Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 20 (2013) 325 – 330

Procedia Computer Science

Complex Adaptive Systems, Publication 3
Cihan H. Dagli, Editor in Chief
Conference Organized by Missouri University of Science and Technology
2013 - Baltimore, MD

The Root Cause of Failure in Complex IT Projects: Complexity Itself

Kaitlynn M. Whitney[a], MEM, Charles B. Daniels[b], Ph.D., PEM, PMP,

[a]*Old Dominion University, 2401 Kaufman Hall, Norfolk, Virginia 23529, USA*
[b]*Old Dominion University, 2401 Kaufman Hall, Norfolk, Virginia 23529, USA*

Abstract

Increased demand for better technology and perpetual global expansion continue to provide developers with many project opportunities for success, as well as failure. While no industry is immune from project failure, the Information Technology (IT) industry is shown to be more susceptible to risk and failure than those of other industries. Agile project management, which facilitates adaptation to changing circumstances and alleviates rigid formal controls, has become more popular in the software development industry though is not entirely compatible with traditional project management approaches.

In this paper we will examine the primary causes of IT project management failure stated in modern literature, analyze these causes, and discuss the degree of complexity within the projects from a systemic perspective related to emergence, non-monotonicity, and non-ergodicity. The paper concludes with some conceptual management approaches that respond to these "true" root causes of failure, for applications in agile organizations and beyond.

Selection and peer-review under responsibility of Missouri University of Science and Technology

Keywords: project management, project failure, information technology, information systems, complex adaptive systems, complexity, PMBOK, agile

1. Investigating Failure in Complex Projects

Any undertaking that involves creating a new product or process is fraught with peril, but IT projects regularly fail. In a study published by The Standish Group of over 50,000 IT projects between 1992 and 2004, only 29 percent could be classified as successes [1]. Most project failures can be classified into one or more of the following categories: (1) failure to meet the approved schedule, (2) failure to achieve cost objectives, and (3) failure to provide the expected project scope. These aspects of failure are often characteristics within the following four categories of failure, defined by Lyytien and Hirchheim [2]:

- Correspondence failure: Systems design objectives or specifications not met.
- Process failure: System cannot be developed within the allocated budget or schedule.
- Interaction failure: User attitude, satisfaction, and frequency of use do not correspond to the level of system usage, i.e. the system is implemented out of necessity and without increased task performance.
- Expectation failure: System does not meet stakeholder requirements, expectations, or values.

Selection and peer-review under responsibility of Missouri University of Science and Technology
doi:10.1016/j.procs.2013.09.280

1.1. Commonly Stated Causes of Project Failures

There are at least two types of projects to consider when evaluating causes of failure [3]. Type 1 are well understood, routine projects with a clearly defined scope and few unknowns. The extent of their complexity is that they may be intensely detailed. They may run late or over budget though will only fail if technical expertise is lacking to handle unexpected deviations from the plan. Type 2 projects, also considered *complex*, typically have many unknowns and an unclear scope. Difficulties may arise in these projects even in the beginning, and the client will often not approve the project itself. Planning problems, especially those dealing with defining project scope, are generally a major cause of failure when dealing with projects that are of the second type [3].

The term "perceived failure" was fathered by Dr. Harold Kerzner who suggested that failed projects are the result of some combination of both "actual failure" and "planning failure" [4]. *Actual failure* occurs because there is a discrepancy between what was planned and what was accomplished, whereas *planning failure* occurs because there is a discrepancy in what was planned and what was actually achievable. Kerzner acknowledged the fact that human dynamics play an important role in project management failure, citing poor motivation, productivity, and human relations; lack of employee and functional commitment; delayed problem solving; and unresolved policy and stakeholder issues [4].

Murray provided the following attributing factors of IT project failure, some which are characteristic of tendencies observed in complex projects [5]:

- Unrealistic project scope given the available resources and project development experience.
- Improper management of scope creep, the continuous expansion of the project scope.
- New technology that is critical to the project has not been previously developed.
- The organization's issues are not understood.
- Custom work is needed for the organization's business activities.

