pre-identified, production rates are known, defects can be statistically analyzed, and requirements can be elicited up front.

nical outcome is difficult to plan. The processes used to manage the outcome may be chaotic. Software projects are often subjected to forces outside the control of the project manager, developers, and stakeholders.

More importantly the development and deployment of software creates a non-linear feedback loop between the development and the deployment processes. Once the software is deployed the users have new and sometimes disruptive requirements – once they know understand how the application works.

The framework for examining this situation can be found in a similar approach to management of systems engineering activities. [59]

Figure 1 presents an overview of both the elements and dimensions of project management. The “control systems” involved in project management are not shown, since this is a static view of the elements and their interactions. The important aspect of Figure 1 is the connection between the components of the problem domain and the solution domain.

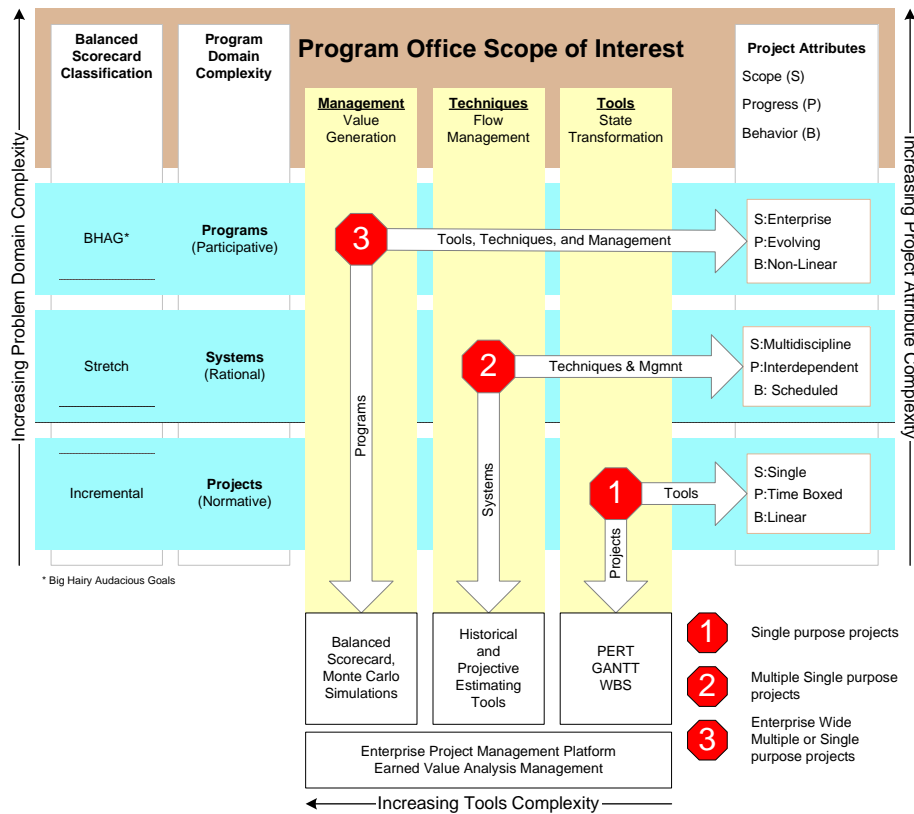


Figure 1 – Dimensions of Software Project Management

The normative advice provided by the traditional project management bodies of knowledge – planning, execution, and control – forms a closed loop linear system. This advice is usually based on *rules* that specify which choices will maximize benefits to the participants. Normative theory suggests a project is a series of sequentially related activities. In practice software project management is a set of multiply interacting interdependent activities behaving in a non-linear and adaptive manner. Complex adaptive systems (CAS) are one way of looking at project management. Adap-

tive control systems offer a simpler model without the complex and intractable mathematics of CAS.

The distinctions between traditional and agile can be summarized in Figure 2:

Traditional Methods	Emergent or Agile methods
Planning drives results	Results drive planning
Delivery is focused on planned results	Delivery is focused on derived results
Defined process steps	Self-organizing process steps using principles of <i>agile alliance</i> or similar statements.

Figure 2 – Distinctions between Traditional and Agile PM methods

1.2 Information Technology Project Management

Information Technology (IT) projects traditionally use formal management processes for the acquisition or development, deployment, and operation of the system that emphasizes planning in depth. This approach organizes work into phase's separated by decision points. Supporters of this approach emphasize that changes made early in the project can be less expensive than changes made late in the project.