Kweku Ewusi-Mensah, a professor of Information Systems at Loyola Marymount University, developed a unique view of project failure by focusing on discarded projects, in particular the ones that were cancelled by managers or sponsors because they believed that the project would not be successful. He projected the risk factors of abandonment that were associated with these projects and paralleled them to those theorized in software risk management research [6]. A comprehensive overview of what he identified as "abandonment factors" is shown in Table 1, as well as the other risk factors he found most noteworthy in literature.

Table 1: Software management risks

Boehm, 1991	Ropponen & Lyytinen, 2000	Ewusi-Mensah, 2003: "Abandonment Factors"
• Personnel shortfall and straining computer science abilities • Unrealistic schedules and budgets • Developing wrong functions, properties, and/or user interfaces • Constantly changing requirements • Shortfalls in procured components or labor	• Scheduling and timing • System functionality • Subcontracting • Requirements management • Resource usage and performance • Personnel management	• Unrealistic project goals and objectives • Poor project team composition • Project management and control problems • Inadequate technical expertise • Problematic technology base/infrastructure • Lack of executive or support/commitment • Changing requirements • Cost overruns and schedule delays

These stated causes of failure are indeed striking; however, many appear to be manifested as by-products of two enveloped root causes: a lack of adequate and sufficient resources (including skilled management personnel provided for the project) and the complexity inherent within the project itself.

Recent studies in IT project failure broaden the paradigm slightly by including the complexity and size of a multi-faceted projects as a root cause [7]. Project management education teaches us that identifying and considering areas of risk as well as their impact in the beginning and throughout the project can increase the likelihood of success. Even so, the major frameworks used are not conducive to understanding the underlying nature of systemic complexity inherent in complex adaptive systems (CAS).

1.2. Complex Adaptive Systems (CAS) Definition

For the purpose of this paper, we will define CAS as (1) *non-linear:* as a whole, are unable to be determined or represented through the sum of their components and subsystems; (2) *non-ergodic:* interacting with its environment by receiving inputs and providing outputs, but with limited control over the outcomes; and (3) *emergent:* dynamic in that it changes and evolves its behavior in response to its inputs. Order emerges through the interaction among the system's parts as they evolve (within the larger system) in response to the changing environment.

2. The Inherent Complexity in Projects

Complexity issues in elaborate IT projects will be present even when the most optimal development methodologies are used to achieve the specific organizational goals. In developing a deeper understanding of these issues we may hopefully be more adept in managing them.

There are two dimensions to consider when evaluating an IT project: the amount of turbulence caused by project volatility and internal and external uncertainty, and the degree of which the project's structure encompasses a traditional management approach [9]. External organizations soliciting the development of technology systems essential for their operations generally do not understand system development methodology and process capabilities, nor do they comprehend the necessary upgrades and maintenance. Their efforts will often bring them into the formidable territories that exemplify the nature of complex systems, which we will begin discussing in this paper.

2.1. Simple/Rational System Approaches to Complex Systems

Unlike a rational, simple system, the complete knowledge of a complex system exists at a tacit level that we will never be able to fully understand or precisely represent in a model. A representation of a complex system will be always be incomplete, abstracted, historical, and subject to the perception of the observer's vantage point in time [10]. Yet traditional management approaches treat projects like "rational" systems and considers the artifacts that result from the planning and design phases of the project as part of the *formalized structure* to be executed with encompassing authority of the project manager. Their *bureaucracy* comes in the form of cumbersome documentation processes that provide limited benefit as software design documents are constantly becoming outdated as requirements become more defined. For this reason, agile development methodologies have become more appropriate for IT projects because they embrace a loosely defined design phase.