In the past this approach has been called waterfall.⁴ The waterfall approach contains several erroneous assumptions that negatively impact IT projects:

⁴ The term *waterfall* has been used many times as a *strawman* by the agile community. In fact very few pure waterfall projects exist today. This is not to say there are not abuses of the concept of waterfall – sequential development based on the simple algorithm REPEAT [Design, Code, Test] UNTIL Money = 0. In practice, development and deployment processes based on incremental and iterative methodologies are the norm. The literature contains numerous references and guidelines to this iterative project management approach dating back to the 1980's [62].

- Planning – the assumption that it is possible to produce a plan so that its implementation is merely a matter of executing a defined set of tasks in a predefined order.
 - Plans for complex projects rarely turn out to be good enough for to remain intact through out the project life cycle.
 - Continuous re–planning, re–adjusting of priorities, and re–analyzing the consequences of these changes is common practice.
 - Unanticipated problems are the norm rather than the exception.
- Change – It is not possible to protect against late changes.
 - All businesses face late changing competitive environments.
 - The window of business opportunity opens and closes at the whim of the market, not the direction of the project manager.
- Stability – Management usually wants a plan to which it can commit. By making this commitment, they give up the ability to take advantage of fortuitous developments in the business and technology environment [66].
 - In a financial setting this is the *option value* of the decision.
 - Deferring decisions to take advantage of new information and new opportunities is rarely taken into account on IT projects [67].

1.3 Post–Normal Science

The term “Post–Normal” was coined by Funtowicz and Ravetz [26, 27]. A simple definition is...

(g)oinc beyond the traditional assumptions that science is both certain and value-free, it makes system certainties and decision stakes the essential elements of its analysis. It distinguishes between “applied sciences” where both dimensions are low, “professional consultancy” where one of the dimensions is salient, and “post-normal science” where at least one dimension is extreme.

Figure 3 describes the relationships between the various “domains” of scientific process to demarcate the realms of applied science, professional consultancy, and post-normal science.

The realm of applied science is the search for objective truth. The interests of the client are the realms of consultancy. Post-normal science contains a *theoretical core* of quality assurance. It argues the need for new methods with involve extended peer communities who deploy extended facts and take an active part in the solution of their own problems [28].

These concepts form the basis of many of the “agile” processes in management and development of software systems:

- Full participation of the stakeholders in defining the “value” delivered by the system.
- Emerging requirements from the “deployment” of the systems, rather than from the pre-definition of the system. ⁵

⁵ Many would argue that non-functional requirements need to be defined “up front.” And this is likely the case, since irrevocable decisions need to be made regarding infrastructure.

- Errors in the system are expunged as they are encountered, rather than through a formal process of quality assurance.
- The stakeholders take an active part in the solution. They become the “customer” in all of its logical definitions.

These concepts are distinctly different from the “normal” science point of view for software projects, in which analysis, design, code, and test are the typical linear cycles.

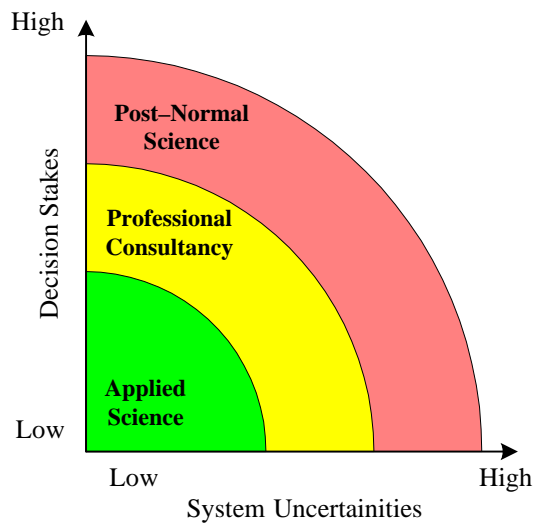


Figure 3 – Post-Normal Science Domain

1.4 Project Management as “Post-Normal Science”

Modern project management is heavily influenced by the belief that a project management process can be improved by scientific methods [13, 7]. These projects create a set of myths based on the “normal science” paradigm that:

- Clear-cut investment opportunities exist with an explicit purpose, beginning, duration, and end can be identified early in the project.
- Low opportunity costs for each business or technical decision exist, in most instances with a reversible decision process.
- Feasible, suitable, and acceptable project attributes can be identified.
- Accurate predictions of project duration and resource demands are possible once the requirements have been defined.
- Worst-case consequences can be determined in advance.
- The failure of the project was due to lack of technical and managerial skills rather than inappropriate feasibility, suitability, or acceptability of the solution.