2.2. Actors in the Complex System Environment

Stafford Beer describes the project team as *homeostats*, regulating their internal environment to maintain stability with discrete goals. Beer noted that two problems generally plague management, (1) the "artful design of information flows - so that the firm could be quickly and adequately informed of what the outside world was actually doing" and (2) "acting on those flows" [11]. The project manager's information flow is designed based on the baselined plan, while the project team's information flow is directly connected to their environment, in which they react in a manner similar to Beer's homeostasis: "always in the thick of things - responding to changing inputs in real-time; reconfiguring themselves internally to change their outputs; monitoring what came back at them from the world" [11]. According to Ashby's Law of Requisite Variety, maintaining homeostasis is successful when the observers, either through flexibility, which effectively adapts to change, or by systemically reducing enough uncertainty (variety) within the complex system, are able to maintain or control it.

Much to the observer's dismay, this can lead to major complications, as we will discuss in the next section. By recognizing their powerlessness and inability to have complete control, greater value is emphasized in having complementary perspectives, an important aspect of working with other actors, including external stakeholders.

2.3. Non-linear Behavior within a System

Anomalies, variation, and unexpected events are to be anticipated when executing projects involving complex systems. As we saw with emergence, this can mean different things for a complex system's entities and their interrelations. The incompressible nature of the complex system implies that the behavior and interrelationships of the system are non-monotonous and non-linear [12]. This characteristic often renders the incognizant project manager's efforts to detect an overall trend or pattern useless: s/he cannot anticipate future complications, delays in delivery, or a change in requirements.

Unexpected events also have the potential to be positive. It may have been previously estimated that more time would be needed to develop a particular feature, meaning more time is available to work on optional features or finish early. The outcomes of random events (negative or positive) that occur in projects may influence future decisions. If some random event creates a positive outcome, it may seep into the project management body of knowledge as a best practice. In the computer industry, the reuse revolution of open source software and the overall trend toward knowledge resources has led to efforts in storing code and other assets from prior projects in databases for reuse. Unfortunately, this requires a deep understanding of the asset in question and the manner of reapplying the code to meet the new conditions. In many cases the rework can also take longer than the work, or the reapplication would prove to be inappropriate. Skills, focus, and good requirements are necessary when assuming that any steps of the project will be predictable.

2.4. Non-ergodicity within a System

A system that exhibits non-ergodicity is characteristically one in which a subsequent stage depends only on the current stage, and, over time, a system that continuously evolves could inevitably “forget” its initial state, as described by Markov Processes. At this point, a project manager is forced to deal with a newly defined problem without the ability to return to prior stages for reasons such as limited time or resources, or for reasons that regression of a functionally emergent system is either too arduous or an impossibility.

The nature of non-ergodicity poses significant problems in complex IT projects that are breaking new ground or require a unique design. Changing requirements is natural and will make the system better, although it can affect prior work completed or future planned work. These emergent requirements are generally perceived as unfavorable to those working on the project. Rework is often necessary when working on project releases that significantly impact the functionality of existing modules. Iterative methodology is a flexible approach for these situations, however if the client is involved, they may not be satisfied with the project team's interpretation of the fulfilled requirements during a phase and will request *seemingly* minor changes throughout the process while significantly impacting project progress. When working on a part of the system that disables functionality of other components, the system is essentially “broken” and without functionality until the addition or modification is reconciled with its interacting components. Undesirable results after these types of releases can leave a project far behind schedule. Often times, in the case of an overrun budget, one must decide to either *fold* or *push* without knowing if proceeding will improve or severely limit other aspects of the design. Some *flash cutover* conversions that make a significant change in a complex system without gradual migration also may not allow for regression.

2.5. Emergence within a System

In the search for understanding and control, an observer will try to decompose a phenomenon into ever smaller and smaller elements, attempting to exert more control of those decomposed elements and understand the behavior of the system. This is ineffective for complex systems. Treating them as static, decomposable hierarchies of modules or “work packages” provides essentially the same results as inappropriate fast tracking by resulting in rework and an increased risk of technical incompatibility. In these circumstances, one may use *rolling wave planning*, a form of progressive elaboration commonly used in agile methodologies. *Rolling wave planning* allows for initially high-level planning in the WBS to evolve into more detailed planning iteratively.

Emergence invalidates the notion of modularity for complex projects, yet decomposition and integration is often common practice in large coding teams. New components and interactions also may not be compatible with the existing system, and may not be integrable when they are performing in an optimal way. It also might not be possible to invest additional resources in obtaining system optimization and compatibility. In these cases, one may have to accept a loss of functionality, or prioritize attributes such as security versus stability, or speed versus access.

3. Project Management Institute (PMI) Resources and Certifications for IT Project Managers

PMI, the world's largest project management certification body, is internationally recognized for its development of *A Guide to the Project Management Body of Knowledge* (PMBOK)® with the intentions of clearly articulating “generally recognized” knowledge, skills, tools and techniques that are considered “good practice” and applicable to most projects, most of the time [8]. PMI offers the Project Management Professional (PMP)® certification, a globally recognized and demanded credential allowing project managers to demonstrate their competency. Proficient demonstration and application of the knowledge and skills found in PMBOK® is required for PMP certification.

PMI states that many project management methodologies can implement its project management framework found in PMBOK®, including agile methodologies. PMI offers the Agile Certified Practitioner (PMI-ACP)® to recognize professional proficiency in agile methodologies implemented in organizations and on projects (not to be misrepresented as a project management certification). Currently there is not an agile body of knowledge comparable to the PMBOK®, although many third party references are recommended for preparation of the PMI-ACP® exam.

4. Discussion

Concomitant with the assumption of perfect *a priori* knowledge is a concept the PMI calls *progressive elaboration* [8]. When a project team executes a project's plan, it may choose to implement progressive elaboration, defined as "the iterative process of increasing the level of detail in a project management plan as greater amounts of information and more accurate estimates become available" [9]. Note that traditionally, once a project is baselined, the baseline can only be changed through formal change control procedures, a practice that divorces the individuals who perform the daily work.

Although PMBOK® recognizes the benefits of "capitalizing" on the different backgrounds of each of the team members, there is no emphasis of how these unique perspectives may impact one's systemic perspective, although complementarity is crucially relevant when working on a complex project as it contributes to a more wholesome and edifying view of the project boundaries. Analysis of the contextual plane should precede planning as it provides critical input information for each of the key project management processes. Planning processes defined in PMBOK® presume nearly perfect knowledge about the relative contexts and path of the project, even being that the inputs to these processes provide for limited basis for contextual analysis of the various technical, social, organizational, managerial, and political dimensions of a project. Instead, many contextual factors in PMBOK® are casually grouped together under the categorical process inputs "enterprise environmental factors" and "organizational process assets," superficially described and with limited examples of their potential impact and implications. Human dynamics may (more often than not) affect process inputs, especially with regard to the social and political contexts. As such, the lack of consideration of these dimensions may even result in selecting a suboptimal methodology or ineffective project team members, essentially setting the project up to fail.

It is notable that information systems project failure is more attributed to organizational and communication related issues than to technological issues [11]. In a study of the most valuable project management competencies according to IT recruiters, leadership, the ability to communicate at multiple levels, verbal and written skills, and the ability to deal with ambiguity and change were viewed as more important than experience, work history, education, and expertise [14]. Since many of the issues that arise in the execution of complex projects are socio-technical in nature and require a talented and motivated team, project management skills should be accompanied by emotional intelligence (EQ) and spiritual intelligence (SQ) to adequately address project uncertainty and complexity [15].

In the IT industry, human-centric, collaborative (agile) development methodologies such as Extreme Programming (XP) and Scrum are beginning to become more popular in response to the shift from traditional management practices, modern technology movements, globalization, and non-collocated teamwork. These methodologies are more flexible for use in volatile project environments. They are also versatile in accommodating changing circumstances observed in CAS projects. These contemporary methodologies pride themselves in being adaptive rather than predictive, welcoming collaboration among all levels, receptive of a client's changing requirements, and allowing plans and development to evolve through iterations. Although this idea works well for small projects and teams, it is not typically successful in large, complex projects that are mission-critical and require more rigidity [13]. Interestingly enough, emerging research is beginning to articulate the presence of complexity in projects and warns of the hazards of treating a complex system as a linear system [16].