This is a “normal science” view of software project management can be replaced with a post-normal view, in which there are:⁶

- Highly uncertain facts about the project attributes.
- Constant disputes about the values and expectations.
- High decision stakes with irreversible consequences.
- Urgently needed decisions must be made in the presence of insufficient information.

⁶ Classical science and conventional problem solving were labeled “normal science” by Kuhn [44]. Post-Normal science acknowledges there is high system uncertainty, increasing decision stakes, and extends the peer review community to include the participants and stakeholders, who insure the quality and validity of the conclusions [25, 49, 16].

- Outcomes that affect broad communities of interest beyond the direct participants and stakeholder in the project.

Agile methods do not mean that the normal–science model is irrelevant, just that such a model is applicable only when uncertainty and decision stakes are low [26].

A fundamental attribute of post–normal science is the reliance on heuristics. Using heuristics to guide the development using agile methods allows the management of IT projects to be placed in a post–normal science context.

1.5 Agile Methods

Agile methods have entered the market as a remake of Lightweight Software Development processes. Agile processes emphasize both the rapid and flexible adaptation to changes in the process, the product, and the development environment [Aoyama 98A]. This is a very general definition and therefore not very useful without some specific context — which will be developed below.

Before establishing this context, agile methods include three major attributes, they are:

- Incremental and Evolutionary – allowing adaptation to both internal and external events.
- Modular and Lean – allowing components of the process to come and go depending on specific needs if the participants and stakeholders.
- Time Based – built on iterative and concurrent work cycles.

- Self-Organizing – in the sense that normative guides have little to offer in terms of structure and control. Agile methods rely primarily on heuristics and participative processes rather than normative and rational methods and guidelines.

1.6 Project Management as a “Control System”

The vocabulary of the project management [17] is similar to that found in control systems [40, 49]. These terms includes:

Project Management Process	Control System
Monitoring – tracking and reporting of progress to a reference.	Reference signal – an independent variable (or set of variables) that defines the desired output. The error signal is the arithmetic difference between the reference signal and the output signal. ⁷
Evaluating – an assessment of the project’s progress to plan using some normative unit of measure, usually money, or time.	Plant or process – is a continuous operation or development marked by a series of gradual changes that success one another in a relatively fixed way and lead toward a particular result. An artificial or voluntary, progressively continuing operation that consists of a series of controlled actions or movements

⁷ In the case of a simple temperature controller, the reference signal is the desired temperature. The error signal is the difference between the desired temperature and the current temperature. If this error is positive, then the process is instructed to lower the temperature. If this error is negative, then the process is instructed to raise the temperature. This is a very *simple* example, but will serve to illustrate the point that project management has similar terms and concepts as closed loop controller.

	systematically directed toward a particular result or end.
Control – monitors and measures progress against plan to identify variances and provide corrective action, generating feedback to the decision making process.	Controller – which uses feedback, in the presence of disturbances, tends to reduce the difference between the output of the system and the reference input.

Figure 4 – Project Management and Control Systems Vocabulary

2 Project Management as a Control System

Control systems play an important role in engineering, science, economics, and biological systems. They has play an important role is creating *models* of other general systems, either as models of these systems or as metaphors of the models of these systems. [8].

Early control systems were based on linear feedback models. As the entities being controlled became more complex, the classical control theory, which dealt with single input and single output systems, became less useful. Multiple input and output systems now dominate control systems theory and practice. Recently adaptive and optimal control systems have been developed. Applications of modern control theory to non-physical fields are also the norm. Biology, economics, sociology and other dynamic systems are also common practice. Complex Adaptive Systems is a popular topic today.

Constructing a connection between control systems, especially adaptive control systems and project management is the goal of this section.

2.1 Basic Problems in Control System Design

Before moving forward some comparisons between control systems and project management systems will be helpful.

	Process Control	Project Management
Process	A natural and progressively continuing operation or development marker by a series of gradual changes that succeed one another in a relatively fixed way and lead toward a particular result.	
Systems	A combination of components that act together and perform a certain objective.	
Disturbance	A signal which tends to adversely affect the value of the output of a system.	
Feedback control	An operation which, in the presence of disturbances, tends to reduce the difference between the output of a system and the reference input.	
Damping		

	Process Control	Project Management
Feedback control system	A system which tends to maintain a prescribed relationship between the output and the reference input by comparing these and using the difference as a means of control.	
Closed loop system	Is one in which the output signal has direct impact on the control action, as shown in Figure 6. In a closed loop system the error signal, which is the difference between the input and the feedback is fed to the controller to reduce the error and bring the output of the system to a desired value.	
Open loop system	Is one in which the output signal has no direct impact on the control action, as shown in Figure 6. In an open loop system the output is neither measured nor fed back for comparison with the input. For each reference input there is a fixed operating condition.	

	Process Control	Project Management
Adaptive control system		
Performance index	<p>Is a quantitative measure of the performance, measuring the deviation from the ideal performance. The specification of the control signal over the operating time is called the control law.</p>	
Adaptive	<p>Any alteration in structure or function of an organism to make it better fitted to survive or multiply. Change in response of sensory organs to changed environmental conditions.</p>	
Learning control systems	<p>Many open-loop control systems can be converted to closed-loop control system if a human operator is placed in the loop. This operator compares inputs with outputs and makes corrective actions based on the resulting errors.</p>	

Figure 5 – Attributes of Control Systems and Project Management Systems

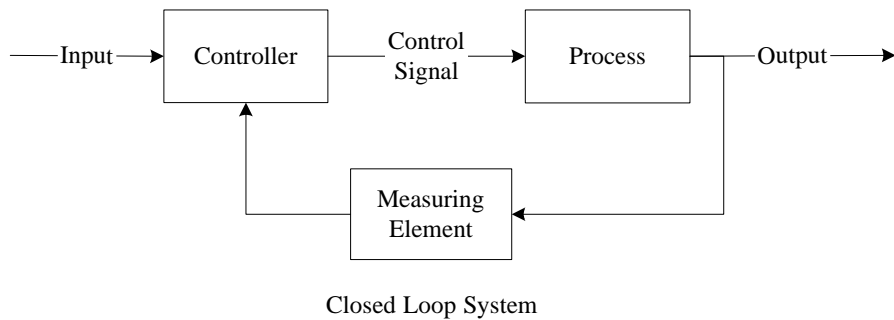
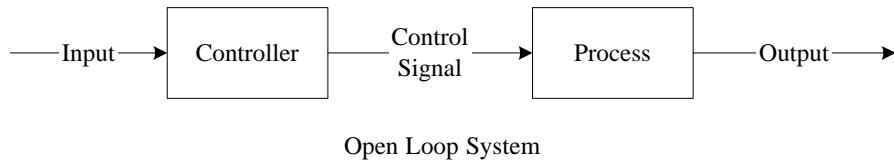


Figure 6 – Open and Closed Loop Systems

2.1.1 General Requirements for a Control System

Any useful control system must satisfy the following conditions:

- The first requirement of any control system is stability.
- In addition to absolute stability, the control system must have relative stability, that is the speed of response must be reasonably fast and must show reasonable damping.
- A control system must be capable of reducing errors to zero or to some small tolerance level.

The requirement for relative stability and steady-state accuracy are actually incompatible. The design of a control system becomes a tradeoff between these two requirements.

2.1.2 Adaptive Controls

Adaptation implies the ability to self-adjust or self-modify with unpredictable changes in conditions of environment or structure. In an adaptive control system, the dynamic characteristics must be identified at all times so that the controller parameters can be adjusted in order to maintain optimal performance.

2.2 Basic Approach to Control Systems Design

One approach to the design of control systems, which will be useful here, is to use *block diagrams*, which are pictorial representations of the functions performed by each component of the system and the *signals* that flow between these components.⁸

Figure 7 is a logical depiction of a closed loop control system. This system consists of two elements:

- Block element – is the symbol of the operation performed on the input signal to produce the output signal. The notation inside the block is usually the transfer function of the block given as the Laplace function.

⁸ For the moment the specific notation used in Figure 7 will be ignored, since the interest is in applying control systems theory to project management. The “functions” $R(s)$, $E(s)$, and $C(s)$, represent the reference, error, and control signals respectively. These are functions of Laplace space rather than of time. For not familiar with the Laplace transform it is defined as $\mathcal{L}[f(t)] = F(s) = \int_0^{\infty} e^{-st} dt [f(t)] = \int_0^{\infty} f(t) e^{-st} dt$. By transforming a time varying function to *Laplace space* it can be manipulated as an algebraic expression rather than as a differential equation.