5. Conclusion

Effectively managing the added complexity of the agile mind-set is still a new research area as agile methodology is new to university curricula. Complexity paradigms are necessary yet absent in project management education and credentialing frameworks. The inclusion of complexity not only encompasses conventional beliefs about failure; it shifts blame from humans and the technologies they develop and manage by refocusing attention on the powerful, enigmatic nature of a complex system.

Teams that perform cohesively and purposefully (under the guidance of an effective project manager, team leader or otherwise) are more likely to successfully identify and overcome uncertainties in a complex adaptive system. By developing soft skills, like empathy, influence, creativity, group facilitation, and others that are essential elements of successful socio-technical ventures, we stand a better chance in building understanding when dealing with a complex system and a dynamic environment. Future project managers who take on the challenge of complex adaptive projects will be well-advised to understand human behavior and interaction, be able to motivate project team members and infuse meaning into a situation, and be conscious of the higher levels of human values.

Beyond possessing the skills, knowledge, and capabilities measured by the Project Management Professional (PMP)® and Agile Certified Practitioner (PMI-ACP)® certifications, one should also strive to attain higher levels of emotional intelligence (EQ) and spiritual intelligence (SQ), achievable through an intentional development process. Research topics such as personality, human behavior, positive organizational behavior, ontology, teleology, and other branches of classical philosophy hold significant promise as a starting point for a robust curriculum to enhance project manager development.

References

1. Johnson, J., *My Life Is Failure: 100 Things You Should Know to Be a Better Project Leader*. 2006: Standish Group International. 166.
2. Lyytinen, K. and R. Hirschheim, *Information failures - a survey and classification of the empirical literature.*, in *Oxford Surveys in Information Technology*, P. Zorkoczy, Editor. 1987, Oxford University Press, Inc. p. 257-309.
3. Meredith, J.R. and S.J. Mantel, *Project management : a managerial approach*. 2002, Hoboken, N.J.: J. Wiley.
4. Kerzner, H., *Project management : a systems approach to planning, scheduling and controlling*. 2009, Hoboken, New Jersey: Wiley & Sons.
5. Murray, J.P., *Reducing IT project complexity.* Information Strategy: The Executive's Journal, 2000. **16**(3): p. 30.
6. Ewusi-Mensah, K. *Software development failures anatomy of abandoned projects*. 2003.
7. Al-Ahmad, W., et al., *A Taxonomy of an IT Project Failure: Root Causes.* International Management Review, 2009. **5**(1): p. 93-104.
8. *A Guide to the Project Management Body of Knowledge (PMBOK®)*. Fourth ed. 2008, Newtown Square, PA: Project Management Institute.
9. Bardhan, I., S. Mithas, and L. Shu, *Performance Impacts of Strategy, Information Technology Applications, and Business Process Outsourcing in U.S. Manufacturing Plants.* Production & Operations Management, 2007. **16**(6): p. 747-762.
10. Skyttner, L., *General systems theory : ideas & applications*. 2001, Singapore; River Edge, N.J.: World Scientific.
11. Beer, S., *Designing freedom*. 1974, London; New York: J. Wiley.
12. Richardson, K.A., P. Cilliers, and M. Lissack, *Complexity Science: A "Gray" Science for the "Stuff in Between".* Emergence, 2001. **3**(2): p. 6-18.
13. Khan, A.I., R.J. Qurashi, and U.A. Khan, *A Comprehensive Study of Commonly Practiced Heavy and Light Weight Software Methodologies.* IJCSI International Journal of Computer Science Issues,, 2011. **8**(4): p. 441-450.
14. Stevenson, D.H. and J.A. Starkweather, *PM critical competency index: IT execs prefer soft skills.* International Journal of Project Management, 2010. **28**(7): p. 663-671.
15. Thomas, J. and T. Mengel, *Preparing project managers to deal with complexity – Advanced project management education.* International Journal of Project Management, 2008. **26**(3): p. 304-315.
16. Rogers, K., R. Luton, et al., *Fostering complexity thinking in action research for change in social-ecological systems.* Ecology and Society. 2013, **18**(2): 31.