- Error detector – produces an error signal, $E(s)$, which is the difference between the reference input, $R(s)$ and the feedback signal, $C(s)$. The choice of the error signal is very important. Any imperfections in the error signal will be reflected in the performance of entire system.

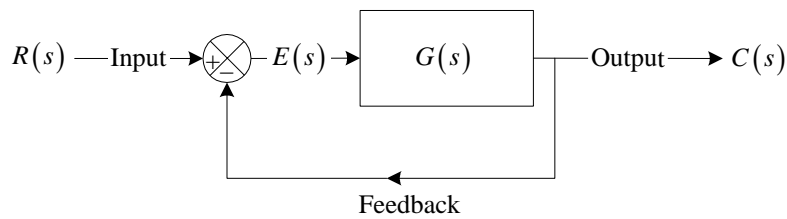


Figure 7 – A Logical Depiction of a Closed Loop Control Systems

2.3 Adaptive Controls Design

In most feedback systems, small deviations in parameters values from their design values will not cause any problem in the normal operations of the system, provided these parameters are inside the loop. If the process parameters vary widely because of environmental changes, then the control system will exhibit unsatisfactory behaviors. In some cases large variations in process parameters will cause instability in non-adaptive systems.

A simple definition of a adaptive control system is: a control system in which continuous and automatic measurements of the dynamic characteristics of the process are taken, comparisons are made with the desired dynamic characteristics, and differences uses to adjust the system parameters – usually the controller characteristics – or the

generation of an actuating signal so as to maintain optimal system performance, regardless of the environmental changes to the process.

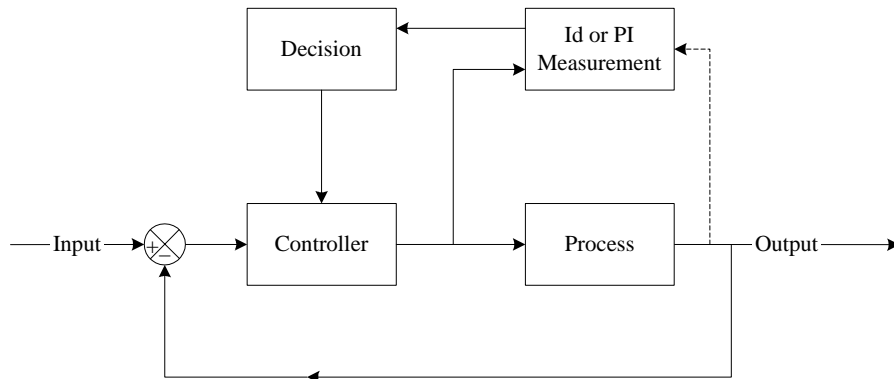


Figure 8 – Adaptive Controller

To be called adaptive, some form of self-organizing features must exist. An adaptive controller consists of the following three functions:

- Identification of the dynamic characteristics of the process.
- Decision making based on the identification of the process.
- Modification or actuation based on the decisions made.

By performing these functions *continuously*, self-organization can take place to compensate for unpredictable changes in the process.

2.3.1 Identification

The dynamic characteristics of the process must be measured and identified continuously. This measure should be accomplished with effecting the normal operation of the system. Identification may be made from normal operating data or by the injection

of test signals. Identification with normal data is possible only when this data has adequate signal characteristics (bandwidth, amplitude, etc.) for proper identification.

2.3.2 Decision making

Decisions are made on the basis of the process characteristics, which have identified and on the computed performance index. Once the process has been identified, it is compared with the optimal characteristics (or optimal performance), and then a decision made as to how the adjustable controller characteristics should be varied in order to maintain optimal performance.

2.3.3 Modification based on Decisions Made

Modification refers to the changes of control signals according to the results of the identification and decision processes. There are two approaches to modifying controls signals:

- Controller parameter modification – in which the controller parameters are adjusted, to compensate for changes in the process dynamics.
- Control signal synthesis – in which optimal control signals are synthesized based on the transfer function, performance index, and desired transient response of the process.

3 Project Management Theory as Control Theory

3.1 Control Theory

Control is a guiding a set of variables towards a common goal. *Management Control Theory* may be seen as *after-the-fact* control or *before-the-fact* control. Control theory, suggests that where consequences are easily monitored, after-the-fact controls are more effective. Where consequences are unique and hard to monitor, before-the-fact control is appropriate.

4 Agile PM and Adaptive Control

What is needed now is some way to tie adaptive control theory to agile project management. A simple approach is to compare the primary attributes of adaptive control with agile PM methods.

Adaptive Control	Agile Project Management
Identification	
Decision Making	
Modification based on the decision made	

5 A Framework for Traditional Project Management Processes

One question is *are the methods described in traditional PM frameworks appropriate for Agile Project Management?* One place to look for traditional frameworks if the

Project Management Institute’s Project Management Body of Knowledge. There are other *BoK*’s but PMBOK will be a good starting point.

First let’s look at the control block picture of the PMBOK’s functions. Figure 9 describes a *simple* view of PMBOK’s control elements.

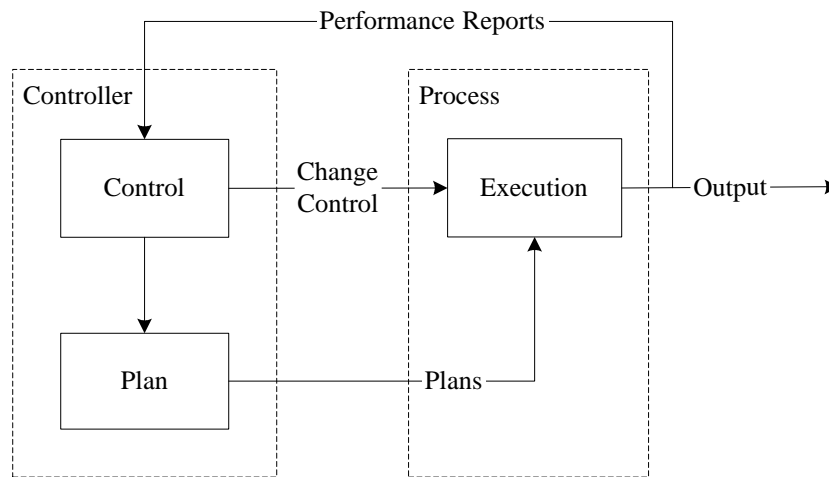


Figure 9 – PMBOK Control Blocks

5.1 What’s Missing?

In Figure 9 there several things missing when viewed from a traditional control loop process.

- There is no reference signal – the flow of control makes use of performance reports to define the change control signal. These performance reports have no reference signal by which create a “error” signal.
- There are multiple control signals – both plans and change control are used as a control signal.

- The dynamics and transfer function of each process is not specified. This includes the sample rate and the response rate of each process.

6 Bibliography

1. Alexander, Christopher, *Notes on the Synthesis of Form*, Harvard University Press, 1964.
2. Alexander, Christopher, *A Timeless Way of Building*, Oxford University Press, 1979.
3. Agresti, *New Paradigms for Software Development*, IEEE Computer Society Press, 1986.
4. Aoyama, Mikio, "Agile Software Process and Its Experience," *International Conference on Software Engineering*, 1998.
5. Ballard, Glenn, "The Last Planner System of Production Control," thesis submitted to the Faculty of Engineering, School of Engineering, University of Birmingham, 2000.
6. Basili, Victor, "Iterative Enhancement: A Practical Technique for Software Improvement," *IEEE Transactions on Software Engineering*, 1(4), December 1975.
7. Bateman, T. S. and C. P. Zeithaml, *Management: Function and Strategy*, Irwin, 1990.
8. von Bertalanffy, Ludwig, *General Systems Theory: Foundations, Development, and Theory*, George Braziller, 1976.
9. Boehm, Barry, "Getting Ready for Agile Methods, with Care," *IEEE Computer*, 35(1), January 2002, pp. 64–69.
10. Charette, Robert N., "Large-Scale Project Management is Risk Management," *IEEE Software*, pp. 110–117, July 1996.

11. Christensen, Mark J. and Richard H. Thayer, *The Project Manager's Guide to Software Engineering's Best Practices*, Computer Society Press, 2002.
12. Cleland, David I., *Project Management: Strategic Design and Implementations*, McGraw Hill, 1998.
13. Charette, Robert N., "Large-Scale Project Management is Risk Management," *IEEE Software*, pp. 110–117, July 1996.
14. Cook, H. E., *Product Management – Value, Quality, Cost, Price, Profit and Organization*, Chapman & Hall, 1997.
15. Davis, Alan M., "Fifteen Principles of Software Engineering," *IEEE Software*, 11(6), pp. 94–96, November/December, 1994.
16. Dempster, Beth, "Science versus Post-Normal Science,"
<http://www.fes.uwaterloo.ca/u/mbldemps>
17. Duncan, William, *A Guide to the Project Management Body of Knowledge*, Project Management Institute, 2000.
18. Earl, Michael, Jeffery Sampler, and James Short, "Strategies for Reengineering: Different ways of Initiating and Implementing Business Process Change," Centre for Research in Information Management, London Business School, 1995.
19. Earl, Michael, "Information Systems Strategy: Why Planning Techniques are Never the Answer," Centre for Research in Information Management, London Business School, 1995.
20. Erdogmus, H., "Valuation Of Complex Options In Software Development," *First Workshop on Economics-Driven Software Engineering Research, EDSE-1*, May 17, 1999.

21. Flatto, Jerry, "The Role of Real Options in Valuing Information Technology Projects," *Association of Information Systems Conference*, 1996.
22. Funtowicz, S. and J. Ravetz, "Post-Normal Science: A New Science for New Times," *Scientific European*, pp. 95–97, March 1992.
23. Georgescu-Roegen, Nicholas, *The Entropy Laws and Economic Progress*, Harvard University, 1971.
24. Feng, Gang, *Adaptive Control Systems*, Newnes, 1999.
25. Foster, Jason, James Kay, and Peter Roe, "Teaching Complexity and Systems Theory to Engineers," 4th UICEE Annual Conference on Engineering Education, 7–10 February 2001.
26. Funtowicz, S. and Jerome R. Ravetz, "Post-Normal Science: A New Science for New Times," *Scientific European*, pp. 95–97, March 1992. Also in *Futures*, 25(7), pp. 739–751.
27. Funtowicz S, and Ravetz, "Post-Normal Science – An Insight Now Maturing," *Futures*, 31:641–646, 1999.
28. Funtowicz S, and Jerome R. Ravetz, "Three Types of Risk Assessment and the Emergence of Post-Normal Science", in Krinsky S, and Golding D (editors), *Social Theories of Risk*, Westport CT, Greenwood. Pp. 251–273, 1992.
29. Glass, Robert L., *Software Runaways: Lessons Learned from Massive Software Project Failures*, Prentice Hall, 1998.
30. Giglioni, Giovanni B. "A Conspectus of Management Control Theory: 1900–1972," *Academy of Management Journal*, 17(2), June 1974.
31. Hallows, Jolyon E. *Information Systems Project Management*, AMACOM, 1997.

32. Harmsen, Frank, Ivo Lubbers, and Gerard Wijers, "Success-Driven Selection of Fragments for Situational Methods: The S3 Model," Design Methodology Research Group, Department of Computer Science, University of Twente.
33. Hofstade, Geert, "The Poverty of Management Control Philosophy," *Academy of Management Review*, July 1979, pp. 450-461.
34. Howell, Greg and Lauri Koskela, "Reforming Project Management: The Role of Lean Construction," *Eighth Annual Conference of the International Group for Lean Construction*, (IGLC-8), July 2000.
35. Jackson, M. C., "Towards Coherent Pluralism in Management Science," *Journal of Operational Research*, 50(1), pp. 12-22, 1999.
36. Jackson, E. T., "Teaching Project Management for the 21st Century: Why it is Important and What is New?" Carleton University, School Of Business Administration, Ottawa, Ontario K1S 5B6, Canada.
37. Jones, Capers, "What it Means to be Best in Class," Version 5, February 10, 1998.
38. Jones, Capers, *Patterns of Software Systems Failure and Success*, International Thompson Computer Press, 1996.
39. Kogut, Bruce and Nalin Kulatilaka, "Strategy, Heuristics, and Real Options," *The Oxford Handbook of Strategy (2001)*, Chapter 30, 2001.
40. Kogut, Bruce and Nalin Kulatilaka, "Strategy, Heuristics, and Real Options," *The Oxford Handbook of Strategy (2001)*, Chapter 30, 2001.
41. Kogut, Bruce and Nalin Kulatilaka, "What is Critical Capability?" Reginald H. Jones Center Working Paper, Wharton School, 1992.

42. Koskela, Lauri and Greg Howell, "Reforming Project Management: The Role of Planning, Execution, and Controlling," Ninth Annual Conference of the International Group for Lean Construction, (IGLC-9), 2001.
43. Koskela, Lauri, "We need a theory of construction," *Berkeley-Stanford CE&M Workshop: Defining a Research Agenda for AEC Process/Product Development in 2000 and Beyond*. Stanford, 26 – 28 Aug. 1999. Berkeley. University of California; Stanford University, 1999.
44. Kuhn, T. S., *Structure of Scientific Revolutions*, Chicago University Press, 1962.
45. Lewis, Marianne, M. Ann Welsh, Gordon E. Dehler, and Stephen G. Green, "Product Development Tensions: Exploring Contrasting Styles of Project Management," *Academy of Management*, forthcoming, 2002.
46. Luks, F., "Post-Normal Science and the Rhetoric of Inquiry: Deconstruction Normal Science?," *Futures*, 31(7), pp. 705–719, 1999.
47. May, Lorin J., "Major Causes of Software Failures," *Crosstalk*, July 1998.
48. Maciarello, Joseph A. and Calvin J. Kirby, *Management Control Systems: Using Adaptive Systems to Attain Control*, 2nd Edition, Prentice Hall 1994.
49. McCarthy, Dan, "Normal Science and Post-Normal Inquiry," University of Waterloo, Waterloo Ontario.
50. Morris, Peter W. G., "Researching the Unanswered Questions of Project Management," *Project Management Research at the turn of the Millennium*, Proceedings of PMI Research Conference, June 2000, pp. 87–101.
51. Nelles, Oliver, *Nonlinear Systems Identification: From Classical Approaches to Neural Networks and Fuzzy Models*, Springer Verlag, 2000.

52. Ogata, Katsuhiko, *Modern Control Engineering*, 4th Edition, Prentice Hall, 2002.
53. Osterweil, Leon J., "Software Processes are Software Too," *Proceedings of the 9th International Conference on Software Engineering (ICSE 1987)*, pp. 2–13, March 1987, Monterey, CA.
54. Osterweil, Leon J., "Software Processes Are Software Too, Revisited," *Proceedings of the 19th International Conference on Software Engineering (ICSE 1997)*, pp. 540–548, May 1997, Boston, MA.
55. Pajares, Frank, "The Structure of Scientific Revolutions," Outline and Study Guide, Emory University, <http://www.emory.edu/EDUCATION/mfp/Kuhn.html>.
56. Potters, Marc, et. al. "Financial Markets as Adaptive Ecosystems," May 31, 2001. arXiv:cond-mat/9609172.
57. Ravetz, Jerome R., "What is Post-Normal Science," *Futures*, 31(7), pp. 647–653, 1999.
58. Ravetz, Jerome R. and Silvio Funtowicz, "Post-Normal Science: An Insight now Maturing," *Futures*, 31(7), pp. 641–646, 1999.
59. Rechtin, *System Architecture: 2nd Edition*, CRC Press, 2000.
60. Rockart, J. F. and C. V. Bullen, "A Primer on Critical Success Factors," Center for Information Systems Research, Working Paper No. 69, Sloan School of Management, MIT, 1981.
61. Rockart, J. F., M. Earl, and J. Roos, "Eight Imperatives for the New IT Organization," *Sloan Management Review*, Fall, 1996, pp. 43–56.
62. Royce, Winston W., "Managing the Development of Large Scale Software Systems," *Proceedings of IEEE WESCON*, pp. 1–9, August 1970.

63. Royce, Walker, *Software Project Management*, Addison Wesley, 1998.
64. Selfridge, Oliver G. (editor), *Adaptive Control of Ill-Defined Systems*, Plenum Publishing, 1984.
65. Shenhar, A. J. and D. Dvir, "Toward a Typological Theory of Project Management," *Research Policy*, 25(4), pp. 607–, 1996.
66. Sullivan, Kevin, P. Chalasani, S. Jha, and V. Sazawal, "Software Design as an Investment Activity: A Real Options Perspective," in *Real Options and Business Strategy: Applications to Decision-Making*, edited by Lenos Trigeorgis, Rick Books, 1999.
67. Szulanski, Gabriel, "Unpacking Stickiness: An Empirical Investigation of the Barriers to Transfer Best Practices Inside the Firm," INSEAD Study, *Academy of Management Best Paper Proceedings*, pp. 437–441, November 1995.
68. Tao, Gang, *Adaptive Control Design and Analysis*, John Wiley & Sons, 2003.
69. Thorburn, W. M. "The Myth of Occam's Razor," *Mind* 27:345–353, 1918.